User's Manual

16

RL78/D1A

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.10 Mar 2015

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 RL78/D1A and design and develop application systems and programs for these devices.
The target products are as follows.

- 48-pin: R5F10CGxJ, R5F10DGxJ, R5F10CGxL, R5F10DGxL (x = B, C, D)
- 64-pin: R5F10CLDJ, R5F10DLxJ, R5F10CLDL, R5F10DLxL (x = D, E)
- 80-pin: R5F10CMxJ, R5F10CMxL (x = D, E)
  R5F10DMxJ, R5F10DMxL (x = D, E, F, G, J)
- 100-pin: R5F10DPxJ, R5F10DPxL (x = E, F, G, J)
  R5F10TPJJ, R5F10TPJL

Purpose

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

Organization

The RL78/D1A manual is separated into two parts: this manual and the instructions edition (common to the RL78 Microcontroller).

RL78/D1A User's Manual (This Manual)	**RL78 Microcontroller User's Manual Instructions**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications (target)	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

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

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
RL78/D1A User's Manual Hardware	This manual
RL78 Microcontroller Instructions User's Manual	R01US0015E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E

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

Other Documents

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

Note See the “Semiconductor Device 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

<R> 1.1 Features

- Minimum instruction execution time can be changed from high speed (0.03125 μs: @ 32 MHz operation with high-speed on-chip oscillator clock) to ultra low-speed (30.5 μs: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM: 24 to 512 KB, RAM: 2 to 24 KB, Data flash memory: 8 KB
- On-chip high-speed on-chip oscillator clocks
  - Select from 32 MHz (TYP.), 24 MHz (TYP.), 16 MHz (TYP.), 8 MHz (TYP.), and 4 MHz (TYP.)
- On-chip single-power-supply flash memory (with prohibition of block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-reset (POR) circuit and voltage detector (LVD)
- On-chip watchdog timer (operable with the dedicated low-speed on-chip oscillator clock)
- On-chip multiplier and divider/multiply-accumulator
  - 16 bits × 16 bits = 32 bits (Unsigned or signed)
  - 32 bits ÷ 32 bits = 32 bits (Unsigned)
  - 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports: 38 to 112
  - CMOS I/O port: 35 to 107 (LED direct drive port: 9 to 16, N-ch OD selectable port: 4 to 6)
  - CMOS input port: 5
  - CMOS output port: 0 to 1
- Timer
  - 16-bit timer: 24 channels
  - Watchdog timer: 1 channel
  - Real-time clock: 1 channel
  - Interval timer: 1 channel
- Serial interface
  - CSI
  - UART (LIN-bus supported)
  - Simplified I^2C communication
  - aFCAN controller
- Stepper motor controller/driver with zero point detection (ZPD): 1, 2, 4-channels
- LCD controller/driver (seg × com): 27 × 4, 39 × 4, 48 × 4, 53 × 4 and 54 × 4
- LCD Bus I/F
- RESET output
- STOP status output
- Sound generator
- 8/10-bit resolution A/D converter (V_{DD} = EV_{DD} =2.7 to 5.5 V): 3+2 to 9+2 channels
- Standby function: HALT, STOP, SNOOZE mode
- Power supply voltage: V_{DD} = 2.7 to 5.5 V
- Operating ambient temperature: J grade products T_A = −40 to +85°C,
  L grade products T_A = −40 to +105°C

Remark The functions mounted depend on the product. See **1.7 Outline of Functions**.

Table 1-1. ROM, RAM capacities

Flash ROM	Data flash	RAM	48-pin		64-pin		80-pin	
512 KB	8 KB	24 KB	–	–	–	–	–	–
384 KB		20 KB	–	–	–	–	–	–
256 KB		16 KB	–	–	–	–	–	R5F10DMJxFB
128 KB		8 KB	–	–	–	–	–	R5F10DMGxFB
96 KB		6 KB	–	–	–	–	–	R5F10DMFxFB
64 KB		4 KB	–	R5F10DGExFB	–	R5F10DLExFB	R5F10CMExFB	R5F10DMExFB
48 KB		3 KB	R5F10CGDxFB	R5F10DGDxFB	R5F10CLDxFB	R5F10DLDxFB	R5F10CMDxFB	R5F10DMDxFB
32 KB		2 KB	R5F10CGCxFB	R5F10DGCxFB	–	–	–	–
24 KB		2 KB	R5F10CGBxFB	–	–	–	–	–
CAN (ch)			0	1	0	1	0	1
Stepper Motor (ch)			1		2		4	
LCD (seg × com)			27 × 4		39 × 4		48 × 4	

Flash ROM	Data flash	RAM	100-pin		128-pin
512 KB	8 KB	24 KB	–	R5F10DPLxFB	R5F10DSLxFB
384 KB		20 KB	–	R5F10DPKxFB	R5F10DSKxFB
256 KB		16 KB	R5F10TPJxFB	R5F10DPJxFB	R5F10DSJxFB
128 KB		8 KB	R5F10DPGxFB	–	–
96 KB		6 KB	R5F10DPFxFB	–	–
64 KB		4 KB	R5F10DPExFB	–	–
48 KB		3 KB	–	–	–
32 KB		2 KB	–	–	–
24 KB		2 KB	–	–	–
CAN (ch)			1	2	2
Stepper Motor (ch)			4		4
LCD (seg × com)			53 × 4		54 × 4

1.2 Applications

Automotive electrical appliances (instrument cluster)

<R> 1.3 Ordering Information

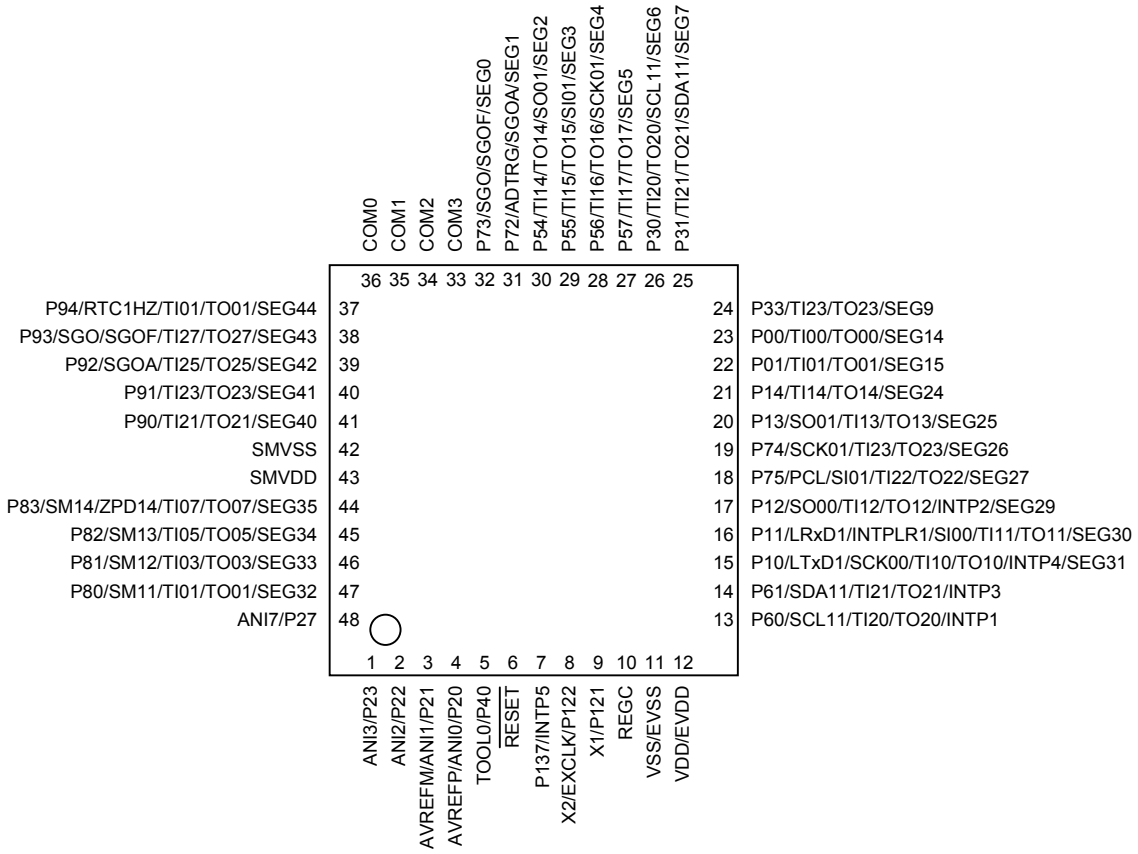
[List of Part Number]

Pin count	Package	Part Number	
		Operating ambient temperature J grade (TA = -40 to +85 °C)	Operating ambient temperature L grade (TA = -40 to +105 °C)
48-pin	48-pin plastic LQFP (fine pitch) (7 × 7)	R5F10CGBJFB R5F10CGCJFB R5F10CGDJFB R5F10DGCJFB R5F10DGDJFB R5F10DGEJFB	R5F10CGBLFB R5F10CGCLFB R5F10CGDLFB R5F10DGCLFB R5F10DGDLFB R5F10DGELFB
64-pin	64-pin plastic LQFP (fine pitch) (10 × 10)	R5F10CLDJFB R5F10DLDJFB R5F10DLEJFB	R5F10CLDLFB R5F10DLDLFB R5F10DLELFB
80-pin	80-pin plastic LQFP (fine pitch) (12 × 12)	R5F10CMDJFB R5F10CMEJFB R5F10DMDJFB R5F10DMEJFB R5F10DMFJFB R5F10DMGJFB R5F10DMJJFB	R5F10CMDLFB R5F10CMELFB R5F10DMDLFB R5F10DMELFB R5F10DMFLFB R5F10DMGLFB R5F10DMJLFB
100-pin	100-pin plastic LQFP (fine pitch) (14 × 14)	R5F10DPEJFB R5F10DPFJFB R5F10DPGJFB R5F10TPJJFB R5F10DPJJFB R5F10DPKJFB R5F10DPLJFB	R5F10DPELFB R5F10DPFLFB R5F10DPGLFB R5F10TPJLFB R5F10DPJLFB R5F10DPKLFB R5F10DPLLFB
128-pin	128-pin plastic LQFP (fine pitch) (14 × 20)	R5F10DSLJFB R5F10DSKJFB R5F10DSJJFB	R5F10DSLLFB R5F10DSKLFB R5F10DSJLFB

1.4 Pin Configuration (Top View)

1.4.1 48-pin products (R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB: with no CAN)

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

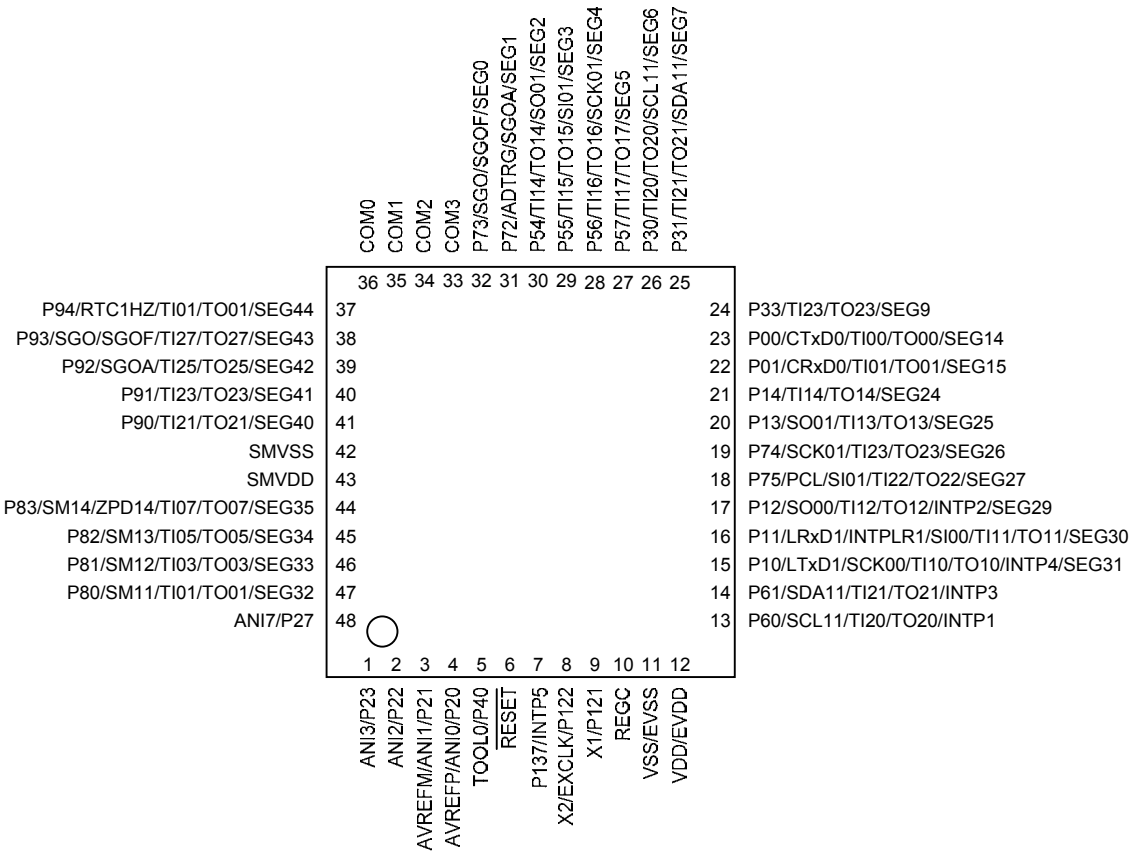


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

1.4.2 48-pin products (R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB: with CAN)

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

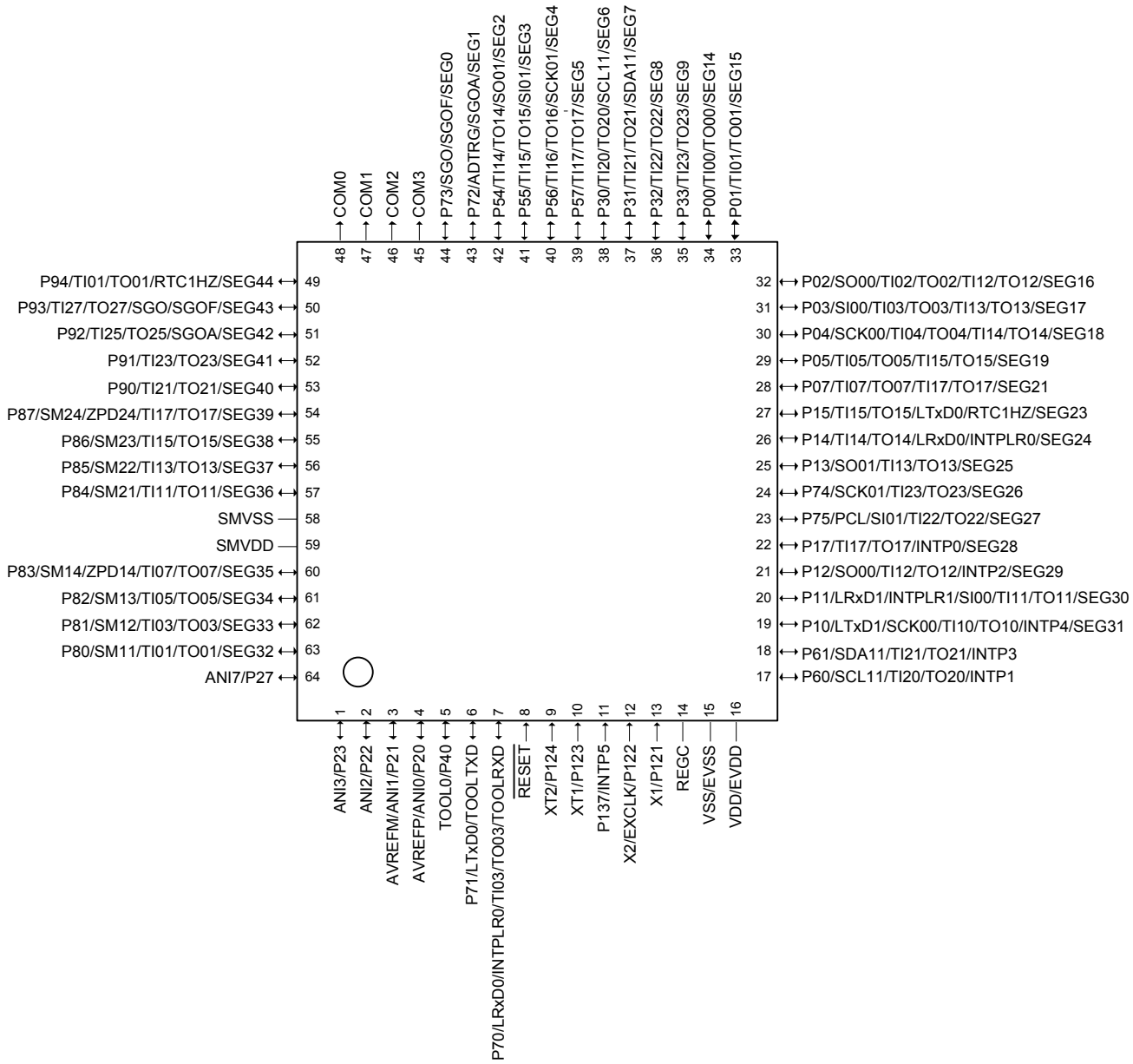


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

1.4.3 64-pin products (R5F10CLDxFB: with no CAN)

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

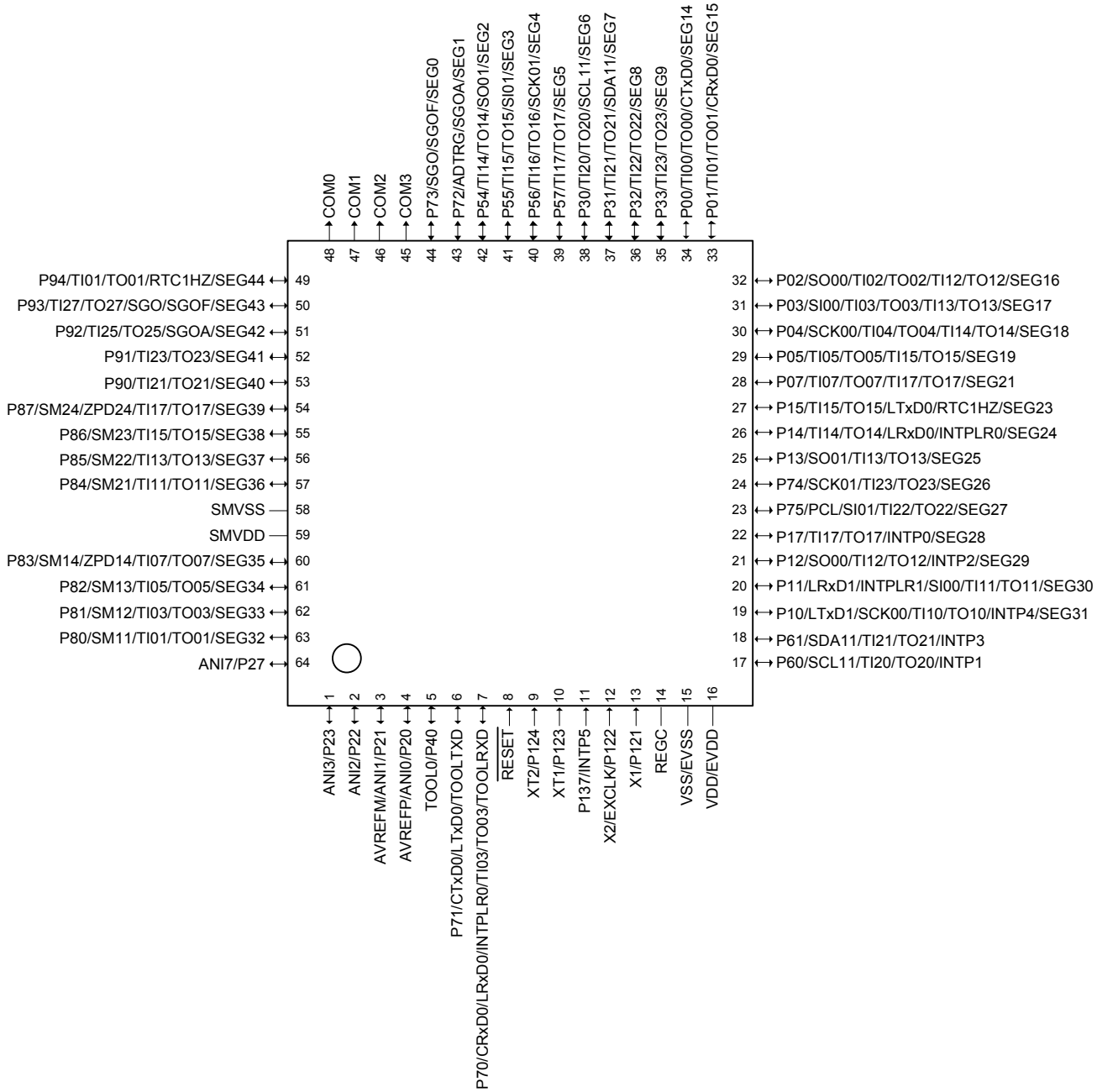


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

1.4.4 64-pin products (R5F10DLDxFB, R5F10DLExFB: with CAN)

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

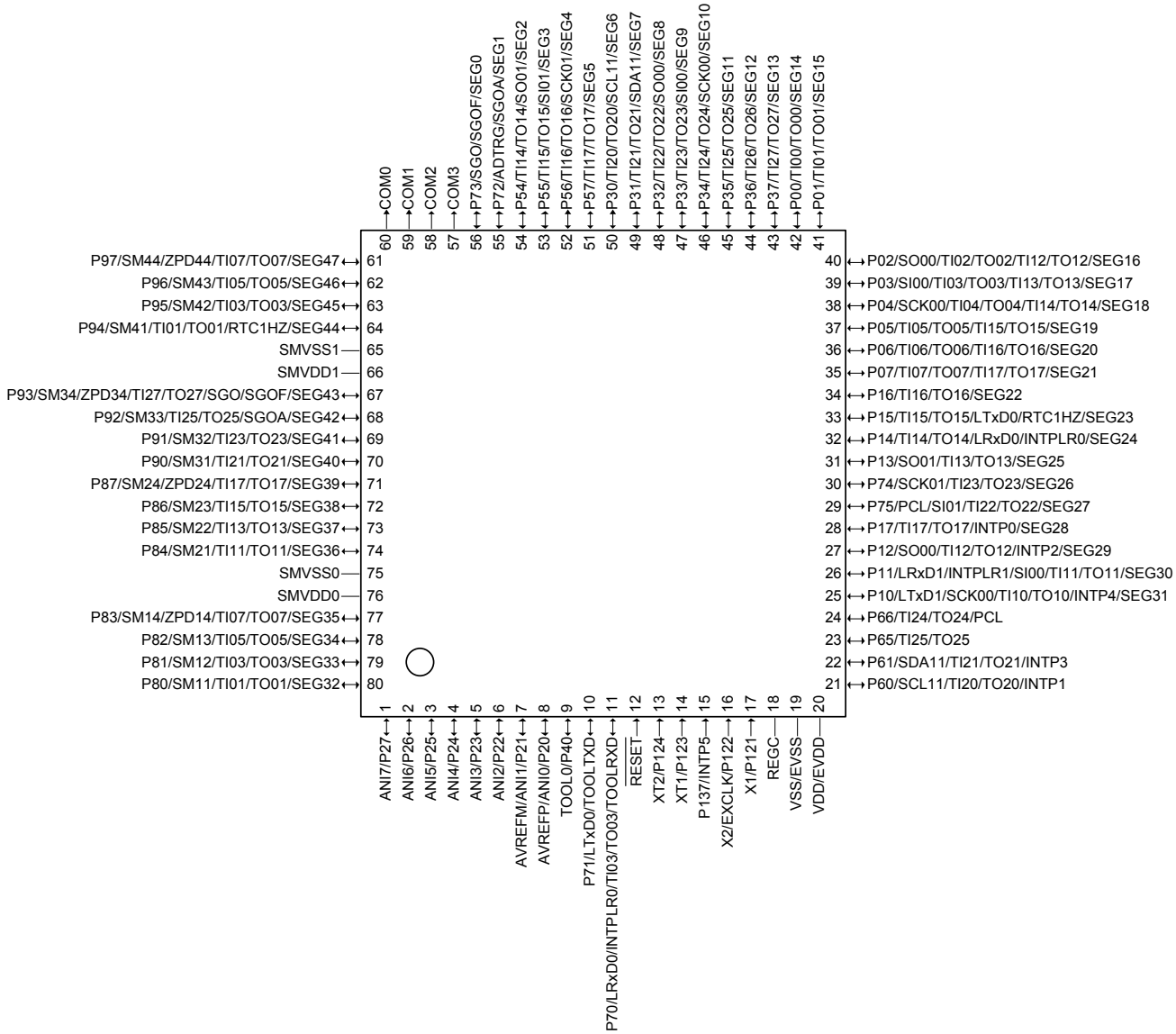


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

1.4.5 80-pin products (R5F10CMDxFB, R5F10CMExFB: with no CAN)

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

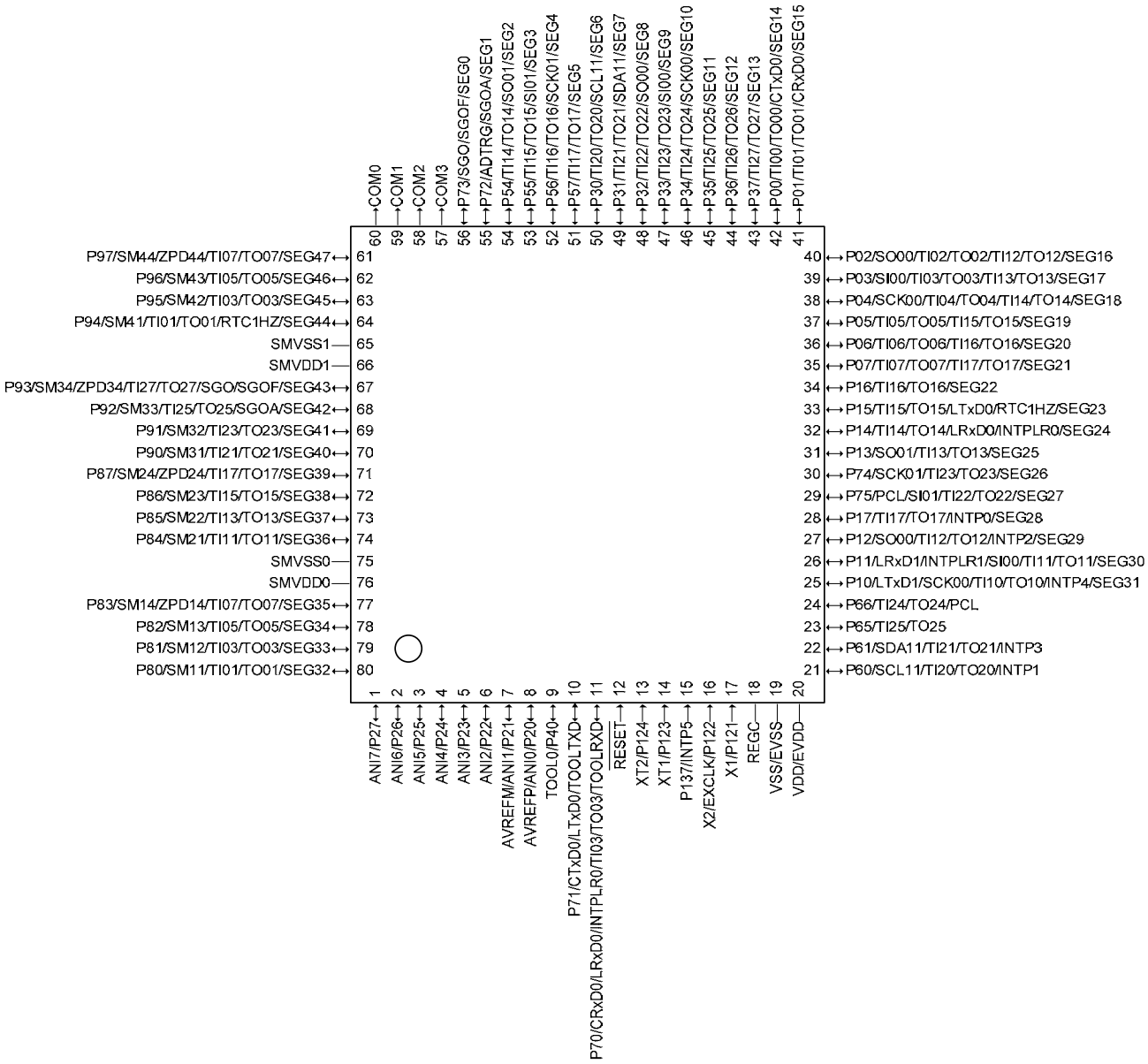


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

1.4.6 80-pin products (R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB: with CAN)

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

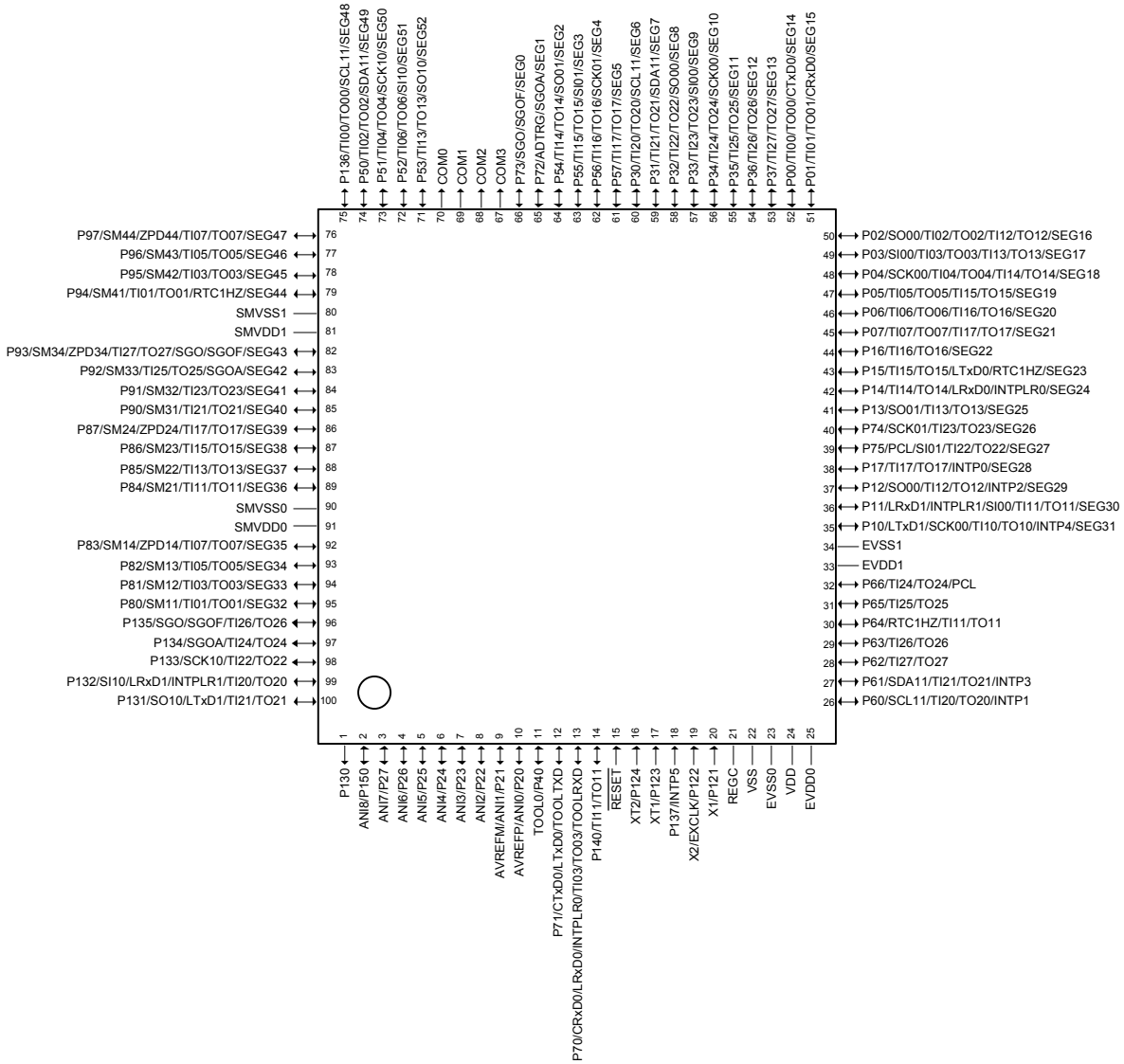


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

1.4.7 100-pin products (R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB: with 1 ch of CAN)

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

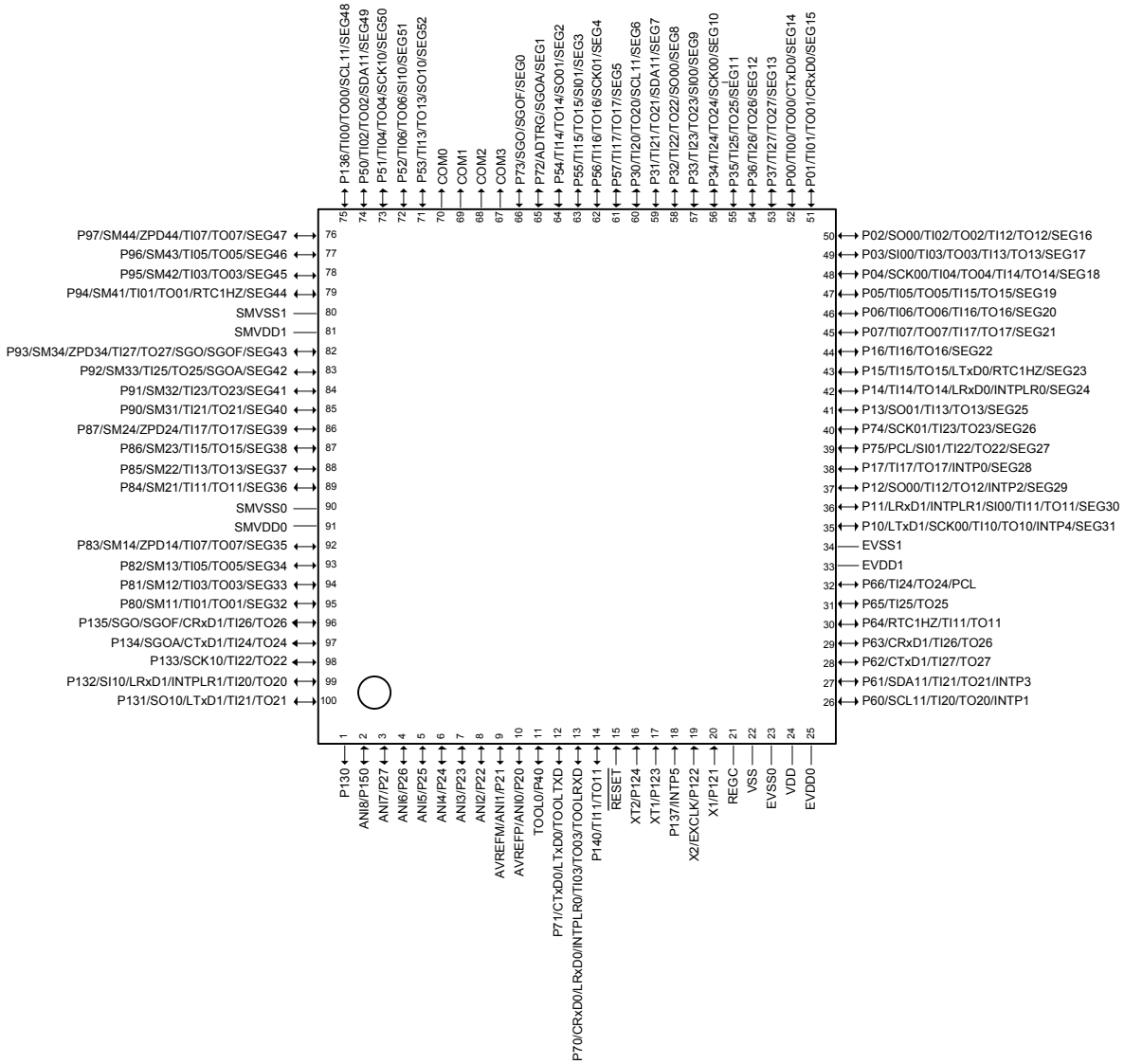


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

<R> 1.4.8 100-pin products (R5F10DPJxFB, R5F10DPKxFB, R5F10DPLxFB: with 2 ch of CAN)

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

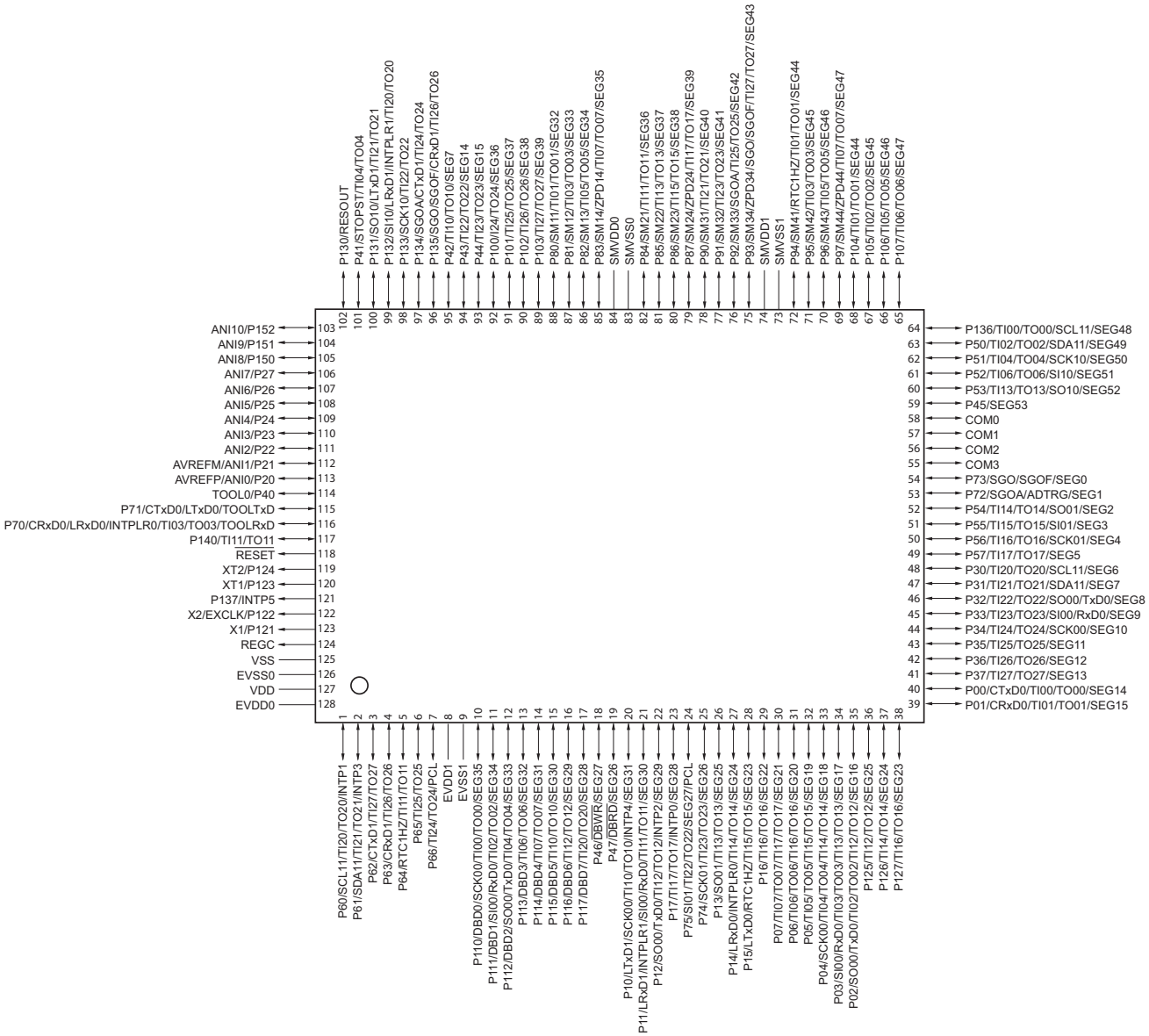


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF)

Remark For pin identification, see 1.5 Pin Identification.

<R> 1.4.9 128-pin products (R5F10DSLxxFB, R5F10DSKxxFB, R5F10DSJxxFB)

128-pin plastic LFQFP (fine pitch) (14 × 20)



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

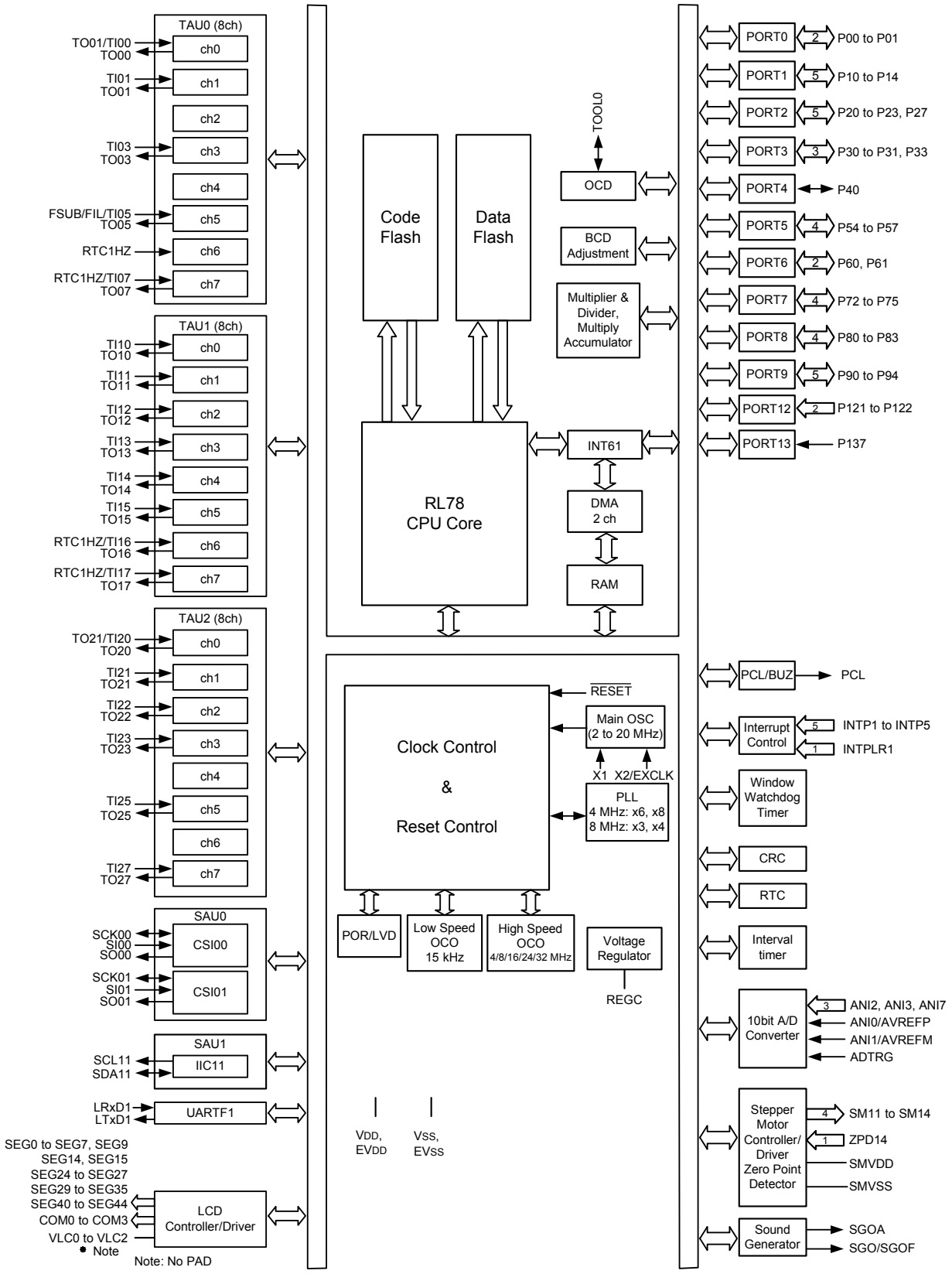
Remark For pin identification, see 1.5 Pin Identification.

<R> 1.5 Pin Identification

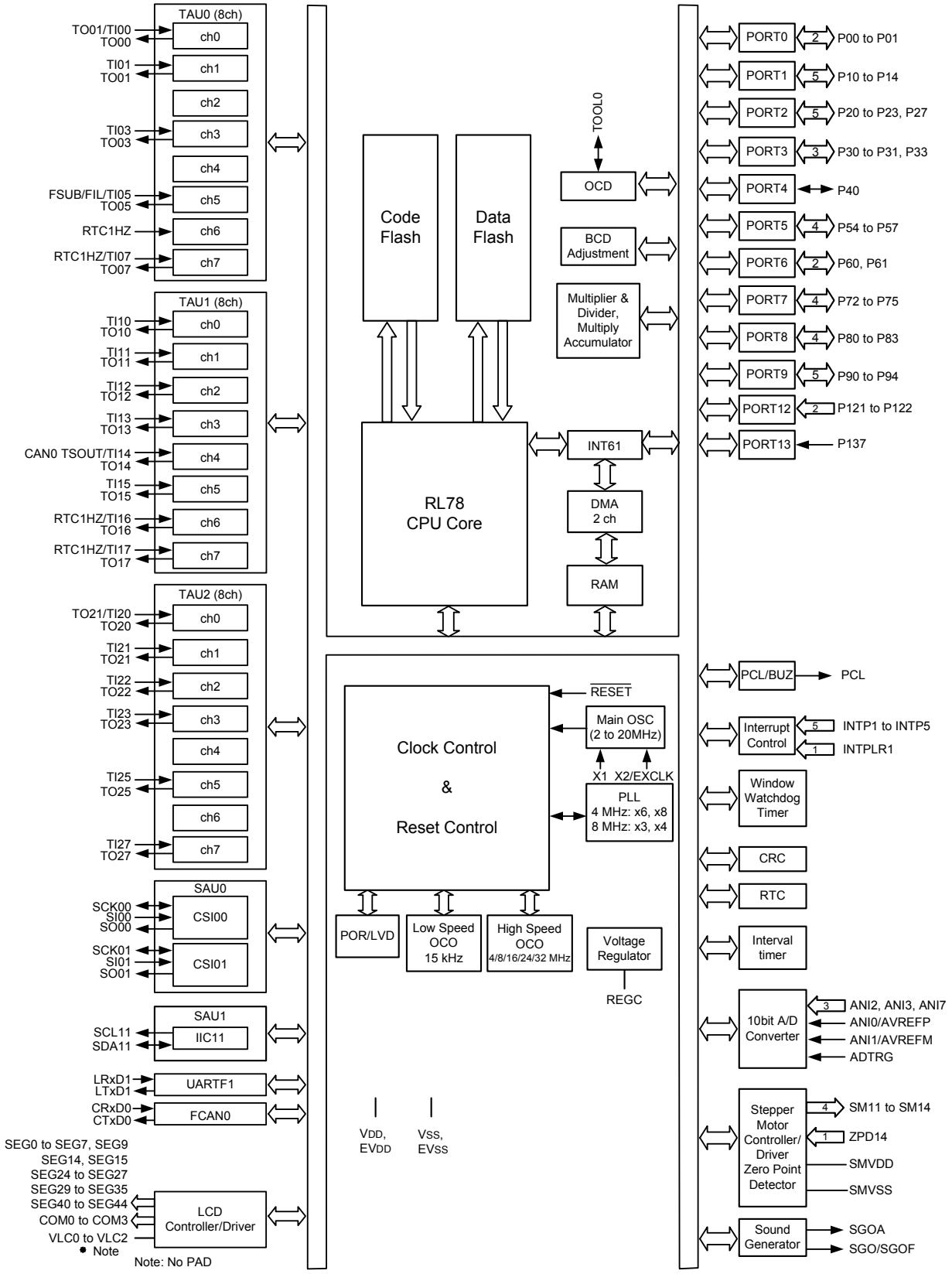
ANI0 to ANI10: Analog input
ADTRG: A/D conversion start trigger external input
AV_{REFM}: A/D converter reference potential (− side) input
AV_{REFP}: A/D converter reference potential (+ side) input
COM0 to COM3: Common output
CRxD0, CRxD1: Receive data for CAN
CTxD0, CTxD1: Transmit data for CAN
DBD0 to DBD7: LCD Bus I/F data lines
$\overline{DBWR}$: LCD Bus I/F write strobe
$\overline{DBRD}$: LCD Bus I/F read strobe
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)
INTP0 to INTP5: External interrupt input
INTPLR0, INTPLR1: External interrupt input for LIN
LRxD0, LRxD1: Serial data input to LIN
LTxD0, LTxD1: Serial data output from LIN
P00 to P07: Port 0
P10 to P17: Port 1
P20 to P27: Port 2
P30 to P37: Port 3
P40: Port 4
P50 to P57: Port 5
P60 to P66: Port 6
P70 to P75: Port 7
P80 to P87: Port 8
P90 to P97: Port 9
P121 to P124: Port 12
P130 to P137: Port 13
P140: Port 14
P150 to P152: Port 15
PCL: Programmable clock output
REGC: Regulator capacitance
$\overline{RESET}$: Reset
RESOUT: Reset output signal
RTC1HZ: Real-time clock correction clock (1 Hz) output
RxD0: Receive data for UART
SCK00, SCK01, SCK10: Serial clock input/output
SCL11: Serial clock input/output
SDA11: Serial data input/output
SEG0 to SEG53: Segment output
SGO: Sound generator output
SGOA: Sound generator amplitude PWM output
SGOF: Sound generator frequency output
SI00, SI01, SI10: Serial data input
SM11 to SM14, SM21 to SM24, SM31 to SM34, SM41 to SM44: Stepper motor outputs
SMV_{DD}, SMV_{DD0}, SMV_{DD1}: Stepper motor controller/driver supply voltage
SMV_{SS}, SMV_{SS0}, SMV_{SS1}: Stepper motor controller/driver ground
SO00, SO01, SO10: Serial data output
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
TOOLRxD, TOOLTxD: Data input/output for external device
TxD0: Transmit data for UART
V_{DD}: Power supply
V_{SS}: Ground
X1, X2: Crystal oscillator (Main system clock)
XT1, XT2: Crystal oscillator (Sub system clock)
ZPD14, ZPD24, ZPD34, ZPD44: Zero point detection input

1.6 Block Diagram

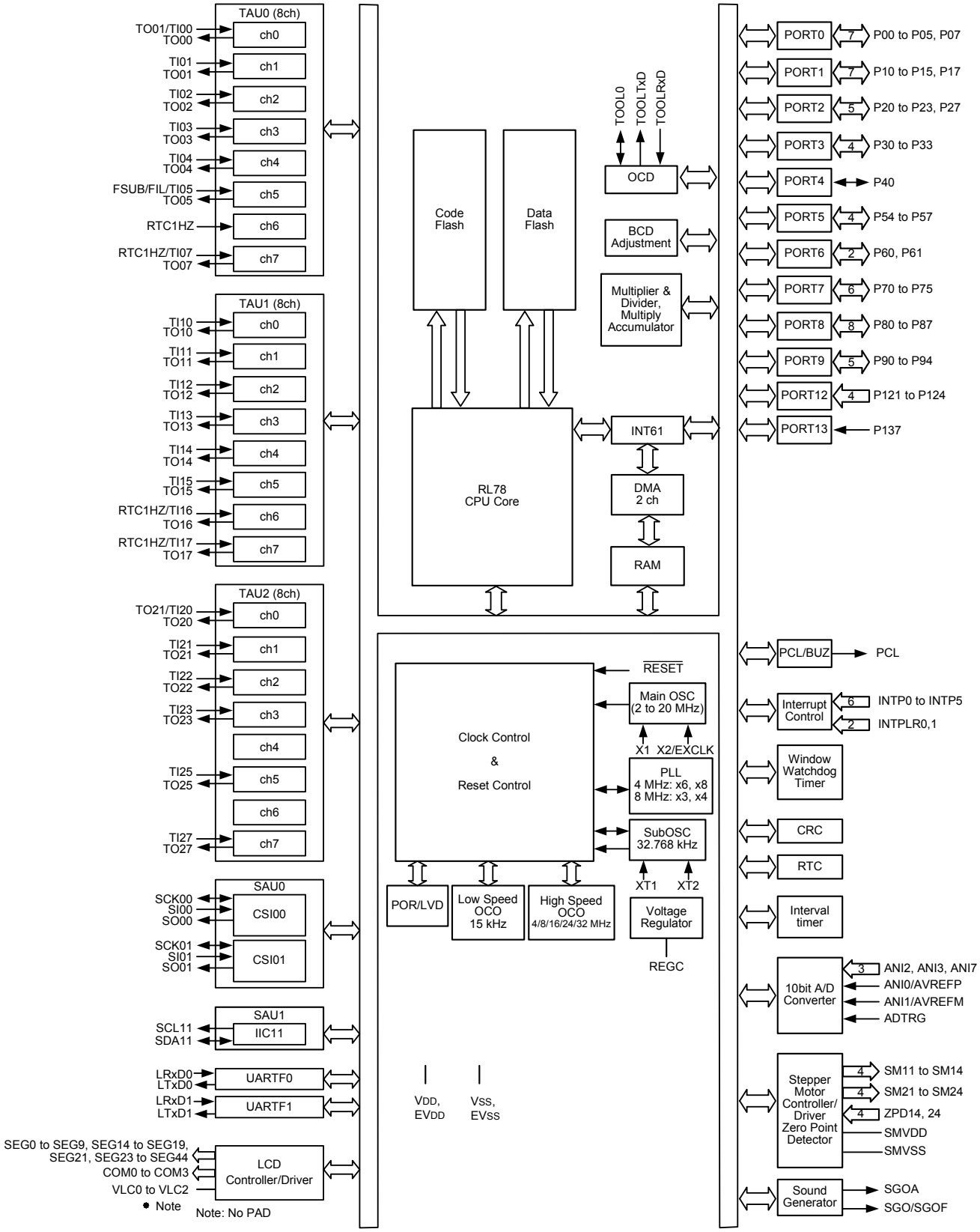
1.6.1 48-pin products (R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB: with no CAN)



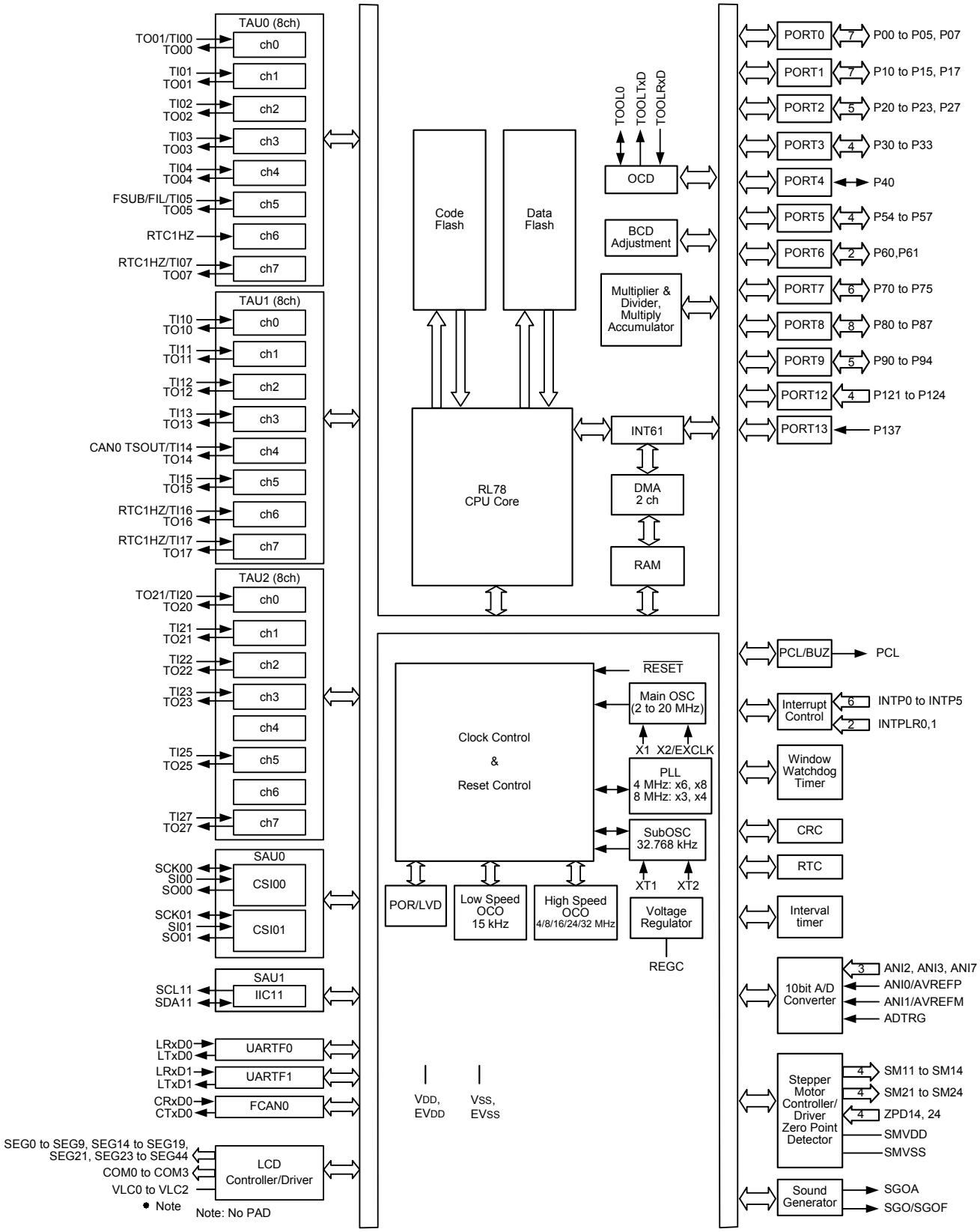
1.6.2 48-pin products (R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB: with CAN)



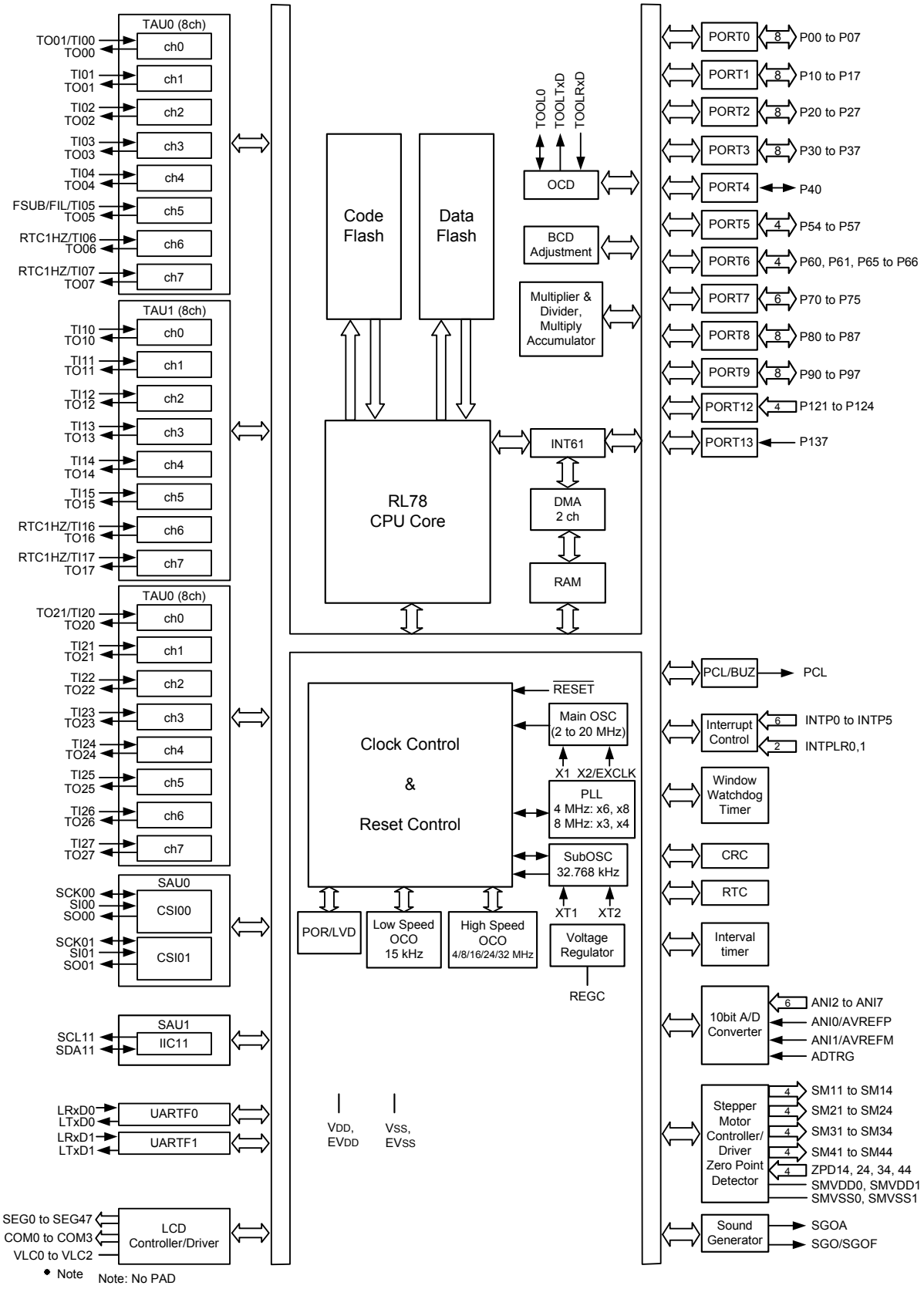
1.6.3 64-pin products (R5F10CLDxFB: with no CAN)



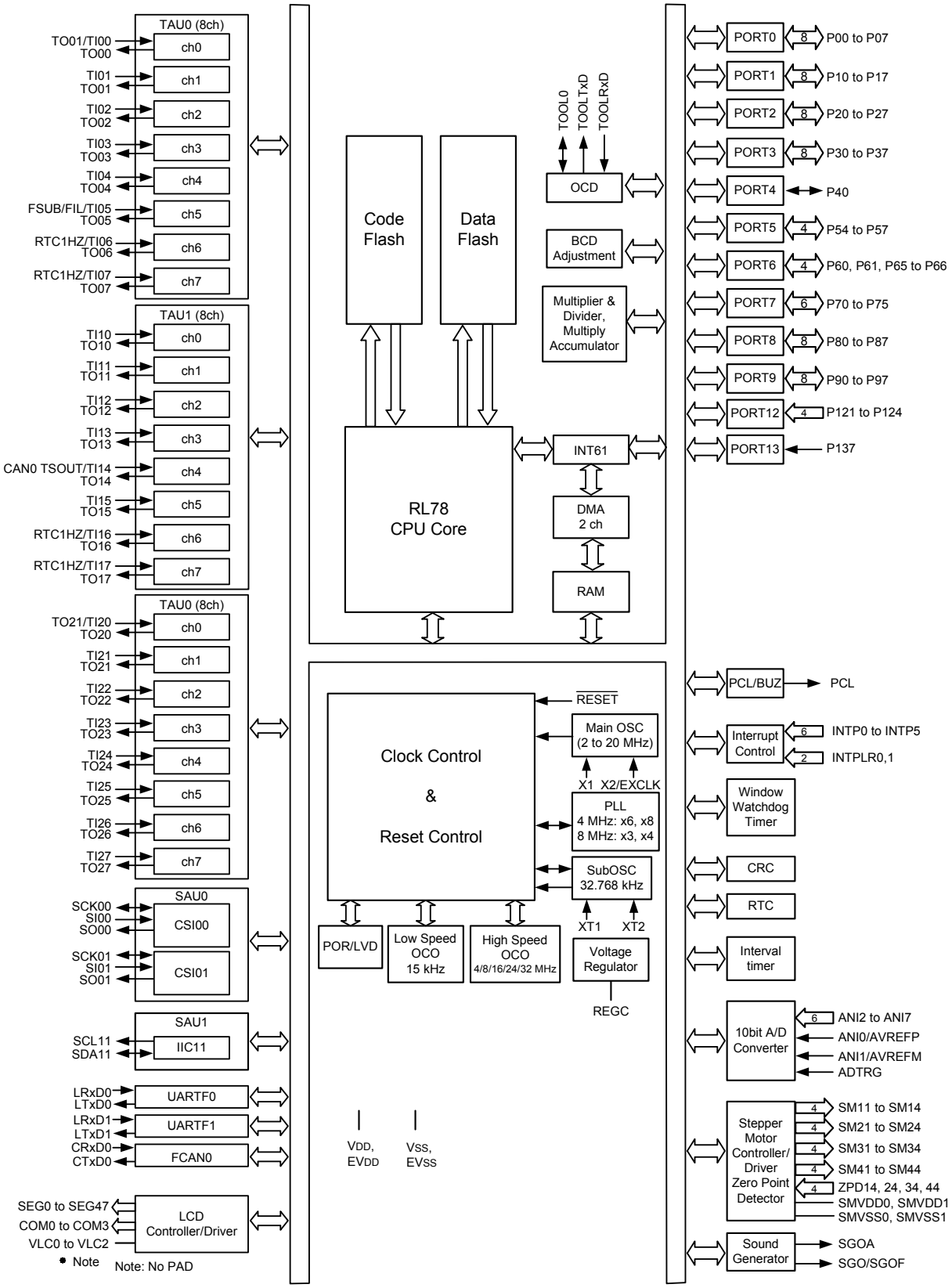
1.6.4 64-pin products (R5F10DLDxFB, R5F10DLExFB: with CAN)



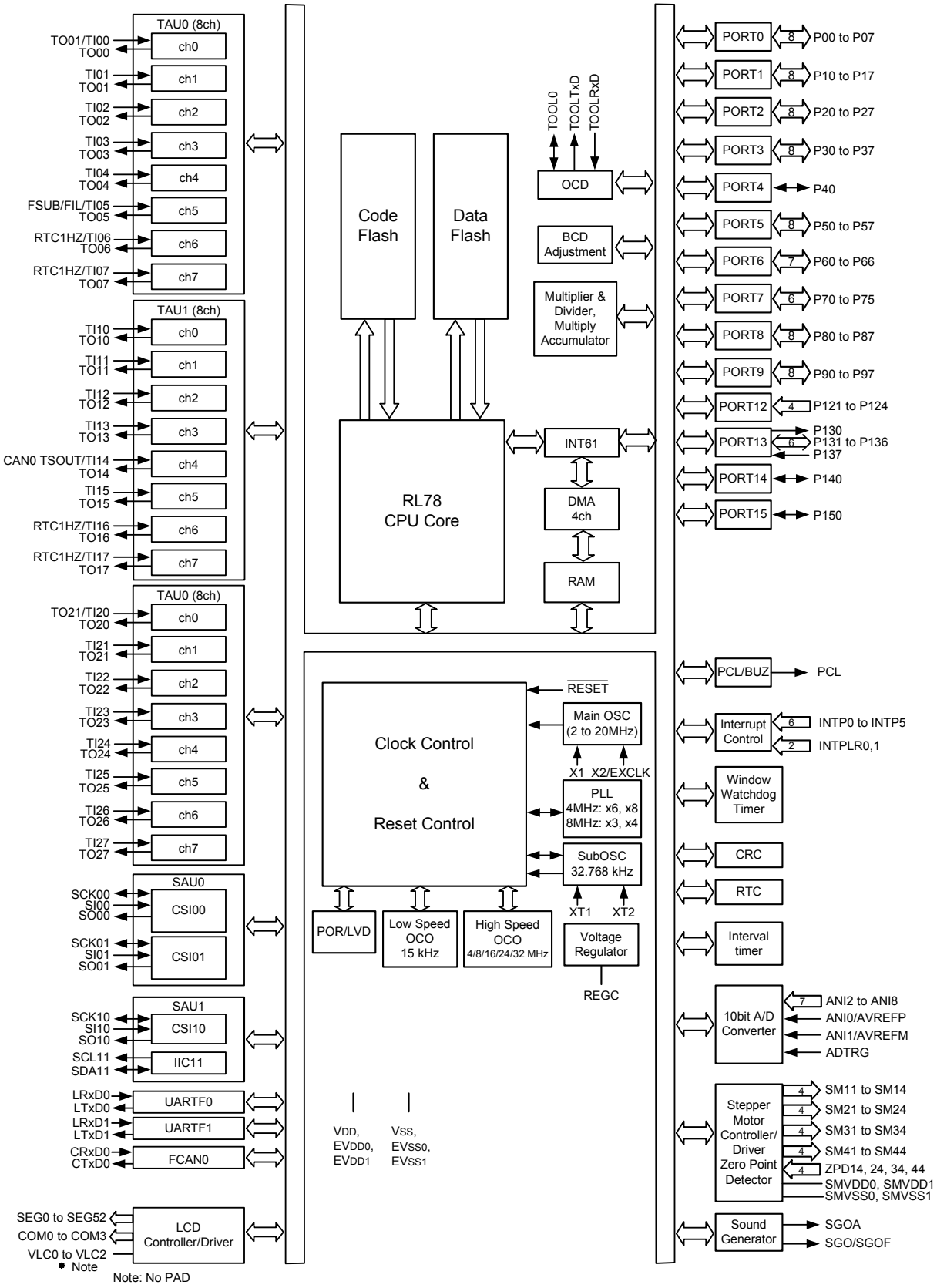
1.6.5 80-pin products (R5F10CMDxFB, R5F10CMExFB: with no CAN)



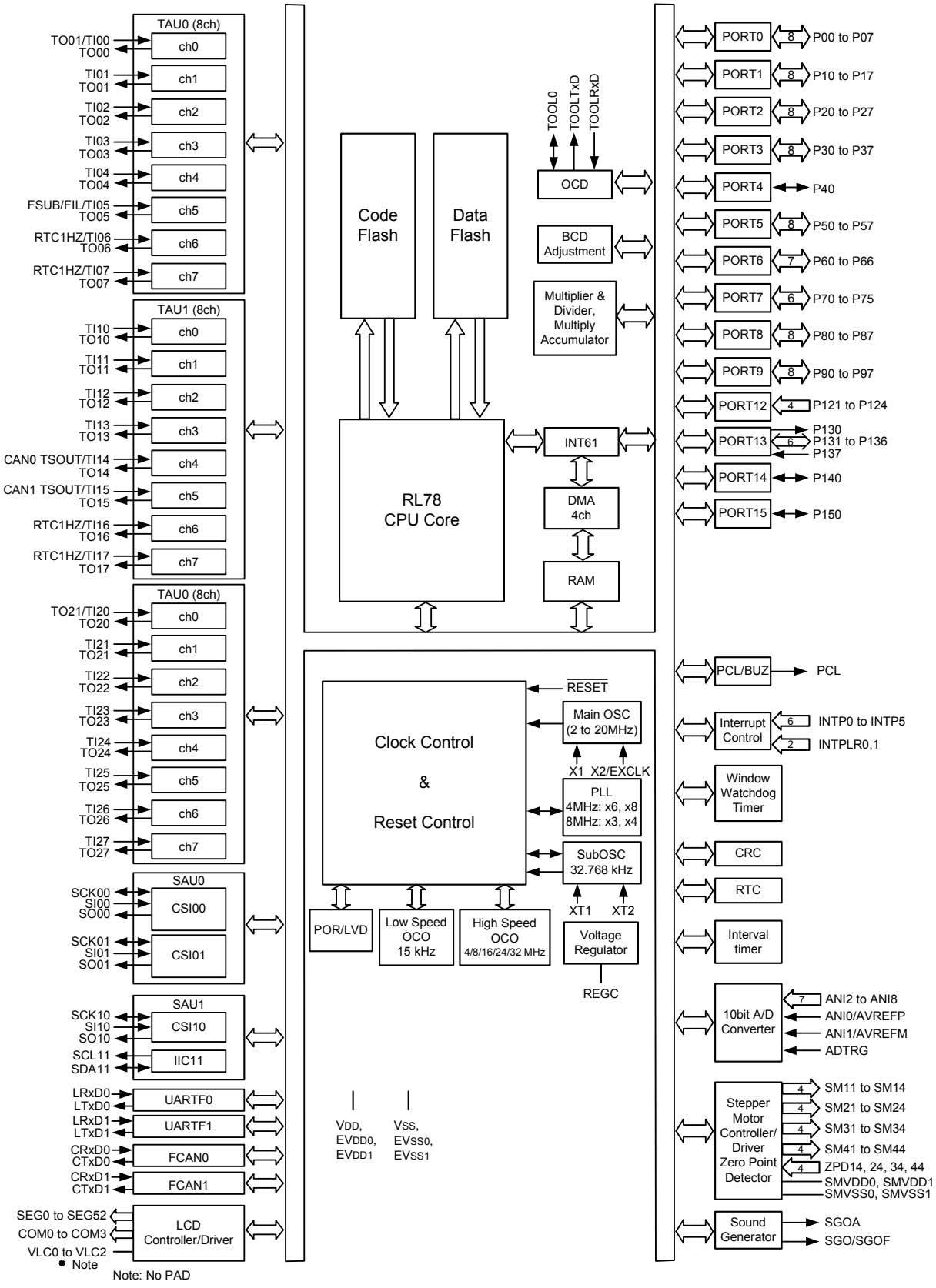
1.6.6 80-pin products (R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB: with CAN)



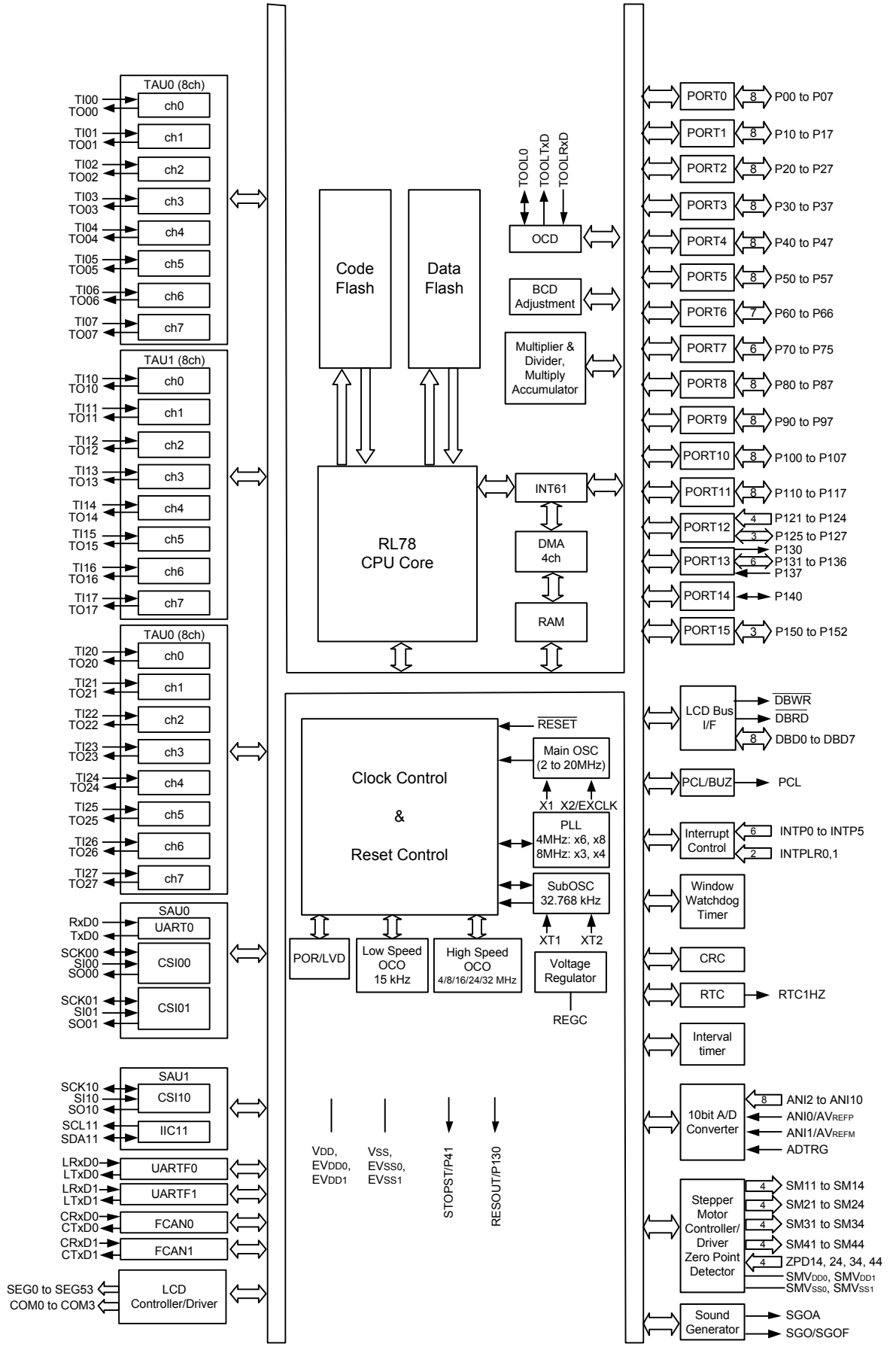
1.6.7 100-pin products (R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB: with 1 ch of CAN))



<R> 1.6.8 100-pin products (R5F10DPJxFB, R5F10DPKxFB, R5F10DPLxFB: with 2 ch of CAN)



<R> 1.6.9 128-pin products (R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB)



<R> 1.7 Outline of Functions

(1/4)

Item			48-pin						64-pin			80-pin						
	ROM	RAM	R5F10CGBxFB	R5F10CGCxFB	R5F10CGDxFB	R5F10DGCxFB	R5F10DGDxFB	R5F10DGExFB	R5F10CLDxFB	R5F10DLDxFB	R5F10DLExFB	R5F10CMDxFB	R5F10CMExFB	R5F10DMDxFB	R5F10DMExFB	R5F10DMFxFB	R5F10DMGxFB	R5F10DMJxFB
ROM/ RAM capacities	512 KB	24 KB																
	384 KB	20 KB																
	256 KB	16 KB																√
	128 KB	8 KB															√	
	96 KB	6 KB														√		
	64 KB	4 KB						√			√		√		√			
	48 KB	3 KB			√		√		√	√		√		√				
	32 KB	2 KB		√		√												
	24 KB	2 KB	√															
Data flash memory			8 KB															
Memory space			1 MB															
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)															
Main System clock	High-speed system clock		1 to 20 MHz (VDD = 2.7 V to 5.5 V)															
	High-speed on-chip oscillation clock		4/8/16/24/32 MHz (T_A = -40 to 85°C) 4/8/16/24 MHz (T_A = -40 to 105°C)															
PLL			4 MHz × 16/2 = 32 MHz, 8 MHz × 16/4 = 32 MHz (T_A= -40 to 85°C) 4 MHz × 12/2 = 24 MHz, 8 MHz × 12/4 = 24 MHz (T_A= -40 to 105°C)															
Subsystem clock			32.768kHz															
Low-speed on-chip oscillation clock			15 kHz															
Minimum instruction execution time			0.03125 μs (Main system clock 32 MHz T_A = -40 to +85°C) 0.04167 μs (Main system clock 24 MHz T_A = -40 to 105°C) 30.5 μs (Subsystem clock 32.768 kHz operation)															
Instruction set			• 8-bit operation,16-bit operation • Multiplication (8 bits × 8 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.															
I/O port	Total number of port		38						54			68						
	CMOS I/O port		35						49			63						
		N-ch open-drain Selectable port	4															
		LED direct drive port	9						13			16						
	CMOS input port		3						5									
	CMOS output port		0															
Timer	16-bit timer		8 ch × 3 units															
	Real-time clock (RTC)		1 ch															
	Interval timer		1 ch															
	Watchdog timer (WDT)		1 ch															
	Timer output		19						21			24						
	RTC output		1															
Clock output/buzzer output			1															
10-bit resolution A/D converter			3+2									6+2						

(2/4)

Item		48-pin						64-pin			80-pin						
		R5F10CGBxFB	R5F10CGCxFB	R5F10CGDxFB	R5F10DGCxFB	R5F10DGDxFB	R5F10DGExFB	R5F10CLDxFB	R5F10DLDxFB	R5F10DLExFB	R5F10CMDxFB	R5F10CMExFB	R5F10DMDxFB	R5F10DMExFB	R5F10DMFxFB	R5F10DMGxFB	R5F10DMJxFB
Serial interface	CSI	2 ch															
	UART	–															
	Simplified IIC	1 ch															
	LIN-UART	1 ch						2 ch									
	aFCAN	0 ch			1 ch			0 ch		1 ch	0 ch			1 ch			
Multiplier and divider/multiply accumulator		• 16-bits × 16 bits = 32 bits (Unsigned or signed) • 32-bits ÷ 32 bits = 32 bits (Unsigned) • 16-bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)															
DMA controller		2 ch															
Vectored interrupt sources	Internal	39			43			42		46	42			46			
	External	6						8									
	Software	1															
	Debugger	1															
LCD controller driver	Bias	Static, 1/3 bias, 1/3 or 1/4 duty															
	SEG × COM	27 × 4						39 × 4			48 × 4						
Sound generator		1 channel															
Stepper motor controller/driver (with ZPD)		1 ch						2 ch			4 ch						
Safety function	FLASH memory CRC calculation	Provided															
	RAM parity bit error detection	Provided															
	Illegal-memory access detection	Provided															
	Frequency detection	Provided															
	Clock monitor	Provided															
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution • Internal reset by RAM parity error • Internal reset by illegal-memory access • Internal reset by clock monitor															
Power on reset (POR)		Power on reset: 1.51 V ± 0.06 V Power down reset: 1.50 V ± 0.06 V															
Voltage detector		Rising edge detection voltage = 2.81 to 4.06 V (6 step) Falling edge detection voltage = 2.75 to 3.98 V (6 step)															
On-chip debug function		Provided															
Power supply voltage		V_{DD} = 2.7 to 5.5 V															
Operating ambient temperature		J grade products:T_A = -40 to +85 °C, L grade products: T_A = -40 to +105 °C															

(3/4)

Item			100-pin							128-pin		
	ROM	RAM	R5F10DPExFB	R5F10DPFxFB	R5F10DPGxFB	R5F10TPJxFB	R5F10DPJxFB	R5F10DPKxFB	R5F10DPLxFB	R5F10DSJxFB	R5F10DSKxFB	R5F10DSLxFB
ROM/ RAM capacities	512 KB	24 KB							√			√
	384 KB	20 KB						√			√	
	256 KB	16 KB				√	√			√		
	128 KB	8 KB			√							
	96 KB	6 KB		√								
	64 KB	4 KB	√									
	48 KB	3 KB										
	32 KB	2 KB										
	24 KB	2 KB										
Data flash memory			8 KB									
Memory space			1 MB									
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)									
Main System clock	High-speed system clock		1 to 20 MHz (VDD = 2.7 V to 5.5 V)									
	High-speed on-chip oscillation clock		4/8/16/24/32 MHz (T_A = -40 to 85°C) 4/8/16/24 MHz (T_A = -40 to 105°C)									
PLL			4 MHz × 16/2 = 32 MHz, 8 MHz × 16/4 = 32 MHz (T_A= -40 to 85°C) 4 MHz × 12/2 = 24 MHz, 8 MHz × 12/4 = 24 MHz (T_A= -40 to 105°C)									
Subsystem clock			32.768kHz									
Low-speed on-chip oscillation clock			15 kHz									
Minimum instruction execution time			0.03125 μs (Main system clock 32 MHz T_A = -40 to +85°C) 0.04167 μs (Main system clock 24 MHz T_A = -40 to 105°C) 30.5 μs (Subsystem clock 32.768 kHz operation)									
Instruction set			• 8-bit operation,16-bit operation • Multiplication (8 bits × 8 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.									
I/O port	Total number of port		84							112		
	CMOS I/O port		78							107		
		N-ch open-drain Selectable port	6									
		LED direct drive port	16									
	CMOS input port		5									
	CMOS output port		1									
Timer	16-bit timer		8 ch × 3 units									
	Real-time clock (RTC)		1 ch									
	Interval timer		1 ch									
	Watchdog timer (WDT)		1 ch									
	Timer output		24									
	RTC output		1									
Clock output/buzzer output			1									
10-bit resolution A/D converter			7+2							9+2		
Reset output			Can be output from P130									
STOP status output			Can be output from P41									

(4/4)

Item		100-pin							128-pin		
		R5F10DPExFB	R5F10DPFxFB	R5F10DPGxFB	R5F10TPJxFB	R5F10DPJxFB	R5F10DPKxFB	R5F10DPLxFB	R5F10DSJxFB	R5F10DSKxFB	R5F10DSLxFB
Serial interface	CSI	3 ch									
	UART	–							1 ch		
	Simplified IIC	1 ch									
	LIN-UART	2 ch									
	aFCAN	1 ch				2 ch					
Multiplier and divider/multiply accumulator		• 16-bits × 16 bits = 32 bits (Unsigned or signed) • 32-bits ÷ 32 bits = 32 bits (Unsigned) • 16-bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)									
DMA controller		4 ch									
Vectored interrupt sources	Internal	53									
	External	8									
	Software	1									
	Debugger	1									
LCD controller driver	Bias	Static, 1/3 bias, 1/3 or 1/4 duty									
	SEG × COM	53 × 4							54 × 4		
LCD Bus I/F		–							Provided (8 bit, RD, WR)		
Sound generator		1 channel									
Stepper motor controller/driver (with ZPD)		4 ch									
Safety function	FLASH memory CRC calculation	Provided									
	RAM parity bit error detection	Provided									
	Illegal-memory access detection	Provided									
	Frequency detection	Provided									
	Clock monitor	Provided									
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution • Internal reset by RAM parity error • Internal reset by illegal-memory access • Internal reset by clock monitor									
Power on reset (POR)		Power on reset: 1.51 V ± 0.06 V Power down reset: 1.50 V ± 0.06 V									
Voltage detector		Rising edge detection voltage = 2.81 to 4.06 V (6 step) Falling edge detection voltage = 2.75 to 3.98 V (6 step)									
On-chip debug function		Provided									
Power supply voltage		V_{DD} = 2.7 to 5.5 V									
Operating ambient temperature		J grade products:T_A = -40 to +85 °C, L grade products: T_A = -40 to +105 °C									

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

(1) 48-pin products

Power Supply	Corresponding Pins
V_{DD}/EV_{DD}	Port pins other than P80 to P83 and P90 to P94
SMV_{DD}	P80 to P83, P90 to P94

(2) 64-pin products

Power Supply	Corresponding Pins
V_{DD}/EV_{DD}	Port pins other than P80 to P87and P90 to P94
SMV_{DD}	P80 to P87, P90 to P94

(3) 80-pin products

Power Supply	Corresponding Pins
V_{DD}/EV_{DD}	Port pins other than P80 to P87 and P90 to P97
SMV_{DD0}, SMV_{DD1}	P80 to P87, P90 to P97

(4) 100-pin products

Power Supply	Corresponding Pins
V_{DD}	P20 to P27, P150, P137, P121 to P124, $\overline{\text{RESET}}$
EV_{DD0}, EV_{DD1}	P00 to P07, P10 to P17, P30 to P37, P40, P50 to P57, P60 to P66, P70 to P75, P130 to P136, P140
SMV_{DD0}, SMV_{DD1}	P80 to P87, P90 to P97

<R> (5) 128-pin products

Power Supply	Corresponding Pins
V_{DD}	P20 to P27, P150 to P152, P137, P121 to P124, $\overline{\text{RESET}}$
EV_{DD0}, EV_{DD1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P66, P70 to P75, P100 to P107, P110 to P117, P125 to P127, P130 to P136, and P140
SMV_{DD0}, SMV_{DD1}	P80 to P87, P90 to P97

The setting of I/O, buffer and pull-up resistor in each port is also valid for alternate functions.

2.1.1 48-pin products

Table 2-2. Port Pins for R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB, R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB (1/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI00/TO00/CTxD0/SEG14 Note1
P01					TI01/TO01/CRxD0/SEG15 Note1
P10	I/O	Port 1. 5-bit I/O port. Input of P10 and P11 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	LTxD1/SCK00/TI10/TO10/INTP4/SEG31
P11					LRxD1/INTPLR1/SI00/TI11/TO11/SEG30
P12					SO00/TI12/TO12/INTP2/SEG29
P13					SO01/TI13/TO13/SEG25
P14					TI14/TO14/SEG24
P20	I/O	Port 2. 5-bit I/O port. Can be set to analog input Note2 Input/output can be specified in 1-bit units.	HZ	Analog Input	AVREFP/ANI0
P21					AVREFM/ANI1
P22					ANI2
P23					ANI3
P27					ANI7
P30	I/O	Port 3. 3-bit I/O port. Input of P31 can be set to schmitt 1 input buffer. Output of P30 and P31 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI20/TO20/SCL11/SEG6
P31					TI21/TO21/SDA11/SEG7
P33					TI23/TO23/SEG9
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: HZ Int.: PU	Input	TOOL0
P54	I/O	Port 5. 4-bit I/O port. Input of P55 to P57 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI14/TO14/SO01/SEG2
P55					TI15/TO15/SI01/SEG3
P56					TI16/TO16/SCK01/SEG4
P57					TI17/TO17/SEG5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset, PD: Pull down, PU: Pull up, HZ: High impedance

Notes 1. CTxD0 and CRxD0 are not provided for R5F10CGDxFB, R5F10CGCxFB and R5F10CGBxFB with no CAN channel.

2. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Table 2-2. Port Pins for
R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB, R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB (2/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P60	I/O	Port 6. 2-bit I/O port. Input of P61 can be set to schmitt 1 input buffer. Output of P60 and P61 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	SCL11/TI20/TO20/INTP1
P61					SDA11/TI21/TO21/INTP3
P72	I/O	Port 7. 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.	Ext.: PD Int.: HZ	Input	ADTRG/SGOA/SEG1
P73					SGO/SGOF/SEG0
P74					SCK01/TI23/TO23/SEG26
P75					PCL/SI01/TI22/TO22/SEG27
P80	I/O	Port 8. 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.	Ext.: PD Int.: HZ	Input	SM11/TI01/TO01/SEG32
P81					SM12/TI03/TO03/SEG33
P82					SM13/TI05/TO05/SEG34
P83					SM14/ZPD14/TI07/TO07/SEG35
P90	I/O	Port 9. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI21/TO21/SEG40
P91					TI23/TO23/SEG41
P92					TI25/TO25/SGOA/SEG42
P93					TI27/TO27/SGO/SGOF/SEG43
P94					TI01/TO01/RTC1HZ/SEG44
P121	I	Port 12. 2-bit Input port.	HZ	Input	X1
P122					X2/EXCLK
P137	I	Port 13. 1-bit Input port.	HZ	Input	INTP5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, HZ: High impedance

2.1.2 64-pin products

Table 2-3. Port Pins for R5F10CLDxFB, R5F10DLDxFB, R5F10DLExFB (1/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 7-bit I/O port. Input of P01 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI00/TO00/CTxD0/SEG14 Note1
P01					TI01/TO01/CRxD0/SEG15 Note1
P02					SO00/TI02/TO02/TI12/TO12/SEG16
P03					SI00/TI03/TO03/TI13/TO13/SEG17
P04					SCK00/TI04/TO04/TI14/TO14/SEG18
P05					TI05/TO05/TI15/TO15/SEG19
P07					TI07/TO07/TI17/TO17/SEG21
P10	I/O	Port 1. 7-bit I/O port. Input of P10, P11, and P17 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	LTxD1/SCK00/TI10/TO10/INTP4/SEG31
P11					LRxD1/INTPLR1/SI00/TI11/TO11/SEG30
P12					SO00/TI12/TO12/INTP2/SEG29
P13					SO01/TI13/TO13/SEG25
P14					TI14/TO14/LRxD0/INTPLR0/SEG24
P15					TI15/TO15/LTxD0/RTC1HZ/SEG23
P17					TI17/TO17/INTP0/SEG28
P20	I/O	Port 2. 5-bit I/O port. Can be set to analog input Note2 Input/output can be specified in 1-bit units.	HZ	Analog Input	AVREFP/ANI0
P21					AVREFM/ANI1
P22					ANI2
P23					ANI3
P27					ANI7
P30	I/O	Port 3. 4-bit I/O port. Input of P31 can be set to schmitt 1 input buffer. Output of P30 and P31 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI20/TO20/SCL11/SEG6
P31					TI21/TO21/SDA11/SEG7
P32					TI22/TO22/SEG8
P33					TI23/TO23/SEG9
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: HZ Int.: PU	Input	TOOL0
P54	I/O	Port 5. 4-bit I/O port. Input of P55 to P57 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI14/TO14/SO01/SEG2
P55					TI15/TO15/SI01/SEG3
P56					TI16/TO16/SCK01/SEG4
P57					TI17/TO17/SEG5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset, PD: Pull down, PU: Pull up, HZ: High impedance

Notes 1. CTxD0 and CRxD0 are not provided for R5F10CLDxFB with no CAN channel.

2. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Table 2-3. Port Pins for R5F10CLDxFB, R5F10DLDxFB, R5F10DLExFB (2/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P60	I/O	Port 6. 2-bit I/O port. Input of P61 can be set to schmitt 1 input buffer. Output of P60 and P61 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	SCL11/TI20/TO20/INTP1
P61					SDA11/TI21/TO21/INTP3
P70	I/O	Port 7. 6-bit I/O port. Input of P70 can be set to schmitt 1 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.	HZ	Input	CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD[Note]
P71					CTxD0/LTxD0/TOOLTXD[Note]
P72			Ext.: PD Int.: HZ		ADTRG/SGOA/SEG1
P73					SGO/SGOF/SEG0
P74					SCK01/TI23/TO23/SEG26
P75					PCL/SI01/TI22/TO22/SEG27
P80	I/O	Port 8. 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.	Ext.: PD Int.: HZ	Input	SM11/TI01/TO01/SEG32
P81					SM12/TI03/TO03/SEG33
P82					SM13/TI05/TO05/SEG34
P83					SM14/ZPD14/TI07/TO07/SEG35
P84					SM21/TI11/TO11/SEG36
P85					SM22/TI13/TO13/SEG37
P86					SM23/TI15/TO15/SEG38
P87					SM24/ZPD24/TI17/TO17/SEG39
P90	I/O	Port 9. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI21/TO21/SEG40
P91					TI23/TO23/SEG41
P92					TI25/TO25/SGOA/SEG42
P93					TI27/TO27/SGO/SGOF/SEG43
P94					TI01/TO01/RTC1HZ/SEG44
P121	I	Port 12. 4-bit Input port.	HZ	Input	X1
P122					X2/EXCLK
P123					XT1
P124					XT2
P137	I	Port 13. 1-bit Input port.	HZ	Input	INTP5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset, PD: Pull down, HZ: High impedance

Note CTxD0 and CRxD0 are not provided for R5F10CLDxFB with no CAN channel.

2.1.3 80-pin products products

Table 2-4. Port Pins for R5F10CMDxFB, R5F10CMExFB, R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB (1/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 8-bit I/O port. Input of P01 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI00/TO00/CTxD0/SEG14 Note1
P01					TI01/TO01/CRxD0/SEG15 Note1
P02					SO00/TI02/TO02/TI12/TO12/SEG16
P03					SI00/TI03/TO03/TI13/TO13/SEG17
P04					SCK00/TI04/TO04/TI14/TO14/SEG18
P05					TI05/TO05/TI15/TO15/SEG19
P06					TI06/TO06/TI16/TO16/SEG20
P07					TI07/TO07/TI17/TO17/SEG21
P10	I/O	Port 1. 8-bit I/O port. Input of P10, P11, and P17 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	LTxD1/SCK00/TI10/TO10/INTP4/SEG31
P11					LRxD1/INTPLR1/SI00/TI11/TO11/SEG30
P12					SO00/TI12/TO12/INTP2/SEG29
P13					SO01/TI13/TO13/SEG25
P14					TI14/TO14/LRxD0/INTPLR0/SEG24
P15					TI15/TO15/LTxD0/RTC1HZ/SEG23
P16					TI16/TO16/SEG22
P17					TI17/TO17/INTP0/SEG28
P20	I/O	Port 2. 8-bit I/O port. Can be set to analog input Note2 Input/output can be specified in 1-bit units.	HZ	Analog Input	AVREFP/ANI0
P21					AVREFM/ANI1
P22 to P27					ANI2 to ANI7
P30	I/O	Port 3. 8-bit I/O port. Input of P31 can be set to schmitt 1 input buffer. Output of P30 and P31 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI20/TO20/SCL11/SEG6
P31					TI21/TO21/SDA11/SEG7
P32					TI22/TO22/SO00/SEG8
P33					TI23/TO23/SI00/SEG9
P34					TI24/TO24/SCK00/SEG10
P35					TI25/TO25/SEG11
P36					TI26/TO26/SEG12
P37					TI27/TO27/SEG13
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: HZ Int.: PU	Input	TOOL0
P54	I/O	Port 5. 4-bit I/O port. Input of P55 to P57 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI14/TO14/SO01/SEG2
P55					TI15/TO15/SI01/SEG3
P56					TI16/TO16/SCK01/SEG4
P57					TI17/TO17/SEG5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset, PD: Pull down, PU: Pull up, HZ: High impedance

Notes 1. CTxD0 and CRxD0 are not provided for R5F10CMExFB and R5F10CMDxFB with no CAN channel.

2. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Table 2-4. Port Pins for R5F10CMDxFB, R5F10CMExFB, R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB (2/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P60	I/O	Port 6. 4-bit I/O port. Input of P61 can be set to schmitt 1 input buffer. Output of P60 and P61 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	SCL11/TI20/TO20/INTP1
P61					SDA11/TI21/TO21/INTP3
P65					TI25/TO25
P66					TI24/TO24/PCL
P70	I/O	Port 7. 6-bit I/O port. Input of P70 can be set to schmitt 1 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.	HZ	Input	CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD[Note]
P71					CTxD0/LTxD0/TOOLTXD[Note]
P72			Ext.: PD Int.: HZ		ADTRG/SGOA/SEG1
P73					SGO/SGOF/SEG0
P74					SCK01/TI23/TO23/SEG26
P75					PCL/SI01/TI22/TO22/SEG27
P80	I/O	Port 8. 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.	Ext.: PD Int.: HZ	Input	SM11/TI01/TO01/SEG32
P81					SM12/TI03/TO03/SEG33
P82					SM13/TI05/TO05/SEG34
P83					SM14/ZPD14/TI07/TO07/SEG35
P84					SM21/TI11/TO11/SEG36
P85					SM22/TI13/TO13/SEG37
P86					SM23/TI15/TO15/SEG38
P87					SM24/ZPD24/TI17/TO17/SEG39
P90	I/O	Port 9. 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.	Ext.: PD Int.: HZ	Input	SM31/TI21/TO21/SEG40
P91					SM32/TI23/TO23/SEG41
P92					SM33/TI25/TO25/SGOA/SEG42
P93					SM34/ZPD34/TI27/TO27/SGO/SGOF/SEG43
P94					SM41/TI01/TO01/RTC1HZ/SEG44
P95					SM42/TI03/TO03/SEG45
P96					SM43/TI05/TO05/SEG46
P97					SM44/ZPD44/TI07/TO07/SEG47
P121	I	Port 12. 4-bit Input port.	HZ	Input	X1
P122					X2/EXCLK
P123					XT1
P124					XT2
P137	I	Port 13. 1-bit Input port.	HZ	Input	INTP5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset, PD: Pull down, HZ: High impedance

Note CTxD0 and CRxD0 are not provided for R5F10CMExFB and R5F10CMDxFB with no CAN channel.

2.1.4 100-pin products

Table 2-5. Port Pins for R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB, R5F10DPJxFB (1/3)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 8-bit I/O port. Input of P01 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	TI00/TO00/CTxD0/SEG14
P01					TI01/TO01/CRxD0/SEG15
P02					SO00/TI02/TO02/TI12/TO12/SEG16
P03					SI00/TI03/TO03/TI13/TO13/SEG17
P04					SCK00/TI04/TO04/TI14/TO14/SEG18
P05					TI05/TO05/TI15/TO15/SEG19
P06					TI06/TO06/TI16/TO16/SEG20
P07					TI07/TO07/TI17/TO17/SEG21
P10	I/O	Port 1. 8-bit I/O port. Input of P10, P11, and P17 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	LTxD1/SCK00/TI10/TO10/INTP4/SEG31
P11					LRxD1/INTPLR1/SI00/TI11/TO11/SEG30
P12					SO00/TI12/TO12/INTP2/SEG29
P13					SO01/TI13/TO13/SEG25
P14					TI14/TO14/LRxD0/INTPLR0/SEG24
P15					TI15/TO15/LTxD0/RTC1HZ/SEG23
P16					TI16/TO16/SEG22
P17					TI17/TO17/INTP0/SEG28
P20	I/O	Port 2. 8-bit I/O port. Can be set to analog input [Note] Input/output can be specified in 1-bit units.	HZ	Analog Input	AV_{REFP}/ANI0
P21					AV_{REFM}/ANI1
P22 to P27					ANI2 to ANI7
P30	I/O	Port 3. 8-bit I/O port. Input of P31 can be set to schmitt 1 input buffer. Output of P30 and P31 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI20/TO20/SCL11/SEG6
P31					TI21/TO21/SDA11/SEG7
P32					TI22/TO22/SO00/SEG8
P33					TI23/TO23/SI00/SEG9
P34					TI24/TO24/SCK00/SEG10
P35					TI25/TO25/SEG11
P36					TI26/TO26/SEG12
P37					TI27/TO27/SEG13
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: HZ Int.: PU	Input	TOOL0
P50	I/O	Port 5. 8-bit I/O port. Input of P50 to P52 and P55 to P57 can be set to schmitt 1 input buffer. Output of P50 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI02/TO02/SDA11/SEG49
P51					TI04/TO04/SCK10/SEG50
P52					TI06/TO06/SI10/SEG51
P53					TI13/TO13/SO10/SEG52
P54					TI14/TO14/SO01/SEG2
P55					TI15/TO15/SI01/SEG3
P56					TI16/TO16/SCK01/SEG4
P57					TI17/TO17/SEG5

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, PU: Pull up, HZ: High impedance

Note Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Table 2-5. Port Pins for R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB, R5F10DPJxFB (2/3)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P60	I/O	Port 6. 4-bit I/O port. Input of P61, P63 can be set to schmitt 1 input buffer. Output of P60 and P61 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	SCL11/TI20/TO20/INTP1
P61					SDA11/TI21/TO21/INTP3
P62					CTxD1/TI27/TO27Note
P63					CRxD1/TI26/TO26Note
P64					RTC1HZ/TI11/TO11
P65					TI25/TO25
P66					TI24/TO24/PCL
P70	I/O	Port 7. 6-bit I/O port. Input of P70 can be set to schmitt 1 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.	HZ	Input	CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD
P71					CTxD0/LTxD0/TOOLTXD
P72			Ext.: PD Int.: HZ		ADTRG/SGOA/SEG1
P73					SGO/SGOF/SEG0
P74					SCK01/TI23/TO23/SEG26
P75					PCL/SI01/TI22/TO22/SEG27
P80	I/O	Port 8. 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.	Ext.: PD Int.: HZ	Input	SM11/TI01/TO01/SEG32
P81					SM12/TI03/TO03/SEG33
P82					SM13/TI05/TO05/SEG34
P83					SM14/ZPD14/TI07/TO07/SEG35
P84					SM21/TI11/TO11/SEG36
P85					SM22/TI13/TO13/SEG37
P86					SM23/TI15/TO15/SEG38
P87					SM24/ZPD24/TI17/TO17/SEG39
P90	I/O	Port 9. 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.	Ext.: PD Int.: HZ	Input	SM31/TI21/TO21/SEG40
P91					SM32/TI23/TO23/SEG41
P92					SM33/TI25/TO25/SGOA/SEG42
P93					SM34/ZPD34/TI27/TO27/SGO/SGOF/SEG43
P94					SM41/TI01/TO01/RTC1HZ/SEG44
P95					SM42/TI03/TO03/SEG45
P96					SM43/TI05/TO05/SEG46
P97					SM44/ZPD44/TI07/TO07/SEG47
P121	I	Port 12. 4-bit Input port.	HZ	Input	X1
P122					X2/EXCLK
P123					XT1
P124					XT2

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset, PD: Pull down, HZ: High impedance

Note CTxD1 and CRxD1 are not provided for R5F10TPJxFB, R5F10DPGxFB, R5F10DPFxFB and R5F10DPExFB with a CAN channel.

Table 2-5. Port Pins for R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB, R5F10DPJxFB (3/3)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P130	O	Port 13. 1-bit output port, 1-bit input port and 6-bit I/O port. Input of P135 can be set to schmitt 1 input buffer. Output of P136 can be set to N-ch open-drain output. Input/output of P131 to P136 can be specified in 1-bit units. Use of an on-chip pull-up resistor of P131 to P136 can be specified by a software setting	Output Low	Output	–
P131	I/O		HZ	Input	SO10/LTxD1/TI21/TO21
P132					SI10/LRxD1/INTPLR1/TI20/TO20
P133					SCK10/TI22/TO22
P134					SGOA/CTxD1/TI24/TO24[Note1]
P135					SGO/SGOF/CRxD1/TI26/TO26[Note1]
P136			Ext.: PD Int.: HZ		TI00/TO00/SCL11/SEG48
P137	I		HZ	Input	INTP5
P140	I/O	Port 14. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting	HZ	Input	TI11/TO11
P150	I/O	Port 14. 1-bit I/O port. Can be set to analog input[Note2] Input/output can be specified.	HZ	Analog Input	ANI8

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, HZ: High impedance

Notes 1. CTxD1 and CRxD1 are not provided for R5F10TPJxFB, R5F10DPGxFB, R5F10DPFxFB and R5F10DPExFB with a CAN channel.

2. Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

<R> 2.1.5 128-pin products

Table 2-6. R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (1/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 8-bit I/O port. Input of P01 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	CTxD0/TI00/TO00/SEG14
P01					CRxD0/TI01/TO01/SEG15
P02					SO00/TxD0/TI02/TO02/TI12/TO12/SEG16
P03					SI00/RxD0/TI03/TO03/TI13/TO13/SEG17
P04					SCK00/TI04/TO04/TI14/TO14/SEG18
P05					TI05/TO05/TI15/TO15/SEG19
P06					TI06/TO06/TI16/TO16/SEG20
P07					TI07/TO07/TI17/TO17/SEG21
P10	I/O	Port 1. 8-bit I/O port. Input of P10, P11, and P17 can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	LTxD1/SCK00/TI10/TO10/INTP4/SEG31
P11					LRxD1/INTPLR1/SI00/RxD0/TI11/TO11/SEG30
P12					SO00/TxD0/TI12/TO12/INTP2/SEG29
P13					SO01/TI13/TO13/SEG25
P14					LRxD0/INTPLR0/TI14/TO14/SEG24
P15					LTxD0/RTC1HZ/TI15/TO15/SEG23
P16					TI16/TO16/SEG22
P17					TI17/TO17/INTP0/SEG28
P20	I/O	Port 2. 8-bit I/O port. Can be set to analog input [Note] Input/output can be specified in 1-bit units.	HZ	Analog Input	AV_{REFP}/ANI0
P21					AV_{REFM}/ANI1
P22 to P27					ANI2 to ANI7
P30	I/O	Port 3. 8-bit I/O port. Input of P31 can be set to schmitt 1 input buffer. Output of P30 and P31 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI20/TO20/SCL11/SEG6
P31					TI21/TO21/SDA11/SEG7
P32					TI22/TO22/SO00/TxD0/SEG8
P33					TI23/TO23/SI00/RxD0/SEG9
P34					TI24/TO24/SCK00/SEG10
P35					TI25/TO25/SEG11
P36					TI26/TO26/SEG12
P37					TI27/TO27/SEG13
P40	I/O	Port 4. 8-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: HZ Int.: PU	Input	TOOL0
P41			HZ		STOPST/TI04/TO04
P42			Ext.: PD Int.: HZ		TI10/TO10/SEG7
P43					TI22/TO22/SEG14
P44					TI23/TO23/SEG15
P45					SEG53
P46					$\overline{DBWR}$/SEG27
P47					$\overline{DBRD}$/SEG26

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, PU: Pull up, HZ: High impedance

Note Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

Table 2-6. R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (2/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P50	I/O	Port 5. 8-bit I/O port. Input of P50 to P52 and P55 to P57 can be set to schmitt 1 input buffer. Output of P50 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Ext.: PD Int.: HZ	Input	TI02/TO02/SDA11/SEG49
P51					TI04/TO04/SCK10/SEG50
P52					TI06/TO06/SI10/SEG51
P53					TI13/TO13/SO10/SEG52
P54					TI14/TO14/SO01/SEG2
P55					TI15/TO15/SI01/SEG3
P56					TI16/TO16/SCK01/SEG4
P57					TI17/TO17/SEG5
P60	I/O	Port 6. 7-bit I/O port. Input of P61, P63 can be set to schmitt 1 input buffer. Output of P60 and P61 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	SCL11/TI20/TO20/INTP1
P61					SDA11/TI21/TO21/INTP3
P62					CTxD1/TI27/TO27
P63					CRxD1/TI26/TO26
P64					RTC1HZ/TI11/TO11
P65					TI25/TO25
P66					TI24/TO24/PCL
P70	I/O	Port 7. 6-bit I/O port. Input of P70 can be set to schmitt 1 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.	HZ	Input	CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRxD
P71					CTxD0/LTxD0/TOOLTxD
P72			Ext.: PD Int.: HZ		SGOA/ADTRG/SEG1
P73					SGO/SGOF/SEG0
P74					SCK01/TI23/TO23/SEG26
P75					SI01/TI22/TO22/SEG27/PCL
P80	I/O	Port 8. 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.	Ext.: PD Int.: HZ	Input	SM11/TI01/TO01/SEG32
P81					SM12/TI03/TO03/SEG33
P82					SM13/TI05/TO05/SEG34
P83					SM14/ZPD14/TI07/TO07/SEG35
P84					SM21/TI11/TO11/SEG36
P85					SM22/TI13/TO13/SEG37
P86					SM23/TI15/TO15/SEG38
P87					SM24/ZPD24/TI17/TO17/SEG39

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, HZ: High impedance

Table 2-6. R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (3/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P90	I/O	Port 9. 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.	Ext.: PD Int.: HZ	Input	SM31/TI21/TO21/SEG40
P91					SM32/TI23/TO23/SEG41
P92					SM33/SGOA/TI25/TO25/SEG42
P93					SM34/ZPD34/SGO/SGOF/TI27/TO27/SEG43
P94					SM41/RTC1HZ/TI01/TO01/SEG44
P95					SM42/TI03/TO03/SEG45
P96					SM43/TI05/TO05/SEG46
P97					SM44/ZPD44/TI07/TO07/SEG47
P100	I/O	Port 10. 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.	Ext.: PD Int.: HZ	Input	TI24/TO24/SEG36
P101					TI25/TO25/SEG37
P102					TI26/TO26/SEG38
P103					TI27/TO27/SEG39
P104					TI01/TO01/SEG44
P105					TI02/TO02/SEG45
P106					TI05/TO05/SEG46
P107					TI06/TO06/SEG47
P110	I/O	Port 10. 8-bit I/O port. Input can be set to schmitt 1 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.	Ext.: PD Int.: HZ	Input	DBD0/SCK00/TI00/TO00/SEG35
P111					DBD1/SI00/RxD0/TI02/TO02/SEG34
P112					DBD2/SO00/TxD0/TI04/TO04/SEG33
P113					DBD3/TI06/TO06/SEG32
P114					DBD4/TI07/TO07/SEG31
P115					DBD5/TI10/TO10/SEG30
P116					DBD6/TI12/TO12/SEG29
P117					DBD7/TI20/TO20/SEG28
P121	I	Port 12. 4-bit Input port, 3-bit I/O port. Input/output of P125 to P127 can be specified in 1-bit units. Use of an on-chip pull-up resistor of P125 to P127 can be specified by a software setting.	HZ	Input	X1
P122					X2/EXCLK
P123					XT1
P124					XT2
P125	I/O		Ext.: PD Int.: HZ		TI12/TO12/SEG25
P126					TI14/TO14/SEG24
P127					TI16/TO16/SEG23

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, HZ: High impedance

Table 2-6. R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (4/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P130	O	Port 13. 1-bit output port, 1-bit input port and 6-bit I/O port. Input of P135 can be set to schmitt 1 input buffer. Output of P136 can be set to N-ch open-drain output. Input/output of P131 to P136 can be specified in 1-bit units. Use of an on-chip pull-up resistor of P131 to P136 can be specified by a software setting	Output Low	Output	RESOUT
P131	I/O		HZ	Input	SO10/LTxD1/TI21/TO21
P132					SI10/LRxD1/INTPLR1/TI20/TO20
P133					SCK10/TI22/TO22
P134					SGOA/CTxD1/TI24/TO24
P135					SGO/SGOF/CRxD1/TI26/TO26
P136			Ext.: PD Int.: HZ		TI00/TO00/SCL11/SEG48
P137	I		HZ	Input	INTP5
P140	I/O	Port 14. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting	HZ	Input	TI11/TO11
P150	I/O	Port 15. 3-bit I/O port. Can be set to analog input Note Input/output can be specified.	HZ	Analog Input	ANI8
P151					ANI9
P152					ANI10

Remark Ext. (external reset): POR reset or pin reset, Int. (internal reset): WDT reset or LVD reset,
PD: Pull down, HZ: High impedance

Note Setting digital or analog to each pin can be done in A/D port configuration register (ADPC).

<R> 2.1.6 Pins for each product (pins other than port pins)

(1/6)

Function Name	I/O	Function	48-pin	64-pin	80-pin	100-pin	128-pin
			R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
ADTRG	Input	A/D conversion start trigger external input	31	43	55	65	53
ANI0/AVREFP		A/D converter analog input	4	4	8	10	113
ANI1/AVREFM			3	3	7	9	112
ANI2			2	2	6	8	111
ANI3			1	1	5	7	110
ANI4			–	–	4	6	109
ANI5			–	–	3	5	108
ANI6			–	–	2	4	107
ANI7			48	64	1	3	106
ANI8			–	–	–	2	105
ANI9			–	–	–	–	104
ANI10			–	–	–	–	103
COM0	Output	LCD common output 0	36	48	60	70	58
COM1		LCD common output 1	35	47	59	69	57
COM2		LCD common output 2	34	46	58	68	56
COM3		LCD common output 3	33	45	57	67	55
SEG0	Output	LCD segment output 0	32	44	56	66	54
SEG1		LCD segment output 1	31	43	55	65	53
SEG2		LCD segment output 2	30	42	54	64	52
SEG3		LCD segment output 3	29	41	53	63	51
SEG4		LCD segment output 4	28	40	52	62	50
SEG5		LCD segment output 5	27	39	51	61	49
SEG6		LCD segment output 6	26	38	50	60	48
SEG7		LCD segment output 7	25	37	49	59	47, 95
SEG8		LCD segment output 8	–	36	48	58	46
SEG9		LCD segment output 9	24	35	47	57	45
SEG10		LCD segment output 10	–	–	46	56	44
SEG11		LCD segment output 11	–	–	45	55	43
SEG12		LCD segment output 12	–	–	44	54	42
SEG13		LCD segment output 13	–	–	43	53	41
SEG14		LCD segment output 14	23	34	42	52	40, 94
SEG15		LCD segment output 15	22	33	41	51	39, 93
SEG16		LCD segment output 16	–	32	40	50	35
SEG17		LCD segment output 17	–	31	39	49	34
SEG18		LCD segment output 18	–	30	38	48	33
SEG19		LCD segment output 19	–	29	37	47	32
SEG20		LCD segment output 20	–	–	36	46	31
SEG21		LCD segment output 21	–	28	35	45	30
SEG22		LCD segment output 22	–	–	34	44	29
SEG23		LCD segment output 23	–	27	33	43	28, 38
SEG24		LCD segment output 24	21	26	32	42	27, 37
SEG25		LCD segment output 25	20	25	31	41	26, 36
SEG26		LCD segment output 26	19	24	30	40	25, 19

(2/6)

Function Name	I/O	Function	48-pin	64-pin	80-pin	100-pin	128-pin
			R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
SEG27		LCD segment output 27	18	23	29	39	24, 18
SEG28		LCD segment output 28	-	22	28	38	23, 17
SEG29	Output	LCD segment output 29	17	21	27	37	22, 16
SEG30		LCD segment output 30	16	20	26	36	21, 15
SEG31		LCD segment output 31	15	19	25	35	20, 14
SEG32		LCD segment output 32	47	63	80	95	88 ,13
SEG33		LCD segment output 33	46	62	79	94	87 ,12
SEG34		LCD segment output 34	45	61	78	93	86 ,11
SEG35		LCD segment output 35	44	60	77	92	85 ,10
SEG36		LCD segment output 36	–	57	74	89	82 ,92
SEG37		LCD segment output 37	–	56	73	88	81 ,91
SEG38		LCD segment output 38	–	55	72	87	80 ,90
SEG39		LCD segment output 39	–	54	71	86	79 ,89
SEG40		LCD segment output 40	41	53	70	85	78
SEG41		LCD segment output 41	40	52	69	84	77
SEG42		LCD segment output 42	39	51	68	83	76
SEG43		LCD segment output 43	38	50	67	82	75
SEG44		LCD segment output 44	37	49	64	79	72, 68
SEG45		LCD segment output 45	–	–	63	78	71, 67
SEG46		LCD segment output 46	–	–	62	77	70, 66
SEG47		LCD segment output 47	–	–	61	76	69, 65
SEG48		LCD segment output 48	–	–	–	75	64
SEG49		LCD segment output 49	–	–	–	74	63
SEG50		LCD segment output 50	–	–	–	73	62
SEG51		LCD segment output 51	–	–	–	72	61
SEG52		LCD segment output 52	–	–	–	71	60
SEG53		LCD segment output 53	–	–	–	–	59
TI00	Input	External count clock input/ capture trigger to 16-bit timer 00 to 07	23	34	42	75,52	40, 64,10
TI01			22, 37, 47	63,49,33	80,64,41	95,79,51	39, 88, 72, 68
TI02			–	32	40	74,50	35, 63, 67, 11
TI03			46	62,7,31	79,63,11,39	94,78,49,13	34, 87, 71, 116
TI04			–	30	38	48,73	33, 62, 12, 101
TI05			45	61, 29	78,62,37	93,77,47	32, 86, 70, 66
TI06			–	–	36	46,72	31, 61, 65, 13
TI07			44	60,28	77,61,35	92,76,45	30, 85, 69, 14
TO00	Output	16-bit timer 00 to 07 output	23	34	42	75,52	40, 64, 10
TO01			22, 37, 47	63,49,33	80,64,41	95,79,51	39, 88, 72, 68
TO02			–	32	40	74,50	35, 63, 67, 11
TO03			46	62,7, 31	79,63,11,39	94,78,49,13	34, 87, 71, 116
TO04			–	30	38	48,73	33, 62, 12, 101
TO05			45	61,29	78,62,37	93,77,47	32, 86, 70, 66
TO06			–	–	36	46,72	31, 61, 65, 13
TO07			44	60,28	77,61,35	92,76,45	30, 85, 69, 14

(3/6)

Function Name	I/O	Function	48-pin R5F10CGx/ R5F10DGx	64-pin R5F10CLx/ R5F10DLx	80-pin R5F10CMx/ R5F10DMx	100-pin R5F10TPx/ R5F10DPx	128-pin R5F10DSx
TI10	Input	External count clock input/ capture trigger to 16-bit timer 10 to 17	15	19	25	35	20, 15 , 95
TI11			16	57, 20	74, 26	89,14,30, 36	21, 82, 117, 5
TI12			17	32, 21	40, 27	50, 37	22, 36, 35, 16
TI13			20	56,25,31	73,31,39	88,71,41,49	34, 26, 60, 81
TI14			21, 30	30,26,42	38,32,54	48,42,64	27, 37, 52, 33
TI15			29	55,41,27,29	72,53,33,37	87,63,43,47	28, 32, 51, 80
TI16			28	40	36,34,52	46,44,62	29, 38, 50, 31
TI17			27	54,39,22,28	71,51,28,35	86,61,38,45	23, 30, 79, 49
TO10	Output	16-bit timer 10 to 17 output	15	19	25	35	20, 15, 95
TO11			16	57, 20	74,26	89,14,30,36	21, 82, 117, 5
TO12			17	32,21	40,27	50,37	22, 36, 35, 16
TO13			20	56,25,31	73,31,39	88,71,41,49	34, 26, 60, 81
TO14			21, 30	30,26,42	38,32,54	48,42,64	27, 37, 52, 33
TO15			29	55,41,27,29	72,53,33,37	87,63,43,47	28, 32, 51, 80
TO16			28	40	36,34,52	46,44,62	29, 38, 50, 31
TO17			27	54,39,22,28	71,51,28,35	86,61,38,45	23, 30, 79, 49
TI20	Input	External count clock input /capture trigger to 16-bit timer 20 to 27	13, 26	17,38	21,50	26,60,99	1, 48, 99, 17
TI21			14, 25, 41	53,37,18	70,49,22	100,85,59,27	2, 47, 100, 78
TI22			18	23,36	29,48	39,58,98	24, 46, 98, 94
TI23			19, 24, 40	52,35,24	69,47,30	84,57,40	25, 45, 77, 93
TI24			–	–	24,46	32,56,97	7, 44, 97, 92
TI25			39	51	68,45,23	83,55,31	6, 76, 43, 91
TI26			–	–	44	29,54,96	4, 42, 96, 90
TI27			38	50	67,43	82,53,28	75, 3, 41, 89
TO20	Output	16-bit timer 20 to 27 output	13, 26	17,38	21,50	26,60,99	1, 48, 99, 17
TO21			14, 25, 41	53,37,18	70,49,22	100,85,59,27	2, 47, 100, 78
TO22			18	23,36	29,48	39,58,98	24, 46, 98, 94
TO23			19, 24, 40	52,35,24	69,47,30	84,57,40	25, 45, 77, 93
TO24			–	–	24,46	32,56,97	7, 44, 97, 92
TO25			39	51	68,45,23	83,55,31	6, 76, 43, 91
TO26			–	–	44	29,54,96	4, 42, 96, 90
TO27			38	50	67,43	82,53,28	75, 3, 41, 89
SCK00	Input/ Output	Clock input/output for CSI00	15	19,30	46,25,38	56,35,48	20, 33, 44,10
SI00	Input	Serial data input to CSI00	16	20,31	47,26,39	57,36,49	21, 34, 45, 11
SO00	Output	Serial data output from CSI00	17	21,32	48,27,40	58,37,50	22, 35, 46, 12
SCK01	Input/ Output	Clock input/output for CSI01	19, 28	40,24	52,30	62,40	50, 25
SI01	Input	Serial data input to CSI01	18, 29	41,23	53,29	63,39	51, 24
SO01	Output	Serial data output from CSI01	20, 30	42,25	54,31	64,41	52, 26

(4/6)

Function Name	I/O	Function	48-pin	64-pin	80-pin	100-pin	128-pin
			R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
RxD0	Input	Serial data input to UART0	–	–	–	–	21, 34, 45, 11
TxD0	Output	Serial data output from UART0	–	–	–	–	22, 35, 46, 12
SCK10	Input/ Output	Clock input/output for CSI10	–	–	–	98,73	98, 62
SI10	Input	Serial data input to CSI10	–	–	–	99,72	99, 61
SO10	Output	Serial data output from CSI10	–	–	–	100,71	100, 60
SCL11	Output	Clock output for simplified I^2C	13, 26	38,17	50,21	75,60,26	64, 48, 1
SDA11	Input/ Output	Serial data I/O for simplified I^2C	14, 25	37,18	49,22	74,59,27	63, 47, 2
LRxD0	Input	Serial data input to LIN-UART0	–	26,7	32,11	42,13	27, 116
INTPLR0	Input	External interrupt request input for which the valid edge for LIN-UART0	–	26,7	32,11	42,13	27, 116
LTxD0	Output	Serial data output from LIN-UART0	–	27,6	33,10	43,12	28, 115
LRxD1	Input	Serial data input to LIN-UART1	16	20	26	99,36	99, 21
INTPLR1	Input	External interrupt request input for which the valid edge for LIN-UART1	16	20	26	99,36	99, 21
LTxD1	Output	Serial data output from LIN-UART1	15	19	25	100,35	100, 20
CRxD0	Input	CAN receive data input 0	22Note1	33,7Note1	41,11Note1	51,13	39, 116
CTxD0	Output	CAN transmit data output 0	23Note1	34,6Note1	42,10Note1	52,12	40, 115
CRxD1	Input	CAN receive data input 1	–	–	–	96,29Note2	96, 4
CTxD1	Output	CAN transmit data output 1	–	–	–	97,28Note2	97, 3
RTC1Hz	Output	Realtime clock calibration output (1Hz)	37	49,27	64,33	79,30,43	72, 28, 5
SGOA	Output	SG output(Amplitude PWM)	31, 39	51,43	68,55	97,83,65	97, 76, 53
SGO/SGOF	Output	SG output (AND with PWM & Frequency) / SG output (Frequency)	32, 38	50,44	67, 56	96,82,66	96, 75, 54
SM11	Output	Stepper motor output 11	47	63	80	95	88
SM12	Output	Stepper motor output 12	46	62	79	94	87
SM13	Output	Stepper motor output 13	45	61	78	93	86
SM14	Output	Stepper motor output 14	44	60	77	92	85
ZPD14	Input	Zero point detection input 14	44	60	77	92	85

Notes 1. These pins are only for the following products:
- R5F10DGE, R5F10DGD, R5F10DGC (48 pin products)
- R5F10DLE, R5F10DLD (64 pin products)
- R5F10DMJ, R5F10DMG, R5F10DMF, R5F10DME, R5F10DMD (80 pin products)
- R5F10TPJ, R5F10DPG. R5F10DPF, R5F10DPE, R5F10DPL, R5F10DPK,R5F10DPJ (100-pin products)

2. These pins are only for R5F10DPL, R5F10DPK,R5F10DPJ.

(5/6)

Function Name	I/O	Function	48-pin	64-pin	80-pin	100-pin	128-pin
			R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
SM21	Output	Stepper motor output 21	–	57	74	89	82
SM22	Output	Stepper motor output 22	–	56	73	88	81
SM23	Output	Stepper motor output 23	–	55	72	87	80
SM24	Output	Stepper motor output 24	–	54	71	86	79
ZPD24	Input	Zero point detection input 24	–	54	71	86	79
SM31	Output	Stepper motor output 31	–	–	70	85	78
SM32	Output	Stepper motor output 32	–	–	69	84	77
SM33	Output	Stepper motor output 33	–	–	68	83	76
SM34	Output	Stepper motor output 34	–	–	67	82	75
ZPD34	Input	Zero point detection input 34	–	–	67	82	75
SM41	Output	Stepper motor output 41	–	–	64	79	72
SM42	Output	Stepper motor output 42	–	–	63	78	71
SM43	Output	Stepper motor output 43	–	–	62	77	70
SM44	Output	Stepper motor output 44	–	–	61	76	69
ZPD44	Input	Zero point detection input 44	–	–	61	76	69
DBD0	Input/ Output	LCD Bus I/F data lines 0	–	–	–	–	10
DBD1	Input/ Output	LCD Bus I/F data lines 1	–	–	–	–	11
DBD2	Input/ Output	LCD Bus I/F data lines 2	–	–	–	–	12
DBD3	Input/ Output	LCD Bus I/F data lines 3	–	–	–	–	13
DBD4	Input/ Output	LCD Bus I/F data lines 4	–	–	–	–	14
DBD5	Input/ Output	LCD Bus I/F data lines 5	–	–	–	–	15
DBD6	Input/ Output	LCD Bus I/F data lines 6	–	–	–	–	16
DBD7	Input/ Output	LCD Bus I/F data lines 7	–	–	–	–	17
$\overline{\text{DBWR}}$	Output	LCD Bus I/F write strobe	–	–	–	–	18
$\overline{\text{DBRD}}$	Output	LCD Bus I/F read strobe	–	–	–	–	19
STOPST	Output	STOP status output	–	–	–	–	101
RESOUT	Output	Reset output signal	–	–	–	–	102
PCL	Output	Clock output	18	23	24,29	32,39	24, 7
INTP0	Input	Interrupt from peripheral 0	–	22	28	38	23
INTP1	Input	Interrupt from peripheral 1	13	17	21	26	1
INTP2	Input	Interrupt from peripheral 2	17	21	27	37	22
INTP3	Input	Interrupt from peripheral 3	14	18	22	27	2
INTP4	Input	Interrupt from peripheral 4	15	19	25	35	20
INTP5	Input	Interrupt from peripheral 5	7	11	15	18	121

(6/6)

Function Name	I/O	Function	48-pin	64-pin	80-pin	100-pin	128-pin
			R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
EXCLK	Input	External clock input for main system clock	8	12	16	19	122
X1		Resonator connection for main system clock	9	13	17	20	123
X2			8	12	16	19	122
XT1		Resonator connection for sub clock	–	10	14	17	120
XT2			–	9	13	16	119
$\overline{\text{RESET}}$	Input	External system reset input	6	8	12	15	118
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming	–	7	11	13	116
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming	–	6	10	12	115
TOOL0	Input/ output	Data I/O for flash memory programmer/debugger	5	5	9	11	114
REGC		Connecting regulator output stabilization capacitance for internal operation. Connect to VSS via a capacitor (0.47 to 1 µF).	10	14	18	21	124
V_{DD}		Power supply for P20 to P27, P150, P137, P121 to P124, $\overline{\text{RESET}}$	12	16	20	24	127
EV_{DD}, EV_{DD0}, EV_{DD1}		Power supply for P00 to P07, P10 to P17, P30 to P37, P50 to P57, P60 to P66, P70 to P75, P130 to P136 and P140	12	16	20	25, 33	8, 128
SMV_{DD}, SMV_{DD0}, SMV_{DD1}		Power supply for P80 to P87, P90 to P97	43	59	66, 76	81, 91	74, 84
V_{SS}		Ground potential for P20-P27, P150, P137, P121 to P124, $\overline{\text{RESET}}$	11	15	19	22	125
EV_{SS}, EV_{SS0}, EV_{SS1}		Ground potential for P00 to P07, P10 to P17, P30 to P37, P50 to P57, P60 to P66, P70 to P75, P130 to P136 and P140	11	15	19	23, 34	9, 126
SMV_{SS}, SMV_{SS0}, SMV_{SS1}		Ground potential for P80 to P87, P90 to P97	42	58	65, 75	80, 90	73, 83

<R> 2.2 Description of Pin Function

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

2.2.1 P00 to P07 (port 0)

48-pin products: P00 to P01 function as a 2-bit I/O port.
64-pin products: P00 to P05 and P07 function as a 7-bit I/O port.
80-pin products: P00 to P07 function as an 8-bit I/O port.
100-pin products: P00 to P07 function as an 8-bit I/O port.
128-pin products: P00 to P07 function as an 8-bit I/O port.

These pins also function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

Input to the P01 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer in 1-bit units, using port input mode register 0 (PIM0).

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

(1) Port mode

48-pin products: P00 to P01 function as a 2-bit I/O port.
64-pin products: P00 to P05 and P07 function as a 7-bit I/O port.
80-pin products: P00 to P07 function as an 8-bit I/O port.
100-pin products: P00 to P07 function as an 8-bit I/O port.
128-pin products: P00 to P07 function as an 8-bit I/O port.

P00 to P07 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 P07 function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

(a) TI00 (P00) to TI07 (P07) and TI12 (P02) to TI17 (P07)

These are the external count clock input /capture trigger to 16-bit timer 00 to 07, and 12 to 17.

(b) TO00 (P00) to TO07 (P07) and TO12 (P02) to TO17 (P07)

These are the timer output pins of 16-bit timers 00 to 07, and 12 to 17.

(c) CTxD0 (P00)

This is a CAN serial transmit data output pin of aFCAN0.

(d) CRxD0 (P01)

This is a CAN serial receive data input pin of aFCAN0.

(e) SO00 (P02)

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

(f) SI00 (P03)

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

(g) SCK00 (P04)

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

(h) SEG14 (P00) to SEG21 (P07)

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

(i) TxD0 (P02)

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

(j) RxD0 (P03)

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

2.2.2 P10 to P17 (port1)

48-pin products: P10 to P14 function as a 5-bit I/O port.
64-pin products: P10 to P15 and P17 function as a 7-bit I/O port.
80-pin products: P10 to P17 function as an 8-bit I/O port.
100-pin products: P10 to P17 function as an 8-bit I/O port.
128-pin products: P10 to P17 function as an 8-bit I/O port.

These pins also function as serial interface data I/O, timer I/O, clock I/O, external interrupt request input and segment signal outputs for the LCD controller/driver.

Input to the P10, P11 and P17 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer in 1-bit units, using port input mode register 1 (PIM1).

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

(1) Port mode

48-pin products: P10 to P14 function as a 5-bit I/O port.
64-pin products: P10 to P15 and P17 function as a 7-bit I/O port.
80-pin products: P10 to P17 function as an 8-bit I/O port.
100-pin products: P10 to P17 function as an 8-bit I/O port.
128-pin products: P10 to P17 function as an 8-bit 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 serial interface data I/O, timer I/O, clock I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

(a) LTxD0 (P15), LTxD1 (P10)

These are the Serial data output from LIN-UART.

(b) LRxD0 (P14), LRxD1 (P11)

These are the Serial data input to LIN-UART.

(c) SO00 (P12), SO01 (P13)

These are the serial data output pins of serial interface CSI00 and CSI01.

(d) SI00 (P11)

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

(e) SCK00 (P10)

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

(f) TI10 (P10) to TI17 (P17)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 10 to 17.

(g) TO10 (P10) to TO17 (P17)

These are the timer output pins of 16-bit timers 10 to 17.

(h) RTC1HZ (P15)

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

(i) INTP0 (P17), INTP2 (P12), INTP4 (P10)

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.

(j) INTPLR0 (P14), INTPLR1 (P11)

These are the external interrupt request input for which the valid edge for LIN-UART0 and 1.

(k) SEG22 (P16) to SEG25 (P13), SEG28 (P17) and SEG29 (P12) to SEG31(P10)

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

(l) TxD0 (P12)

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

(m) RxD0 (P11)

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

2.2.3 P20 to P27 (port2)

48-pin products:	P20 to P23 and P27 function as a 5-bit I/O.
64-pin products:	P20 to P23 and P27 function as a 5-bit I/O.
80-pin products:	P20 to P27 function as an 8-bit I/O port.
100-pin products:	P20 to P27 function as an 8-bit I/O port
128-pin products:	P20 to P27 function as an 8-bit I/O port

These pins also function as A/D converter analog input and reference voltage input.

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

(1) Port mode

48-pin products:	P20 to P23 and P27 function as a 5-bit I/O.
64-pin products:	P20 to P23 and P27 function as a 5-bit I/O.
80-pin products:	P20 to P27 function as an 8-bit I/O port.
100-pin products:	P20 to P27 function as an 8-bit I/O port
128-pin products:	P20 to P27 function as an 8-bit I/O port

P20 to P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P20 to P27 function as A/D converter analog input and reference voltage input.

(a) ANI0 (P20) to ANI7 (P27)

These are the analog input pins (ANI0 to ANI7) of A/D converter.

When using these pins as analog input pins, see 11.10 (5) Analog input (ANIn) pins.

(b) AVREFP (P20)

This is a pin that inputs the A/D converter reference potential (+ side).

(c) AVREFM (P21)

This is a pin that inputs the A/D converter reference potential (−side).

2.2.4 P30 to P37 (port3)

48-pin products: P30, P31, P33 function as a 3-bit I/O port.
64-pin products: P30 to P33 function as a 4-bit I/O port.
80-pin products: P30 to P37 function as an 8-bit I/O port.
100-pin products: P30 to P37 function as an 8-bit I/O port.
128-pin products: P30 to P37 function as an 8-bit I/O port.

These pins also function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

Input to the P31 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer, using port input mode register 1 (PIM3).

Output from the P30 and P31 pins can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance) in 1-bit units, using port output mode register (POM).

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

(1) Port mode

48-pin products: P30, P31, P33 function as a 3-bit I/O port.
64-pin products: P30 to P33 function as a 4-bit I/O port.
80-pin products: P30 to P37 function as an 8-bit I/O port.
100-pin products: P30 to P37 function as an 8-bit I/O port.
128-pin products: P30 to P37 function as an 8-bit I/O port.

P30 to P37 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 P37 function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

(a) TI20 (P30) to TI27 (P37)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 20 to 27.

(b) TO20 (P30) to TO27(P37)

These are the timer output pins of 16-bit timers 20 to 27.

(c) SO00 (P32 in 80-pin and 100-pin product)

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

(d) SI00 (P33 in 80-pin and 100-pin product)

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

(e) SCK00 (P34)

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

(f) SCL11 (P30)

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

(g) SDA11 (P31)

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

(h) SEG6 (P30) to SEG13 (P37)

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

(i) TxD0 (P32)

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

(j) RxD0 (P33)

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

2.2.5 P40 (port4)

P40 to P47 function as an 8-bit I/O port.

These pins also functions as data I/O for a flash memory programmer / debugger, STOP status output, timer I/O, segment signal outputs for the LCD controller/driver, LCD Bus I/F write strobe, and LCD Bus I/F read strobe.

When this pin functions as data I/O for a flash memory programmer / debugger, this pin is N-ch open-drain output.

(1) Port mode

P40 to P47 function as an I/O port. These ports can be set to input or output port 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).

(2) Control mode

P40 to P47 function as data I/O for a flash memory programmer / debugger, STOP status output, timer I/O, segment signal outputs for the LCD controller/driver, LCD Bus I/F write strobe, and LCD Bus I/F read strobe.

(a) TOOL0 (P40)

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).

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

Table 2-6. Relationships Between P40/TOOL0 and Operation Mode After Reset Release

P40/TOOL0	Operating mode
VDD	Normal operation mode
0 V	Flash memory programming mode

For details, see 29.5 Programming Method.

(b) STOPST (P41)

This is an output pin of STOP status.

(c) TI04 (P41), TI10 (P42) TI22 (P43) TI23 (P44)

These are the pin for inputting an external count clock/capture trigger to 16-bit timers 04, 10, 22, 23.

(d) TO04 (P41), TO10 (P42), TO22 (P43), TO23 (P44)

These are the timer output pins of 16-bit timers 04, 10, 22, 23.

(e) SEG7 (P42), SEG14 (P43), SEG15 (P44), SEG53 (P45), SEG27 (P46), SEG26 (P47)

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

(f) $\overline{\text{DBWR}}$ (P46)

This is an output pins for LCD Bus I/F write strobe.

(g) $\overline{\text{DBRD}}$ (P47)

This is an output pins for LCD Bus I/F read strobe.

2.2.6 P50 to P57 (port5)

48-pin products: P54 to P57 function as a 4-bit I/O port.
64-pin products: P54 to P57 function as a 4-bit I/O port.
80-pin products: P54 to P57 function as a 4-bit I/O port.
100-pin products: P50 to P57 function as an 8-bit I/O port.
128-pin products: P50 to P57 function as an 8-bit I/O port.

These pins also function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

Input to the P50 to P52 and P55 to P57 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer, using port input mode register 5 (PIM5).

Output from the P50 pin can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance), using port output mode register (POM).

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

(1) Port mode

48-pin products: P54 to P57 function as a 4-bit I/O port.
64-pin products: P54 to P57 function as a 4-bit I/O port.
80-pin products: P54 to P57 function as a 4-bit I/O port.
100-pin products: P50 to P57 function as an 8-bit I/O port.
128-pin products: P50 to P57 function as an 8-bit 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 timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller / driver.

(a) TI02 (P50), TI04 (P51), TI06 (P52) and TI13 (P53) to TI17 (P57)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 02, 04, 06, and 13 to 17.

(b) TO02 (P50), TO04 (P51), TO06 (P52) and TO13 (P53) to TO17 (P57)

These are the timer output pins of 16-bit timers 02, 04, 06 and 13 to 17.

(c) SO01 (P54), SO10 (P53)

These are the serial data output pins of serial interface CSI01 and CSI10.

(d) SI01 (P55), SI10 (P52)

These are the serial data input pins of serial interface CSI01 and CSI10.

(e) SCK01 (P56), SCK10 (P51)

These are the serial clock I/O pins of serial interface CSI01 and CSI10.

(f) SDA11 (P50)

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

(g) SEG2 (P54) to SEG5 (P57), SEG49 (P50) to SEG52 (P53)

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

2.2.7 P60 to P66 (port6)

48-pin products: P60 and P61 function as a 2-bit I/O port.
64-pin products: P60 and P61 function as a 2-bit I/O port.
80-pin products: P60, P61, P65, and P66 function as a 4-bit I/O port.
100-pin products: P60 to P66 function as a 7-bit I/O port.
128-pin products: P60 to P66 function as a 7-bit I/O port.

These pins also function as serial interface data I/O, timer I/O, real-time clock correction clock output, clock / buzzer output, and external interrupt request input.

Input to the P61 and P63 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer in 1-bit units, using port input mode register 6 (PIM6).

Output from the P60 and P61 pins can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance) in 1-bit units, using port output mode register 6 (POM6).

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

(1) Port mode

48-pin products: P60 and P61 function as a 2-bit I/O port.
64-pin products: P60 and P61 function as a 2-bit I/O port.
80-pin products: P60, P61, P65, and P66 function as a 4-bit I/O port.
100-pin products: P60 to P66 function as a 7-bit I/O port.
128-pin products: P60 to P66 function as a 7-bit I/O port.

P60 to P66 can be set to input or output port in 1-bit units using port mode register 6 (PM6). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

(2) Control mode

P60 to P66 function as serial interface data I/O, timer I/O, real-time clock correction clock output, clock / buzzer output, and external interrupt request input.

(a) TI20 (P60), TI21 (P61), TI24 (P66), TI25 (P65), TI26 (P63) and TI27 (P62)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 20, 21, and 24 to 27.

(b) TO20 (P60), TO21 (P61), TO24 (P66), TO25 (P65), TO26 (P63) and TO27 (P62)

These are the timer output pins of 16-bit timers 20, 21, and 24 to 27.

(c) SCL11 (P60)

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

(d) SDA11 (P61)

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

(e) RTC1HZ (P64)

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

(f) PCL (P66)

This is a clock/buzzer output pin.

(g) INTP1 (P60) and INTP3 (P61)

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.

2.2.8 P70 to P75 (port7)

P70 to P75 function as a 6-bit I/O port.

These pins also function as A/D conversion start trigger input, output pins for the sound generator, serial interface data I/O, timer I/O, clock / buzzer output, flash memory programming I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

Input to the P70 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer, using port input mode register 7 (PIM7).

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

(1) Port mode

P70 to P75 function as a 6-bit I/O port.

P70 to P75 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 P75 function as an I/O port. These pins also function as A/D conversion trigger input, output pins for the sound generator, serial interface data I/O, timer I/O, clock / buzzer output, flash memory programming I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

(a) ADTRG (P72)

This is an external input pin for AD conversion start trigger.

(b) SGO (P73)

This is an output pin for the sound generator.

(c) SGOA (P72)

This is an amplitude PWM output pin for the sound generator.

(d) SGOF (P73)

This is a frequency output pin for the sound generator.

(e) TI03 (P70), TI22 (P75), and TI23 (P74)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers03, 22, and 23.

(f) TO03 (P70), TO22 (P75), and TO23 (P74)

These are the timer output pins of 16-bit timers 03, 22, and 23.

(g) CTxD0 (P71)

This is a CAN serial transmit data output pin of aFCAN0.

(h) CRxD0 (P70)

This is a CAN serial receive data input pin of aFCAN0.

(i) SCK01 (P74)

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

(j) SI01 (P75)

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

(k) LTXD0 (P71)

This is a Serial data output from LIN-UART.

(l) LRxD0 (P70)

This is a Serial data input to LIN-UART.

(m) PCL (P75)

This is a clock/buzzer output pin.

(n) TOOLTxD (P71)

This UART serial data output pin for an external device connection is used during flash memory programming.

(o) TOOLRxD (P70)

This UART serial data input pin for an external device connection is used during flash memory programming.

(p) INTPLR0 (P70)

This is an external interrupt request input for which the valid edge for LIN-UART0.

(q) SEG0 (P73), SEG1 (P72), SEG26 (P74), and SEG27 (P75)

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

2.2.9 P80 to P87 (port8)

48-pin products: P80 to P83 function as an 4-bit I/O port.
64-pin products: P80 to P87 function as an 8-bit I/O port.
80-pin products: P80 to P87 function as an 8-bit I/O port.
100-pin products: P80 to P87 function as an 8-bit I/O port.
128-pin products: P80 to P87 function as an 8-bit I/O port.

These pins also function as timer I/O, stepper motor controller/driver outputs/inputs, and segment signal outputs for the LCD controller/driver.

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

(1) Port mode

48-pin products: P80 to P83 function as an 4-bit I/O port.
64-pin products: P80 to P87 function as an 8-bit I/O port.
80-pin products: P80 to P87 function as an 8-bit I/O port.
100-pin products: P80 to P87 function as an 8-bit I/O port.
128-pin products: P80 to P87 function as an 8-bit I/O port.

P80 to P87 can be set to input or output port in 1-bit units using port mode register 8 (PM8). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 8 (PU8).

(2) Control mode

P80 to P87 function as timer I/O, stepper motor controller/driver outputs/inputs, and segment signal outputs for the LCD controller/driver.

(a) TI01 (P80), TI03 (P81), TI05 (P82), TI07 (P83), TI11 (P84), TI13 (P85), TI15 (P86), and TI17 (P87)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers01, 03, 05, 07, 11, 13, 15, and 17.

(b) TO01 (P80), TO03 (P81), TO05 (P82), TO07 (P83), TO11 (P84), TO13 (P85), TO15 (P86), and TO17 (P87)

These are the timer output pins of 16-bit timers 01, 03, 05, 07, 11, 13, 15, and 17.

(c) SM11 (P80) to SM14 (P83) and SM21 (P84) to SM24 (P87)

These are the output pins for the stepper motor controller/driver.

(d) ZPD14 (P83), ZPD24 (P87)

These are the Zero Point Detection (ZPD) input pins for the stepper motor controller/driver.

(e) SEG32 (P80) to SEG39 (P87)

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

2.2.10 P90 to P97 (port9)

48-pin products: P90 to P94 function as a 5-bit I/O port.
64-pin products: P90 to P94 function as a 5-bit I/O port.
80-pin products: P90 to P97 function as an 8-bit I/O port.
100-pin products: P90 to P97 function as an 8-bit I/O port.
128-pin products: P90 to P97 function as an 8-bit I/O port.

These pins also function as timer I/O, output pins for the sound generator, stepper motor controller/driver outputs/inputs, real-time clock correction clock output, and segment signal outputs for the LCD controller/driver.

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

(1) Port mode

48-pin products: P90 to P94 function as a 5-bit I/O port.
64-pin products: P90 to P94 function as a 5-bit I/O port.
80-pin products: P90 to P97 function as an 8-bit I/O port.
100-pin products: P90 to P97 function as an 8-bit I/O port.
128-pin products: P90 to P97 function as an 8-bit I/O port.

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

(2) Control mode

P90 to P97 function as timer I/O, output pins for the sound generator, stepper motor controller/driver outputs/inputs, real-time clock correction clock output, and segment signal outputs for the LCD controller/driver.

(a) TI01 (P94), TI03 (P95), TI05 (P96), TI07 (P97), TI21 (P90), TI23 (P91), TI25 (P92), and TI27 (P93)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers01, 03, 05, 07, 21, 23, 25, and 27.

(b) TO01 (P94), TO03 (P95), TO05 (P96), TO07 (P97), TO21 (P90), TO23 (P91), TO25 (P92), and TO27 (P93)

These are the timer output pins of 16-bit timers01, 03, 05, 07, 21, 23, 25, and 27.

(c) SGO (P93)

This is the output pin for the sound generator.

(d) SGOA (P92)

This is the amplitude PWM output pin for the sound generator.

(e) SGOF (P93)

This is the frequency output pin for the sound generator.

(f) SM31 (P90) to SM34 (P93) and SM41 (P94) to SM44 (P97) (80-pin and 100-pin product)

These are the output pins for the stepper motor controller/driver.

(g) ZPD34 (P93), ZPD44 (P97) (80-pin and 100-pin product)

These are the Zero Point Detection (ZPD) input pins for the stepper motor controller/driver.

(h) RTC1HZ (P94)

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

(i) SEG40 (P90) to SEG47 (P97)

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

2.2.11 P100 to P107 (port10)

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: Not provided
128-pin products: P100 to P107 function as an 8-bit I/O port.

These pins also function as timer I/O, and segment signal outputs for the LCD controller/driver.

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

(1) Port mode

P100 to P107 function as an 8-bit I/O port.

P100 to P107 can be set to input or output port in 1-bit units using port mode register 10 (PM10). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 9 (PU10).

(2) Control mode

P100 to P107 function as timer I/O, and segment signal outputs for the LCD controller/driver.

(a) TI24 (P100) to TI27 (P103) , and TI01 (P104), TI02 (P105), TI05 (P106), and TI06 (P107)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 24 to 27, and 01, 02, 05, 06.

(b) TO24 (P100) to TO27 (P103) , and TO01 (P104), TO02 (P105), TO05 (P106), and TO06 (P107)

These are the timer output pins of 16-bit timers 24 to 27, and 01, 02, 05, 06.

(c) SEG36 (P100) to SEG39 (P103), and SEG44 (P104) to SEG47 (P107)

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

2.2.12 P110 to P117 (port 11)

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: Not provided
128-pin products: P110 to P117 function as an 8-bit I/O port.

These pins also function as LCD Bus I/F data lines I/O, timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

Input can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer in 1-bit units, using port input mode register 0 (PIM0).

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

(1) Port mode

P110 to P117 function as an 8-bit I/O port.

P110 to P117 can be set to input or output port in 1-bit units using port mode register 0 (PM11). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU11).

(2) Control mode

P00 to P07 function as LCD Bus I/F data lines I/O, timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

(a) DBD0 (P110) to DBD7 (P117)

These are the I/O pin of LCD Bus I/F data lines 0 to 7.

(b) TI00 (P110), TI02 (P111), TI04 (P112), TI06 (P113), TI07 (P114), TI10 (P115), TI12 (P116), and TI20 (P117)

These are the external count clock input /capture trigger to 16-bit timer 00, 02, 04, 07, 10, 12, 20.

(b) TO00 (P110), TO02 (P111), TO04 (P112), TO06 (P113), TO07 (P114), TO10 (P115), TO12 (P116), and TO20 (P117)

These are the timer output pins of 16-bit timers 00, 02, 04, 07, 10, 12, 20.

(c) SO00 (P112)

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

(d) SI00 (P111)

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

(e) SCK00 (P110)

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

(f) SEG28 (P117) to SEG35 (P110)

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

(g) TxD0 (P112)

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

(h) RxD0 (P111)

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

2.2.13 P121 to P127 (port12)

48-pin products: P121 to P122 function as a 2-bit Input port.
64-pin products: P121 to P124 function as a 4-bit Input port.
80-pin products: P121 to P124 function as a 4-bit Input port.
100-pin products: P121 to P124 function as a 4-bit Input port.
128-pin products: P121 to P124 function as a 4-bit Input port, P125 to P127 function as a 3-bit I/O port.

These pins also function as external clock input for main system clock, external clock input for subsystem clock, timer I/O, and segment signal outputs for the LCD controller/driver.

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

(1) Port mode

48-pin products: P121 to P122 function as a 2-bit Input port.
64-pin products: P121 to P124 function as a 4-bit Input port.
80-pin products: P121 to P124 function as a 4-bit Input port.
100-pin products: P121 to P124 function as a 4-bit Input port.
128-pin products: P121 to P124 function as a 4-bit Input port, P125 to P127 function as a 3-bit I/O port.

(2) Control mode

P121 to P127 function as external clock input for main system clock, external clock input for subsystem clock, timer I/O, and segment signal outputs for the LCD controller/driver.

(a) X1 (P121), X2 (P122)

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

(b) EXCLK (P122)

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

(c) XT1 (P123), XT2 (P124)

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

(d) TI12 (P125), TI14 (P126), and TI16 (P127)

These are the external count clock input /capture trigger to 16-bit timer 12, 14, 16.

(e) TO12 (P125), TO14 (P126), and TO16 (P127)

These are the timer output pins of 16-bit timers 12, 14, 16.

(f) SEG23 (P127) to SEG25 (P125)

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

2.2.14 P130 to P137 (port13)

48-pin products: P137 functions as a 1-bit Input port.
64-pin products: P137 functions as a 1-bit Input port.
80-pin products: P137 functions as a 1-bit Input port.
100-pin products: P130 functions as a 1-bit Output port, P131 to P136 function as a 6-bit I/O port, and P137 functions as a 1-bit Input port.
128-pin products: P130 functions as a 1-bit Output port, P131 to P136 function as a 6-bit I/O port, and P137 functions as a 1-bit Input port.

These pins also function as timer I/O, output pins for the sound generator, serial interface data I/O ,and segment signal outputs for the LCD controller/driver.

Input to the P135 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer, using port input mode register 13 (PIM13).

Output from the P136 pin can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance), using port output mode register (POM).

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

(1) Port mode

48-pin products: P137 functions as a 1-bit Input port.
64-pin products: P137 functions as a 1-bit Input port.
80-pin products: P137 functions as a 1-bit Input port.
100-pin products: P130 functions as a 1-bit Output port, P131 to P136 function as a 6-bit I/O port, and P137 functions as a 1-bit Input port.
128-pin products: P130 functions as a 1-bit Output port, P131 to P136 function as a 6-bit I/O port, and P137 functions as a 1-bit Input port.

P131 to P136 can be set to input or output port, in 1-bit units using port mode register 13 (PM13). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 13 (PU13).

(2) Control mode

P131 to P137 function as timer I/O, output pins for the sound generator, serial interface data I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

(a) RESOUT (P130)

This is the reset output signal pin.

(b) TI00 (P136), TI20 (P132), TI21 (P131), TI22 (P133), TI24 (P134), and TI26 (P135)

These are the pins for inputting an external count clock/capture trigger to 16-bit timers00, 20, 21, 22, 24, and 26.

(c) TO00 (P136), TO20 (P132), TO21 (P131), TO22 (P133), TO24 (P134), and TO26 (P135)

These are the timer output pins of 16-bit timers00, 20, 21, 22, 24, and 26.

(d) SGO (P135)

This is the output pin for the sound generator.

(e) SGOA (P134)

This is the amplitude PWM output pin for the sound generator.

(f) SGOF (P135)

This is the frequency output pin for the sound generator.

(g) SO10 (P131)

This is a serial data output pin of serial interface CSI10.

(h) SI10 (P132)

This is a serial data input pin of serial interface CSI10.

(i) SCK10 (P133)

This is a serial clock I/O pin of serial interface CSI10.

(j) LTxD1 (P131)

This is a Serial data output from LIN-UART.

(k) LRxD0 (P132)

This is a Serial data input to LIN-UART.

(l) SCL11 (P136)

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

(m) CTxD1 (P134) (R5F10DPJxFB and R5F10DSJxFB only)

This is a CAN serial transmit data output pin of aFCAN1.

(n) CRxD1 (P135) (R5F10DPJxFB and R5F10DSJxFB only)

This is a CAN serial receive data input pin of aFCAN1.

(o) INTP5 (P137)

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

(p) INTPLR1 (P132)

This is an external interrupt request input for which the valid edge for LIN-UART1.

(q) SEG48 (P136)

This is a segment signal output pin for the LCD controller/driver.

2.2.15 P140 (port14)

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: P140 functions as a 1-bit I/O port.
128-pin products: P140 functions as a 1-bit I/O port.

This pin also functions as timer I/O.

(1) Port mode

P140 functions as a 1-bit I/O port.

P140 can be set to input or output port, 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 a 1-bit I/O port.

(a) TI11 (P140)

This is a pin for inputting an external count clock/capture trigger to 16-bit timers11.

(b) TO11 (P140)

This is a timer output pin of 16-bit timers11.

2.2.16 P150 to P152 (port15)

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: P150 functions as a 1-bit I/O port.
128-pin products: P150 to P152 functions as a 3-bit I/O port.

This pin also functions as A/D converter analog input.

(1) Port mode

100-pin products: P150 functions as a 1-bit I/O port.
128-pin products: P150 to P152 functions as a 3-bit I/O port.

P150 to P152 can be set to input or output port, using port mode register 15 (PM15).

(2) Control mode

P150 to P152 function as A/D converter analog inputs.

(a) ANI8 (P150) to ANI10 (P152)

These are the analog input pins of A/D converter.

2.2.17 COM0 to COM3

These are the common signal output pins for the LCD controller/driver.

2.2.18 V_{DD}, EV_{DD0}, EV_{DD1}, SMV_{DD0}, SMV_{DD1}, V_{SS}, EV_{SS0}, EV_{SS1}, SMV_{SS0}, SMV_{SS1}

(1) V_{DD}, EV_{DD}, EV_{DD0}, EV_{DD1}

When using the 48-pin products, V_{DD} is the positive power supply pin for P20 to P23, P27, P121 to P122, P137, $\overline{\text{RESET}}$. When using the 64-pin products, V_{DD} is the positive power supply pin for P20 to P23, P27, P121 to P124, P137, $\overline{\text{RESET}}$ When using the 80-pin products, V_{DD} is the positive power supply pin for P20 to P27, P121 to P124, P137, $\overline{\text{RESET}}$. When using the 100-pin products, V_{DD} is the positive power supply pin for P20 to P27, P121 to P124, P137, P150, $\overline{\text{RESET}}$.

EV_{DD}, EV_{DD0}, EV_{DD1} are the positive power supply pins for the other than V_{DD}, SMV_{DD}, SMV_{DD0}, SMV_{DD1}.

(2) SMV_{DD}, SMV_{DD0}, SMV_{DD1}

When using the 48-pin products, SMV_{DD} is the positive power supply pin for P80 to P83, P90 to P94. When using the 64-pin products, SMV_{DD} is the positive power supply pin for P80 to P87, P90 to P94. When using the 80-pin, 100-pin products, SMV_{DD0}, SMV_{DD1} are the positive power supply pins for P80 to P87, P90 to P97.

(3) V_{SS}, EV_{SS}, EV_{SS0}, EV_{SS1}

When using the 48-pin products, V_{SS} is the ground potential pin for P20 to P23, P27, P121 to P122, P137, $\overline{\text{RESET}}$. When using the 64-pin products, V_{SS} is the ground potential pin for P20 to P23, P27, P121 to P124, P137, $\overline{\text{RESET}}$. When using the 80-pin products, V_{SS} is the ground potential pin for P20 to P27, P121 to P124, P137, $\overline{\text{RESET}}$. When using the 100-pin products, V_{SS} is the ground potential pin for P20 to P27, P121 to P124, P137, P150, $\overline{\text{RESET}}$.

EV_{SS}, EV_{SS0}, EMV_{SS1} are the ground potential pins for the other than V_{SS}, SMV_{SS}, SMV_{SS0}, SMV_{SS1}.

(4) SMV_{SS}, SMV_{SS0}, SMV_{SS1}

When using the 48-pin products, SMV_{SS} is the ground potential pin for P80 to P83, P90 to P94. When using the 64-pin products, SMV_{SS} is the ground potential pin for P80 to P87, P90 to P94. When using the 80-pin, 100-pin products, SMV_{SS0}, SMV_{SS1} are the ground potential pins for P80 to P87, P90 to P97.

2.2.19 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

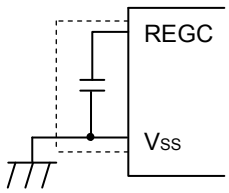
When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.

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

2.2.20 REGC

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

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



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

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

Table 2-7 shows the types of pin I/O circuits and the recommended connections of unused pins.

Table 2-7. Connection of Unused Pins (1/15)

(a) R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB, R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00/TO00/CTxD0/SEG14	17-W	I/O	Input: Independently connect to EVSS via a resistor. Output: Leave open.
P01/TI01/TO01/CRxD0/SEG15	17-AD		
P10/LTxD1/SCK00/TI10/TO10/ INTP4/SEG31	17-AD		
P11/LRxD1/INTPLR1/SI00/TI11/TO11/ SEG30			
P12/SO00/TI12/TO12/INTP2/SEG29	17-W		
P13/SO01/TI13/TO13/SEG25			
P14/TI14/TO14/LRxD0/INTPLR0/SEG24			
P20/AVREFP/ANI0	11-AA		Input: Independently connect to VDD or VSS via a resistor. Output: Leave open.
P21/AVREFM/ANI1			
P22/ANI2, P23/ANI3, P27/ANI7	11-Z		
P30/TI20/TO20/SCL11/SEG6	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P31/TI21/TO21/SDA11/SEG7	17-AD		
P33/TI23/TO23/SI00/SEG9	17-W		
P40/TOOL0	5-AH[Note]		Input: Independently connect to EVDD or leave open. Output: Leave open.
P54/TI14/TO14/SO01/SEG2	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P55/TI15/TO15/SI01/SEG3	17-AD		
P56/TI16/TO16/SCK01/SEG4			
P57/TI17/TO17 /SEG5			
P60/SCL11/TI20/TO20/INTP1	5-AH		Input: Independently connect to EVDD or EVSS via a resistor. Output: Leave open.
P61/SDA11/TI21/TO21/INTP3	5-BA		
P72/ADTRG/SGOA/SEG1	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P73/SGO/SGOF/SEG0			
P74/SCK01/TI23/TO23/SEG26			
P75/PCL/SI01/TI22/TO22/SEG27			
P80/SM11/TI01/TO01/SEG32	17-W		Input: Independently connect to SMVSS via a resistor. Output: Leave open.
P81/SM12/TI03/TO03/SEG33			
P82/SM13/TI05/TO05/SEG34			
P83/SM14/ZPD14/TI07/TO07/SEG35	17-AE		
P90/SM31/TI21/TO21/SEG40	17-W		
P91/SM32/TI23/TO23/SEG41			
P92/SM33/TI25/TO25/SGOA/SEG42			
P93/SM34/ZPD34/TI27/TO27/SGO /SGOF/SEG43	17-AE		
P94/SM41/TI01/TO01/RTC1HZ/SEG44	17-W		
P121/X1	37-C	Input	Independently connect to VDD or VSS via a resistor.
P122/X2/EXCLK			
P137/INTP5	2-H		

Table 2-7. Connection of Unused Pins (2/15)

(a) R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB, R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
COM0 to COM3	18-G	Output	Leave open
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to V_{DD}.
REGC	-	-	Connect to VSS via capacitor (0.47 to 1 μF).

Note Input NAND is Schmitt1. In only OCD mode, NOD can be selected

Table 2-7. Connection of Unused Pins (3/15)

(b) R5F10CLDxFB, R5F10DLDxFB, R5F10DLExFB (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00/TO00/CTxD0/SEG14	17-W	I/O	Input: Independently connect to EVSS via a resistor. Output: Leave open.
P01/TI01/TO01/CRxD0/SEG15	17-AD		
P02/SO00/TI02/TO02/TI12/TO12/SEG16	17-W		
P03/SI00/TI03/TO03/TI13/TO13/SEG17			
P04/SCK00/TI04/TO04/TI14/TO14/SEG18			
P05/TI05/TO05/TI15/TO15/SEG19			
P07/TI07/TO07/TI17/TO17/SEG21			
P10/LTxD1/SCK00/TI10/TO10/INTP4/ SEG31	17-AD		
P11/LRxD1/INTPLR1/SI00/TI11/TO11/ SEG30			
P12/SO00/TI12/TO12/INTP2/SEG29	17-W		
P13/SO01/TI13/TO13/SEG25			
P14/TI14/TO14/LRxD0/INTPLR0/ SEG24			
P15/TI15/TO15/LTxD0/RTC1HZ/SEG23			
P17/TI17/TO17/INTP0/SEG28	17-AD		
P20/AVREFP/ANI0	11-AA		Input: Independently connect to VDD or VSS via a resistor. Output: Leave open.
P21/AVREFM/ANI1			
P22/ANI2, P23/ANI3, P27/ANI7	11-Z		
P30/TI20/TO20/SCL11/SEG6	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P31/TI21/TO21/SDA11/SEG7	17-AD		
P32/TI22/TO22/SEG8	17-W		
P33/TI23/TO23/SEG9			
P40/TOOL0	5-AH[Note]		Input: Independently connect to EVDD or leave open. Output: Leave open.
P54/TI14/TO14/SO01/SEG2	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P55/TI15/TO15/SI01/SEG3	17-AD		
P56/TI16/TO16/SCK01/SEG4			
P57/TI17/TO17/ SEG5			
P60/SCL11/TI20/TO20/INTP1	5-AH		Input: Independently connect to EVDD or EVSS via a resistor. Output: Leave open.
P61/SDA11/TI21/TO21/INTP3	5-BA		

Note Input NAND is Schmitt1. In only OCD mode, NOD can be selected

Table 2-7. Connection of Unused Pins (4/15)

(b) R5F10CLDxFB, R5F10DLDxFB, R5F10DLExFB (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P70/CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD	5-BA	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P71/CTxD0/LTxD0/TOOLTXD	5-AH		
P72/ADTRG/SGOA/SEG1	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P73/SGO/SGOF/SEG0			
P74/SCK01/TI23/TO23/SEG26			
P75/PCL/SI01/TI22/TO22/SEG27			
P80/SM11/TI01/TO01/SEG32	17-W		Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P81/SM12/TI03/TO03/SEG33			
P82/SM13/TI05/TO05/SEG34			
P83/SM14/ZPD14/TI07/TO07/SEG35	17-AE		
P84/SM21/TI11/TO11/SEG36	17-W		
P85/SM22/TI13/TO13/SEG37			
P86/SM23/TI15/TO15/SEG38			
P87/SM24/ZPD24/TI17/TO17/SEG39	17-AE		
P90/TI21/TO21/SEG40	17-W		
P91/TI23/TO23/SEG41			
P92/TI25/TO25/SGOA/SEG42			
P93/TI27/TO27/SGO/SGOF/SEG43			
P94/TI01/TO01/RTC1HZ/SEG44			
P121/X1	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK			
P123/XT1			
P124/XT2			
P137/INTP5	2-H		
COM0 to COM3	18-G	Output	Leave open
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to V_{DD}.
REGC	-	-	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Table 2-7. Connection of Unused Pins (5/15)
(c) R5F10CMDxFB, R5F10CMExFB, R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00/TO00/CTxD0/SEG14	17-W	I/O	Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P01/TI01/TO01/CRxD0/SEG15	17-AD		
P02/SO00/TI02/TO02/TI12/TO12/ SEG16	17-W		
P03/SI00/TI03/TO03/TI13/TO13/SEG17			
P04/SCK00/TI04/TO04/TI14/TO14/ SEG18			
P05 /TI05/TO05/TI15/TO15/ SEG19			
P06/TI06/TO06/TI16/TO16/SEG20			
P07/TI07/TO07/TI17/TO17/SEG21			
P10/LTxD1/SCK00/TI10/TO10/INTP4/ SEG31	17-AD		
P11/LRxD1/INTPLR1/SI00/TI11/TO11/ SEG30			
P12/SO00/TI12/TO12/INTP2/SEG29	17-W		
P13/SO01/TI13/TO13/SEG25			
P14/TI14/TO14/LRxD0/INTPLR0/ SEG24			
P15/TI15/TO15/LTxD0/RTC1HZ/SEG23			
P16/TI16/TO16/SEG22			
P17/ TI17/TO17/INTP0/SEG28	17-AD		
P20/AV_{REFP}/ANI0	11-AA		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P21/AV_{REFM}/ANI1			
P22/ANI2 to P27/ANI7	11-Z		
P30/TI20/TO20/SCL11/SEG6	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P31/TI21/TO21/SDA11/SEG7	17-AD		
P32/TI22/TO22/SO00/SEG8	17-W		
P33/TI23/TO23/SI00/SEG9			
P34/TI24/TO24/SCK00/SEG10			
P35/TI25/TO25/SEG11			
P36/TI26/TO26/SEG12			
P37/TI27/TO27/SEG13			
P40/TOOL0	5-AH[Note]		Input: Independently connect to EV_{DD} or leave open. Output: Leave open.
P54/TI14/TO14/SO01/SEG2	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P55/TI15/TO15/SI01/SEG3	17-AD		
P56/TI16/TO16/SCK01/SEG4			
P57/TI17/TO17/SEG5			
P60/SCL11/TI20/TO20/INTP1	5-AH		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P61/SDA11/TI21/TO21/INTP3	5-BA		
P65/TI25/TO25	5-AH		
P66/TI24/TO24/PCL			

Note Input NAND is Schmitt1. In only OCD mode, NOD can be selected

Table 2-7. Connection of Unused Pins (6/15)
(c) R5F10CMDxFB, R5F10CMExFB, R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P70/CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD	5-BA	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P71/CTxD0/LTxD0/TOOLTXD	5-AH		
P72/ADTRG/SGOA/SEG1	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P73/SGO/SGOF/SEG0			
P74/SCK01/TI23/TO23/SEG26			
P75/PCL/SI01/TI22/TO22/SEG27			
P80/SM11/TI01/TO01/SEG32	17-W		Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P81/SM12/TI03/TO03/SEG33			
P82/SM13/TI05/TO05/SEG34			
P83/SM14/ZPD14/TI07/TO07/SEG35	17-AE		
P84/SM21/TI11/TO11/SEG36	17-W		
P85/SM22/TI13/TO13/SEG37			
P86/SM23/TI15/TO15/SEG38			
P87/SM24/ZPD24/TI17/TO17/SEG39	17-AE		
P90/SM31/TI21/TO21/SEG40	17-W		Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P91/SM32/TI23/TO23/SEG41			
P92/SM33/TI25/TO25/SGOA/SEG42			
P93/SM34/ZPD34/TI27/TO27/SGO/SGOF/SEG43	17-AE		
P94/SM41/TI01/TO01/RTC1HZ/SEG44	17-W		
P95/SM42/TI03/TO03/SEG45			
P96/SM43/TI05/TO05/SEG46			
P97/SM44/ZPD44/TI07/TO07/SEG47	17- AE		
P121/X1	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK			
P123/XT1			
P124/XT2			
P137/INTP5	2-H		
COM0 to COM3	18-G	Output	Leave open
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to V_{DD}.
REGC	-	-	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Table 2-7. Connection of Unused Pins (7/15)

(d) R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00/TO00/CTxD0/SEG14	17-W	I/O	Input: Independently connect to EV_{SS} via a resistor.
P01/TI01/TO01/CRxD0/SEG15	17-AD		Output: Leave open.
P02/SO00/TI02/TO02/TI12/TO12/ SEG16	17-W		
P03/SI00/TI03/TO03/TI13/TO13/ SEG17			
P04/SCK00/TI04/TO04/TI14/TO14/ SEG18			
P05/TI05/TO05/TI15/TO15/ SEG19			
P06/TI06/TO06/TI16/TO16/SEG20			
P07/TI07/TO07/TI17/TO17/SEG21			
P10/LTxD1/SCK00/TI10/TO10/ INTP4/SEG31	17-AD		
P11/LRxD1/INTPLR1/SI00/TI11/TO11/ SEG30			
P12/SO00/TI12/TO12/INTP2/SEG29	17-W		
P13/SO01/TI13/TO13/SEG25			
P14/TI14/TO14/LRxD0/INTPLR0/ SEG24			
P15/TI15/TO15/LTxD0/RTC1HZ/SEG23			
P16/TI16/TO16/SEG22			
P17/TI17/TO17/INTP0/SEG28	17-AD		
P20/AV_{REFP}/ANI0	11-AA		Input: Independently connect to V_{DD} or V_{SS} via a resistor.
P21/AV_{REFM}/ANI1			Output: Leave open.
P22/ANI2 to P27/ANI7	11-Z		
P30/TI20/TO20/SCL11/SEG6	17-W		Input: Independently connect to EV_{SS} via a resistor.
P31/TI21/TO21/SDA11/SEG7	17-AD		Output: Leave open.
P32/TI22/TO22/SO00/SEG8	17-W		
P33/TI23/TO23/SI00/SEG9			
P34/TI24/TO24/SCK00/SEG10			
P35/TI25/TO25/SEG11			
P36/TI26/TO26/SEG12			
P37/TI27/TO27/SEG13			
P40/TOOL0	5-AH[Note]		Input: Independently connect to EV_{DD} or leave open. Output: Leave open.
P50/TI02/TO02/SDA11/SEG49	17- AD		Input: Independently connect to EV_{SS} via a resistor.
P51/TI04/TO04/SCK10/SEG50			Output: Leave open.
P52/TI06/TO06/SI10/SEG51			
P53/TI13/TO13/SO10/SEG52	17-W		
P54/TI14/TO14/SO01/SEG2			
P55/TI15/TO15/SI01/SEG3	17-AD		
P56/TI16/TO16/SCK01/SEG4			
P57/TI17/TO17/SEG5			

Note Input NAND is Schmitt1. In only OCD mode, NOD can be selected

Table 2-7. Connection of Unused Pins (8/15)

(d) R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P60/SCL11/TI20/TO20/INTP1	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P61/SDA11/TI21/TO21/INTP3	5-BA		
P62/TI27/TO27	5-AH		
P63/TI26/TO26	5-BA		
P64/RTC1HZ/TI11/TO11	5-AH		
P65/TI25/TO25			
P66/TI24/TO24/PCL			
P70/CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD	5-BA		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor.
P71/CTxD0/LTxD0/TOOLTXD	5-AH		Output: Leave open.
P72/ADTRG/SGOA/SEG1	17-W		Input: Independently connect to EV_{SS} via a resistor.
P73/SGO/SGOF/SEG0			Output: Leave open.
P74/SCK01/TI23/TO23/SEG26			
P75/PCL/SI01/TI22/TO22/SEG27			
P80/SM11/TI01/TO01/SEG32	17-W(2)		Input: Independently connect to SMV_{SS} via a resistor.
P81/SM12/TI03/TO03/SEG33			Output: Leave open.
P82/SM13/TI05/TO05/SEG34			
P83/SM14/ZPD14/TI07/TO07/SEG35	17-AE		
P84/SM21/TI11/TO11/SEG36	17-W		
P85/SM22/TI13/TO13/SEG37			
P86/SM23/TI15/TO15/SEG38			
P87/SM24/ZPD24/TI17/TO17/SEG39	17-AE		
P90/SM31/TI21/TO21/SEG40	17-W		Input: Independently connect to SMV_{SS} via a resistor.
P91/SM32/TI23/TO23/SEG41			Output: Leave open.
P92/SM33/TI25/TO25/SGOA/SEG42			
P93/SM34/ZPD34/TI27/TO27/SGO/SGOF/SEG43	17-AE		
P94/SM41/TI01/TO01/RTC1HZ/SEG44	17-W		
P95/SM42/TI03/TO03/SEG45			
P96/SM43/TI05/TO05/SEG46			
P97/SM44/ZPD44/TI07/TO07/SEG47	17-AE		
P121/X1	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK			
P123/XT1			
P124/XT2			
P130	3-C	Output	Leave open.
P131/SO10/LTxD1/TI21/TO21	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P132/SI10/LRxD1/INTPLR1/TI20/TO20			
P133/SCK10/TI22/TO22			
P134/SGOA/TI24/TO24			
P135/SGO/SGOF/TI26/TO26	5-BA		
P136/TI00/TO00/SCL11/SEG48	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P137/INTP5	2-H	Input	Independently connect to V_{DD} or V_{SS} via a resistor.

Table 2-7. Connection of Unused Pins (9/15)

(d) R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P140/TI11/TO11	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P150/ANI8	11-Z		Independently connect to V_{DD} or V_{SS} via a resistor.
COM0 to COM3	18-G	Output	Leave open
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to V_{DD}.
REGC	-	-	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Table 2-7. Connection of Unused Pins (10/15)

(e) R5F10DPJxFB (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00/TO00/CTxD0/SEG14	17-W	I/O	Input: Independently connect to EVSS via a resistor.
P01/TI01/TO01/CRxD0/SEG15	17-AD		Output: Leave open.
P02/SO00/TI02/TO02/TI12/TO12/SEG16	17-W		
P03/SI00/TI03/TO03/TI13/TO13/SEG17			
P04/SCK00/TI04/TO04/TI14/TO14/ SEG18			
P05/TI05/TO05/TI15/TO15/ SEG19			
P06/TI06/TO06/TI16/TO16/SEG20			
P07/TI07/TO07/TI17/TO17/SEG21			
P10/LTxD1/SCK00/TI10/TO10/INTP4/ SEG31	17- AD		
P11/LRxD1/INTPLR1/SI00/TI11/TO11/ SEG30			
P12/SO00/TI12/TO12/INTP2/SEG29	17-W		
P13/SO01/TI13/TO13/SEG25			
P14/TI14/TO14/LRxD0/INTPLR0/ SEG24			
P15/TI15/TO15/LTxD0/RTC1HZ/SEG23			
P16/TI16/TO16/SEG22			
P17/TI17/TO17/INTP0/SEG28	17-AD		
P20/AVREFP/ANI0	11-AA		Input: Independently connect to VDD or VSS via a resistor.
P21/AVREFM/ANI1			Output: Leave open.
P22/ANI2 to P27/ANI7	11-Z		
P30/TI20/TO20/SCL11/SEG6	17-W		Input: Independently connect to EVSS via a resistor.
P31/TI21/TO21/SDA11/SEG7	17-AD		Output: Leave open.
P32/TI22/TO22/SO00/SEG8	17-W		
P33/TI23/TO23/SI00/SEG9			
P34/TI24/TO24/SCK00/SEG10			
P35/TI25/TO25/SEG11			
P36/TI26/TO26/SEG12			
P37/TI27/TO27/SEG13			
P40/TOOL0	5-AH[Note]		Input: Independently connect to EVDD or leave open.
			Output: Leave open.
P50/TI02/TO02/SDA11/SEG49	17-AD		Input: Independently connect to EVSS via a resistor.
P51/TI04/TO04/SCK10/SEG50			Output: Leave open.
P52/TI06/TO06/SI10/SEG51			
P53/TI13/TO13/SO10/SEG52	17-W (1)		
P54/TI14/TO14/SO01/SEG2			
P55/TI15/TO15/SI01/SEG3	17-AD		
P56/TI16/TO16/SCK01/SEG4			
P57/TI17/TO17/SEG5			

Note: Input NAND is Schmitt1. In only OCD mode, NOD can be selected

Table 2-7. Connection of Unused Pins (11/15)

(e) R5F10DPJxFB (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P60/SCL11/TI20/TO20/INTP1	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P61/SDA11/TI21/TO21/INTP3	5-BA		
P62/CTxD1/TI27/TO27	5-AH		
P63/CRxD1/TI26/TO26	5-BA		
P64/RTC1HZ/TI11/TO11	5-AH		
P65/TI25/TO25			
P66/TI24/TO24/PCL			
P70/CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRXD	5-BA		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor.
P71/CTxD0/LTxD0/TOOLTXD	5-AH		Output: Leave open.
P72/ADTRG/SGOA/SEG1	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P73/SGO/SGOF/SEG0			
P74/SCK01/TI23/TO23/SEG26			
P75/PCL/SI01/TI22/TO22/SEG27			
P80/SM11/TI01/TO01/SEG32	17-W		Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P81/SM12/TI03/TO03/SEG33			
P82/SM13/TI05/TO05/SEG34			
P83/SM14/ZPD14/TI07/TO07/SEG35	17-AE		
P84/SM21/TI11/TO11/SEG36	17-W		
P85/SM22/TI13/TO13/SEG37			
P86/SM23/TI15/TO15/SEG38			
P87/SM24/ZPD24/TI17/TO17/SEG39	17-AE		
P90/SM31/TI21/TO21/SEG40	17-W		Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P91/SM32/TI23/TO23/SEG41			
P92/SM33/TI25/TO25/SGOA/SEG42			
P93/SM34/ZPD34/TI27/TO27/SGO/SGOF/SEG43	17-AE		
P94/SM41/TI01/TO01/RTC1HZ/SEG44	17-W		
P95/SM42/TI03/TO03/SEG45			
P96/SM43/TI05/TO05/SEG46			
P97/SM44/ZPD44/TI07/TO07/SEG47	17-AE		
P121/X1	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK			
P123/XT1			
P124/XT2			
P130	3-C	Output	Leave open.
P131/SO10/LTxD1/TI21/TO21	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P132/SI10/LRxD1/INTPLR1/TI20/TO20			
P133/SCK10/TI22/TO22			
P134/SGOA/CTxD1/TI24/TO24			
P135/SGO/SGOF/CRxD1/TI26/TO26	5-BA		
P136/TI00/TO00/SCL11/SEG48	17-W		Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.
P137/INTP5	2-H	Input	Independently connect to V_{DD} or V_{SS} via a resistor.

Table 2-7. Connection of Unused Pins (12/15)

(e) R5F10DPJxFB (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P140/TI11/TO11	5-AH	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P150/ANI8	11-Z		Independently connect to V_{DD} or V_{SS} via a resistor.
COM0 to COM3	18-G	Output	Leave open
$\overline{RESET}$	2	Input	Connect directly or via a resistor to V_{DD}.
REGC	-	-	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Table 2-7. Connection of Unused Pins (13/15)

<R> **(f) R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (1/3)**

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/CTxD0/TI00/TO00/SEG14	17-W	I/O	Input: Independently connect to EVSS via a resistor.
P01/CRxD0/TI01/TO01/SEG15	17-AD		Output: Leave open.
P02/SO00/TxD0/TI02/TO02/TI12/TO12/SEG16	17-W		
P03/RxD0/TI03/TO03/TI13/TO13/SEG17			
P04/SCK00/TI04/TO04/TI14/TO14/SEG18			
P05/SI00/TI05/TO05/TI15/TO15/SEG19			
P06/TI06/TO06/TI16/TO16/SEG20			
P07/TI07/TO07/TI17/TO17/SEG21			
P10/LTxD1/SCK00/TI10/TO10/INTP4/SEG31	17- AD		
P11/LRxD1/INTPLR1/SI00/RxD0/TI11/TO11/ SEG30			
P12/SO00/TxD0/TI12/TO12/INTP2/SEG29	17-W		
P13/SO01/TI13/TO13/SEG25			
P14/LRxD0/INTPLR0/TI14/TO14/SEG24			
P15/LTxD0/RTC1HZ/TI15/TO15/SEG23			
P16/TI16/TO16/SEG22			
P17/TI17/TO17/INTP0/SEG28	17-AD		
P20/AVREFP/ANI0	11-AA		Input: Independently connect to VDD or VSS via a resistor.
P21/AVREFM/ANI1			Output: Leave open.
P22/ANI2 to P27/ANI7	11-Z		
P30/TI20/TO20/SCL11/SEG6	17-W		Input: Independently connect to EVSS via a resistor.
P31/TI21/TO21/SDA11/SEG7	17-AD		Output: Leave open.
P32/TI22/TO22/SO00/TxD0/SEG8	17-W		
P33/TI23/TO23/SI00/RxD0/SEG9			
P34/TI24/TO24/SCK00/SEG10			
P35/TI25/TO25/SEG11			
P36/TI26/TO26/SEG12			
P37/TI27/TO27/SEG13			
P40/TOOL0	5-AH[Note]		Input: Independently connect to EVDD or leave open.
			Output: Leave open.
P41/STOPST/TI04/TO04	5-AH		Input: Independently connect to EVDD or EVSS via a resistor.
			Output: Leave open.
P42/TI10/TO10/SEG7	17-W (1)		Input: Independently connect to EVSS via a resistor.
P43/TI22/TO22/SEG14			Output: Leave open.
P44/TI23/TO23/SEG15			
P45/SEG53			
P46/$\overline{\text{DBWR}}$/SEG27			
P47/$\overline{\text{DBRD}}$/SEG26			
P50/TI02/TO02/SDA11/SEG49	17-AD		Input: Independently connect to EVSS via a resistor.
P51/TI04/TO04/SCK10/SEG50			Output: Leave open.
P52/TI06/TO06/SI10/SEG51			

Note: Input NAND is Schmitt1. In only OCD mode, NOD can be selected

Table 2-7. Connection of Unused Pins (14/15)

(f) R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P53/TI13/TO13/SO10/SEG52	17-W (1)	I/O	Input: Independently connect to EVSS via a resistor. Output: Leave open.
P54/TI14/TO14/SO01/SEG2			
P55/TI15/TO15/SI01/SEG3	17-AD		
P56/TI16/TO16/SCK01/SEG4			
P57/TI17/TO17/SEG5			
P60/SCL11/TI20/TO20/INTP1	5-AH	I/O	Input: Independently connect to EVDD or EVSS via a resistor. Output: Leave open.
P61/SDA11/TI21/TO21/INTP3	5-BA		
P62/CTxD1/TI27/TO27	5-AH		
P63/CRxD1/TI26/TO26	5-BA		
P64/RTC1HZ/TI11/TO11	5-AH		
P65/TI25/TO25			
P66/TI24/TO24/PCL			
P70/CRxD0/LRxD0/INTPLR0/TI03/TO03/TOOLRxD	5-BA		Input: Independently connect to EVDD or EVSS via a resistor. Output: Leave open.
P71/CTxD0/LTxD0/TOOLTxD	5-AH		
P72/SGOA/ADTRG/SEG1	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P73/SGO/SGOF/SEG0			
P74/SCK01/TI23/TO23/SEG26			
P75/SI01/TI22/TO22/SEG27/PCL			
P80/SM11/TI01/TO01/SEG32	17-W		Input: Independently connect to SMVSS via a resistor. Output: Leave open.
P81/SM12/TI03/TO03/SEG33			
P82/SM13/TI05/TO05/SEG34			
P83/SM14/ZPD14/TI07/TO07/SEG35	17-AE		
P84/SM21/TI11/TO11/SEG36	17-W		
P85/SM22/TI13/TO13/SEG37			
P86/SM23/TI15/TO15/SEG38			
P87/SM24/ZPD24/TI17/TO17/SEG39	17-AE		
P90/SM31/TI21/TO21/SEG40	17-W		Input: Independently connect to SMVSS via a resistor. Output: Leave open.
P91/SM32/TI23/TO23/SEG41			
P92/SM33/SGOA/TI25/TO25/SEG42			
P93/SM34/ZPD34/SGO/SGOF/TI27/TO27/SEG43	17-AE		
P94/SM41/RTC1HZ/TI01/TO01/SEG44	17-W		
P95/SM42/TI03/TO03/SEG45			
P96/SM43/TI05/TO05/SEG46			
P97/SM44/ZPD44/TI07/TO07/SEG47	17-AE		
P100/TI24/TO24/SEG36	17-W (1)		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P101/TI25/TO25/SEG37			
P102/TI26/TO26/SEG38			
P103/TI27/TO27/SEG39			
P104/TI01/TO01/SEG44			
P105/TI02/TO02/SEG45			
P106/TI05/TO05/SEG46			
P107/TI06/TO06/SEG47			

Table 2-7. Connection of Unused Pins (15/15)

(f) R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P110/DBD0/SCK00/TI00/TO00/SEG35	17-AD	I/O	Input: Independently connect to EVSS via a resistor. Output: Leave open.
P111/DBD1/SI00/RxD0/TI02/TO02/SEG34			
P112/DBD2/SO00/TxD0/TI04/TO04/SEG33			
P113/DBD3/TI06/TO06/SEG32			
P114/DBD4/TI07/TO07/SEG31			
P115/DBD5/TI10/TO10/SEG30			
P116/DBD6/TI12/TO12/SEG29			
P117/DBD7/TI20/TO20/SEG28			
P121/X1	37-C	Input	Independently connect to VDD or VSS via a resistor.
P122/X2/EXCLK			
P123/XT1			
P124/XT2			
P125/TI12/TO12/SEG25	17-W (1)	I/O	Input: Independently connect to EVSS via a resistor. Output: Leave open.
P126/TI14/TO14/SEG24			
P127/TI16/TO16/SEG23			
P130/RESOUT	3-C	Output	Leave open.
P131/SO10/LTxD1/TI21/TO21	5-AH	I/O	Input: Independently connect to EVDD or EVSS via a resistor. Output: Leave open.
P132/SI10/LRxD1/INTPLR1/TI20/TO20			
P133/SCK10/TI22/TO22			
P134/SGOA/CTxD1/TI24/TO24			
P135/SGO/SGOF/CRxD1/TI26/TO26	5-BA		
P136/TI00/TO00/SCL11/SEG48	17-W		Input: Independently connect to EVSS via a resistor. Output: Leave open.
P137/INTP5	2-H	Input	Independently connect to VDD or VSS via a resistor.
P140/TI11/TO11	5-AH	I/O	Input: Independently connect to EVDD or EVSS via a resistor. Output: Leave open.
P150/ANI8	11-Z		Input: Independently connect to VDD or VSS via a resistor. Output: Leave open.
P151/ANI9			
P152/ANI10			
COM0 to COM3	18-G	Output	Leave open
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to VDD.
REGC	-	-	Connect to VSS via capacitor (0.47 to 1 μF).

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

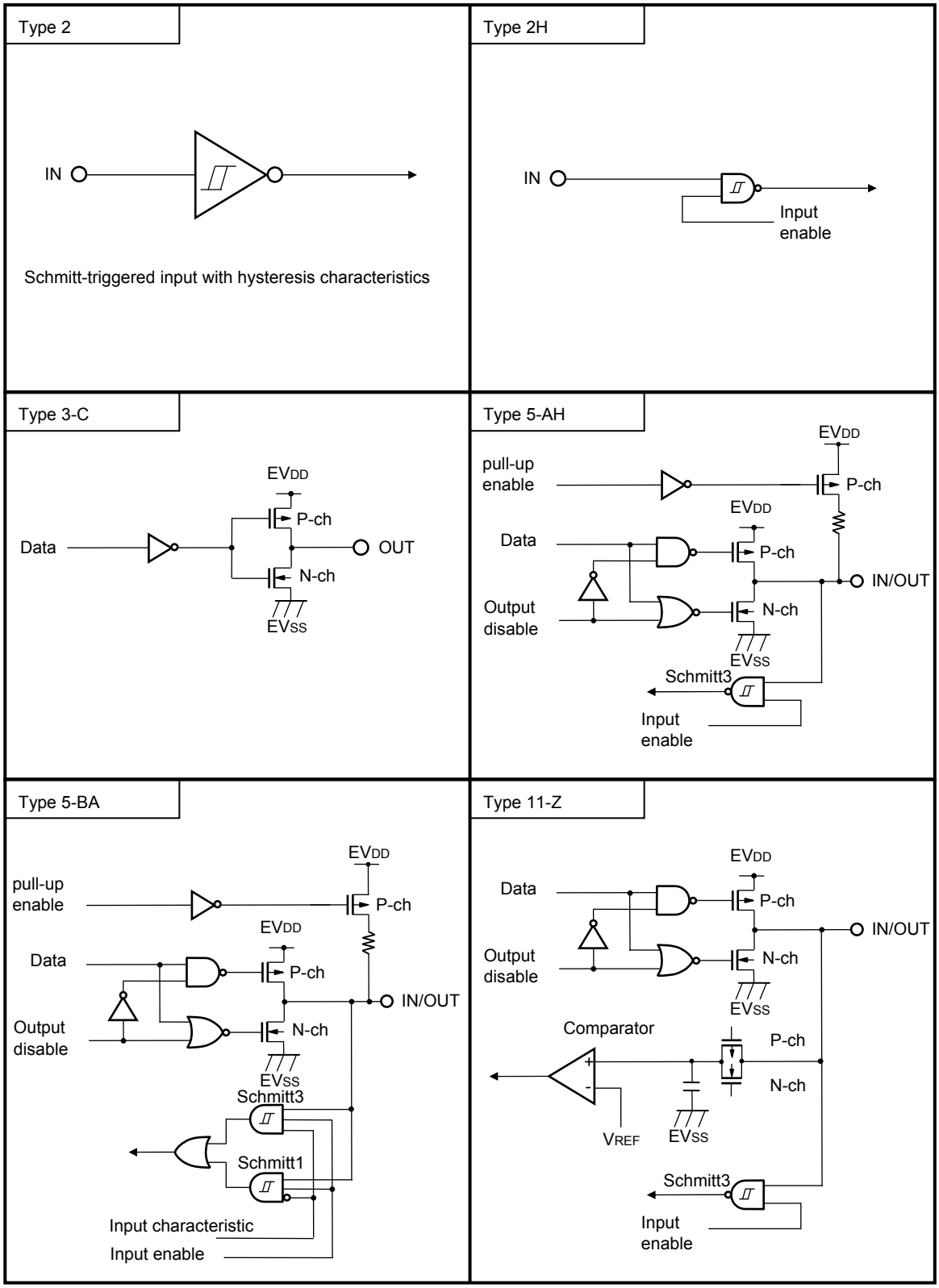


Figure 2-1. Pin I/O Circuit List (2/3)

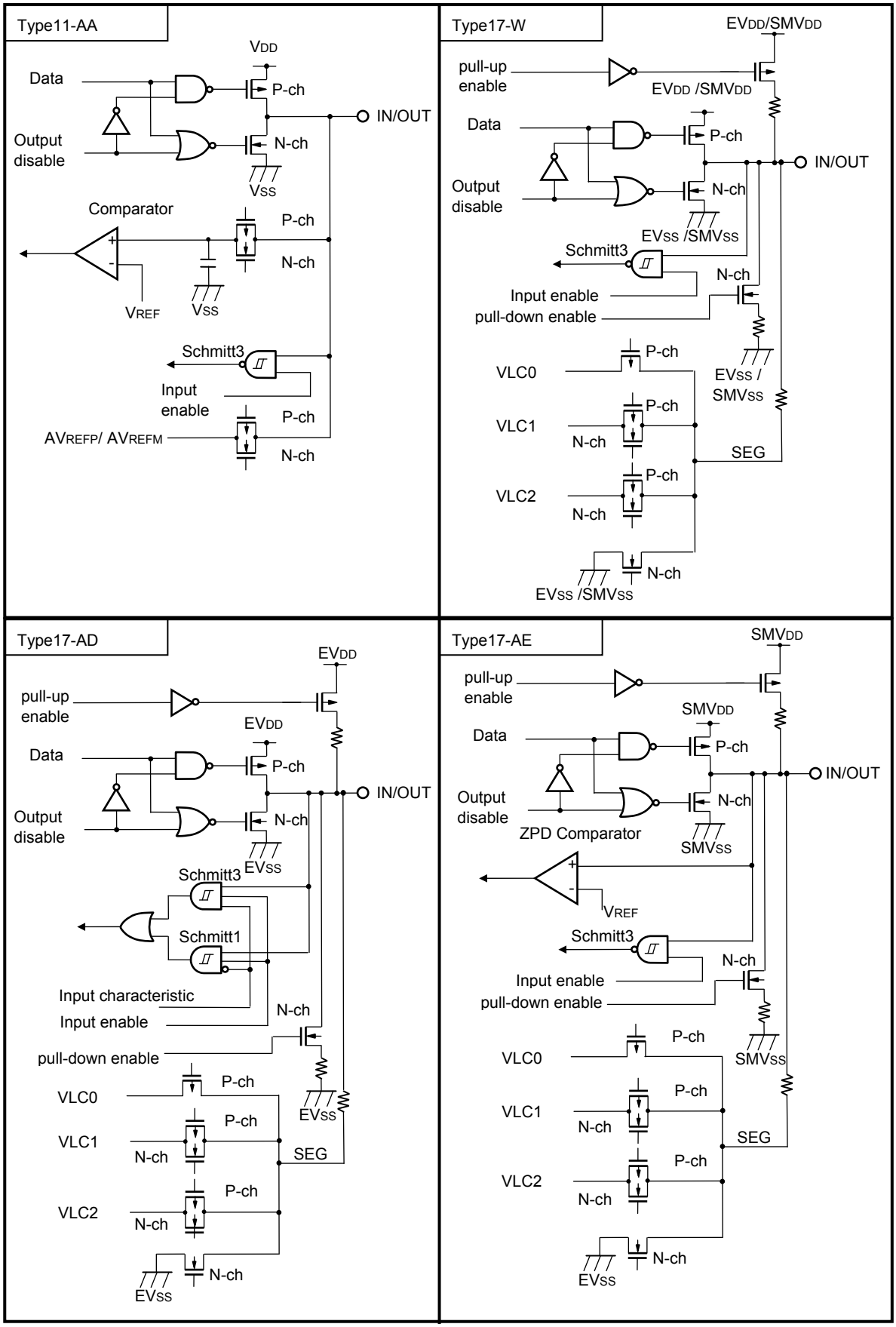
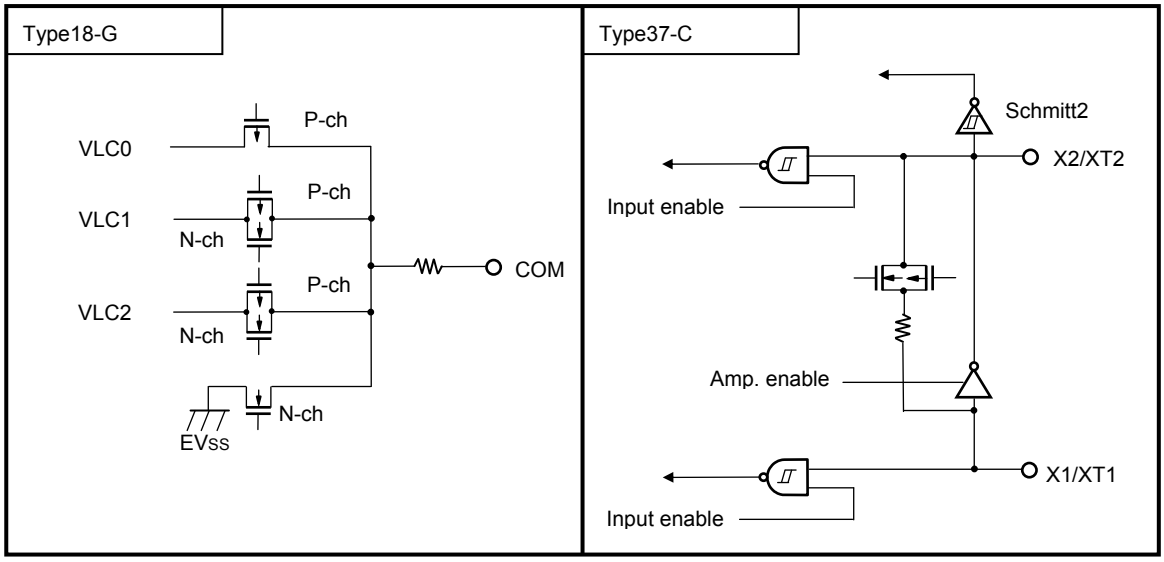


Figure 2-1. Pin I/O Circuit List (3/3)

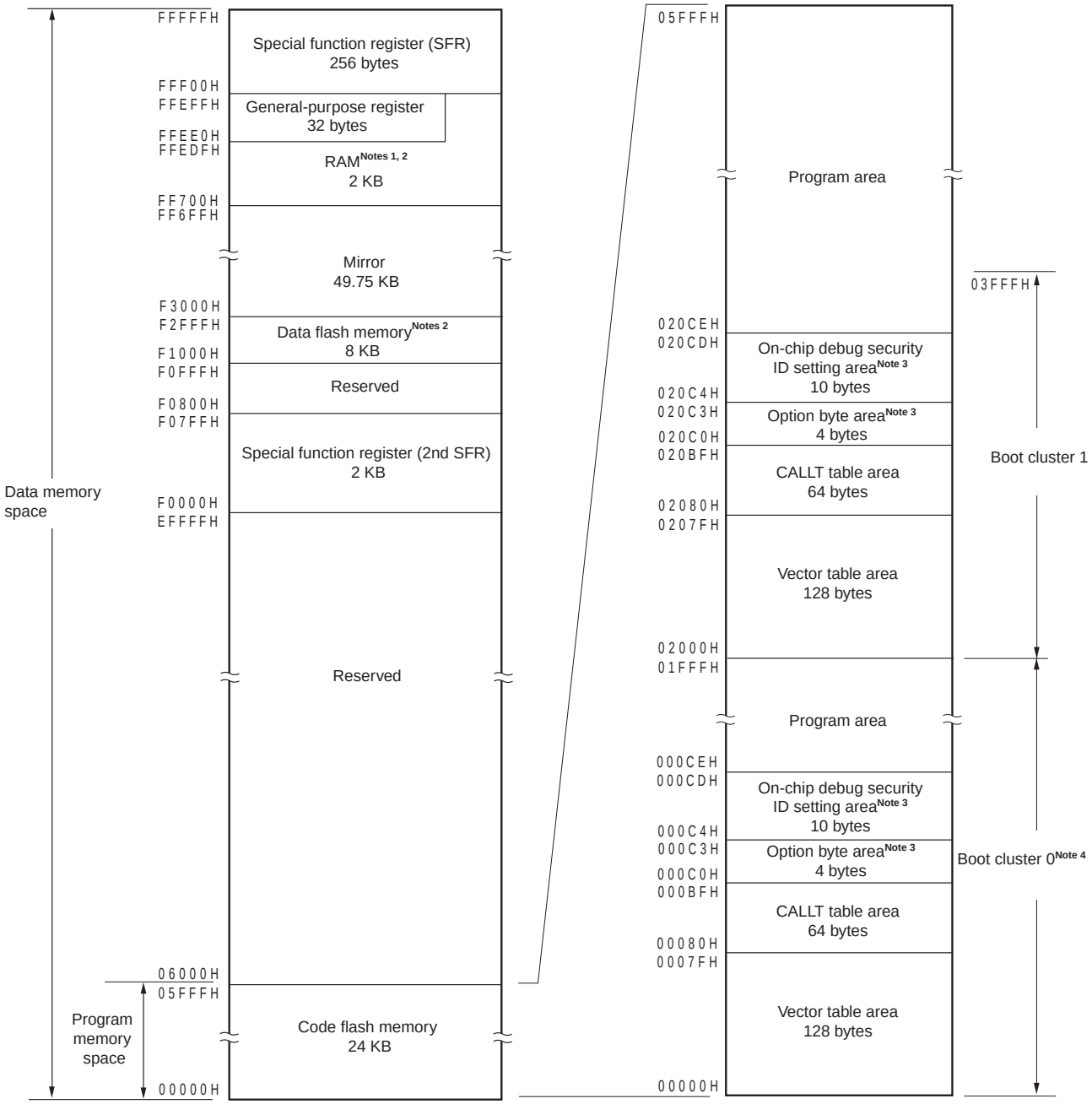


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/D1A can access a 1 MB memory space. Figures 3-1 to 3-9 show the memory maps.

Figure 3-1. Memory Map (R5F10CGBxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

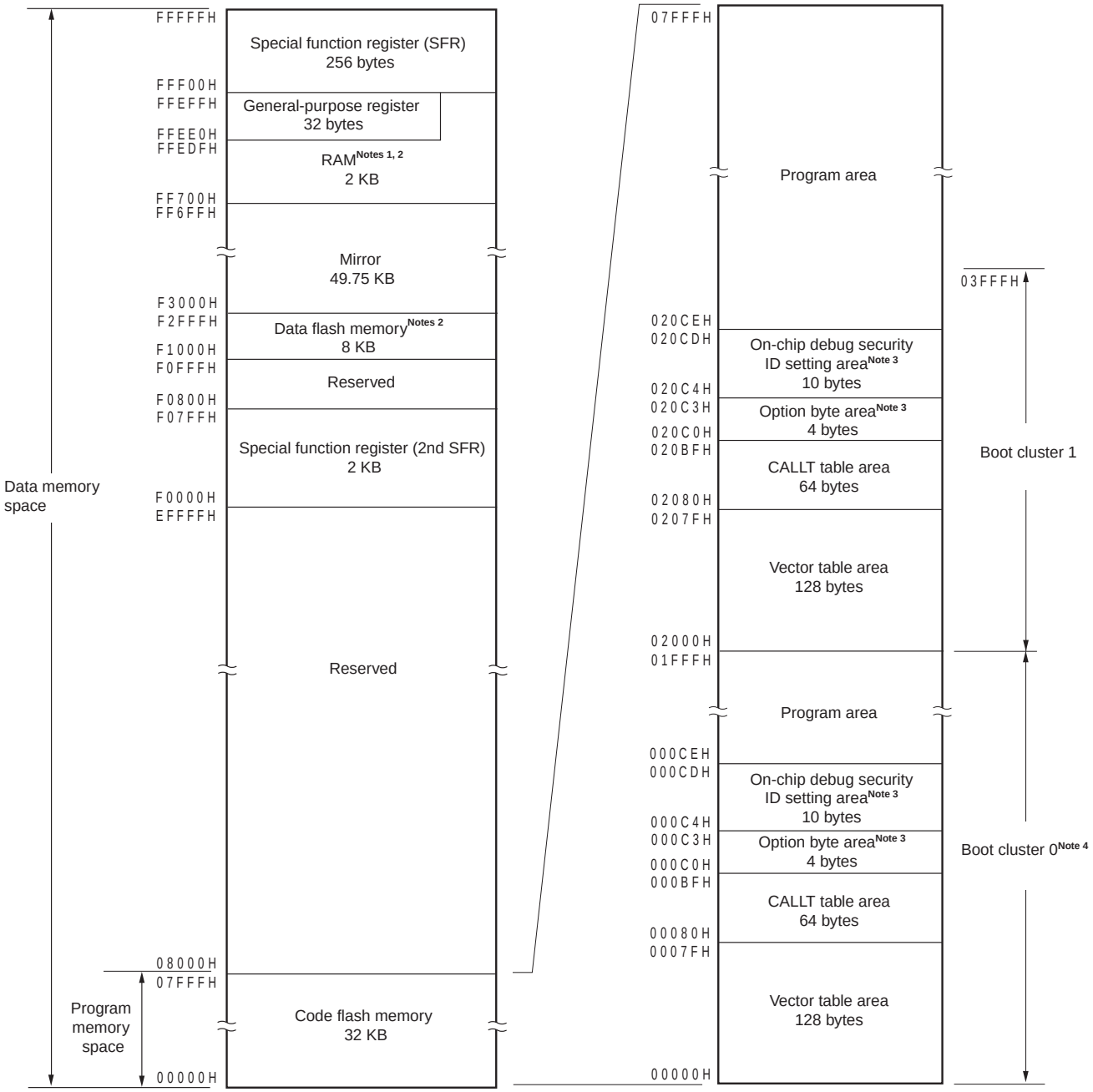
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-2. Memory Map (R5F10CGCxFB, R5F10DGCxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

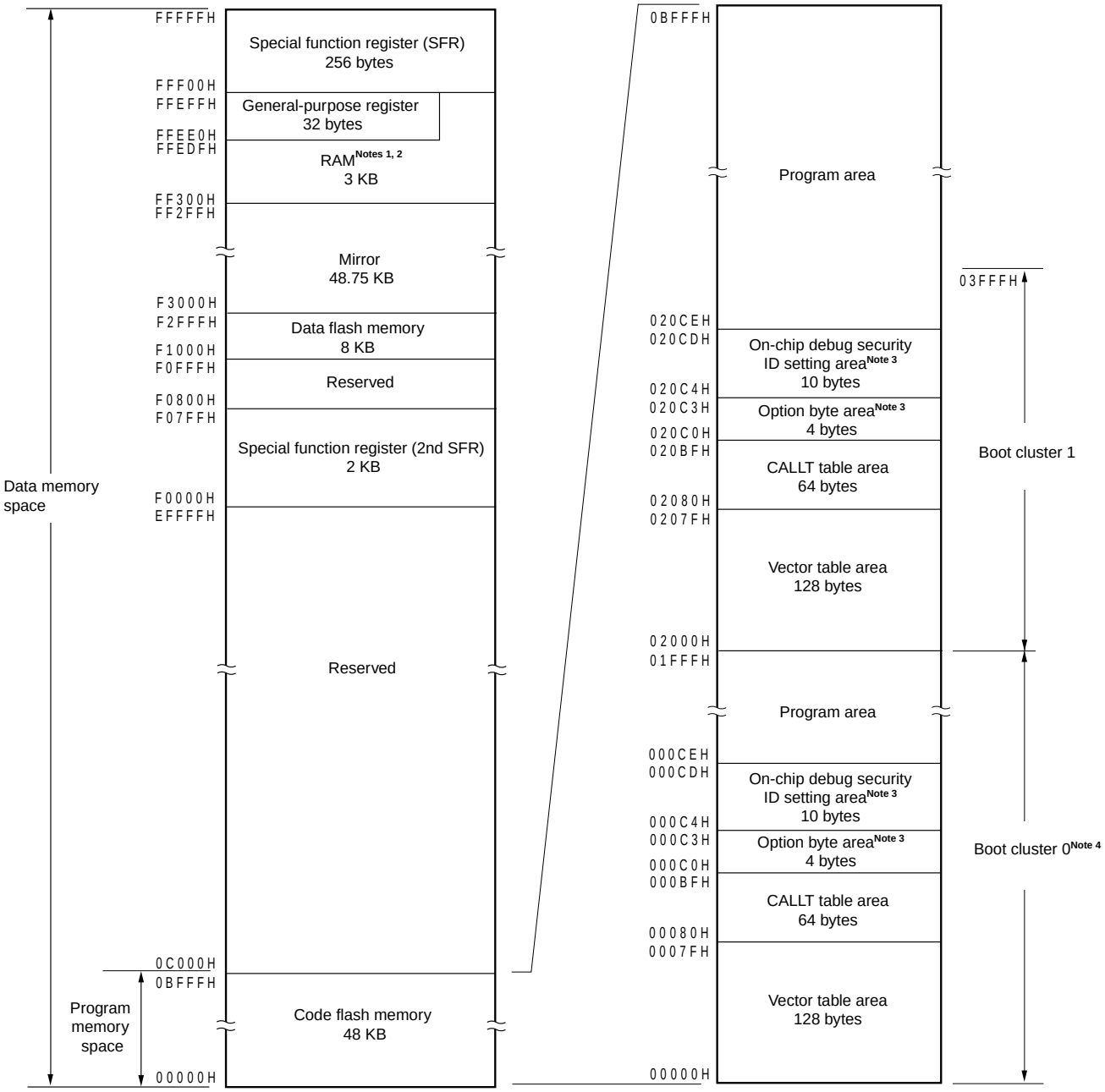
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-3. Memory Map (R5F10CGDxFB, R5F10DGDxFB, R5F10CLDxFB, R5F10DLDxFB, R5F10CMDxFB, R5F10DMDxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

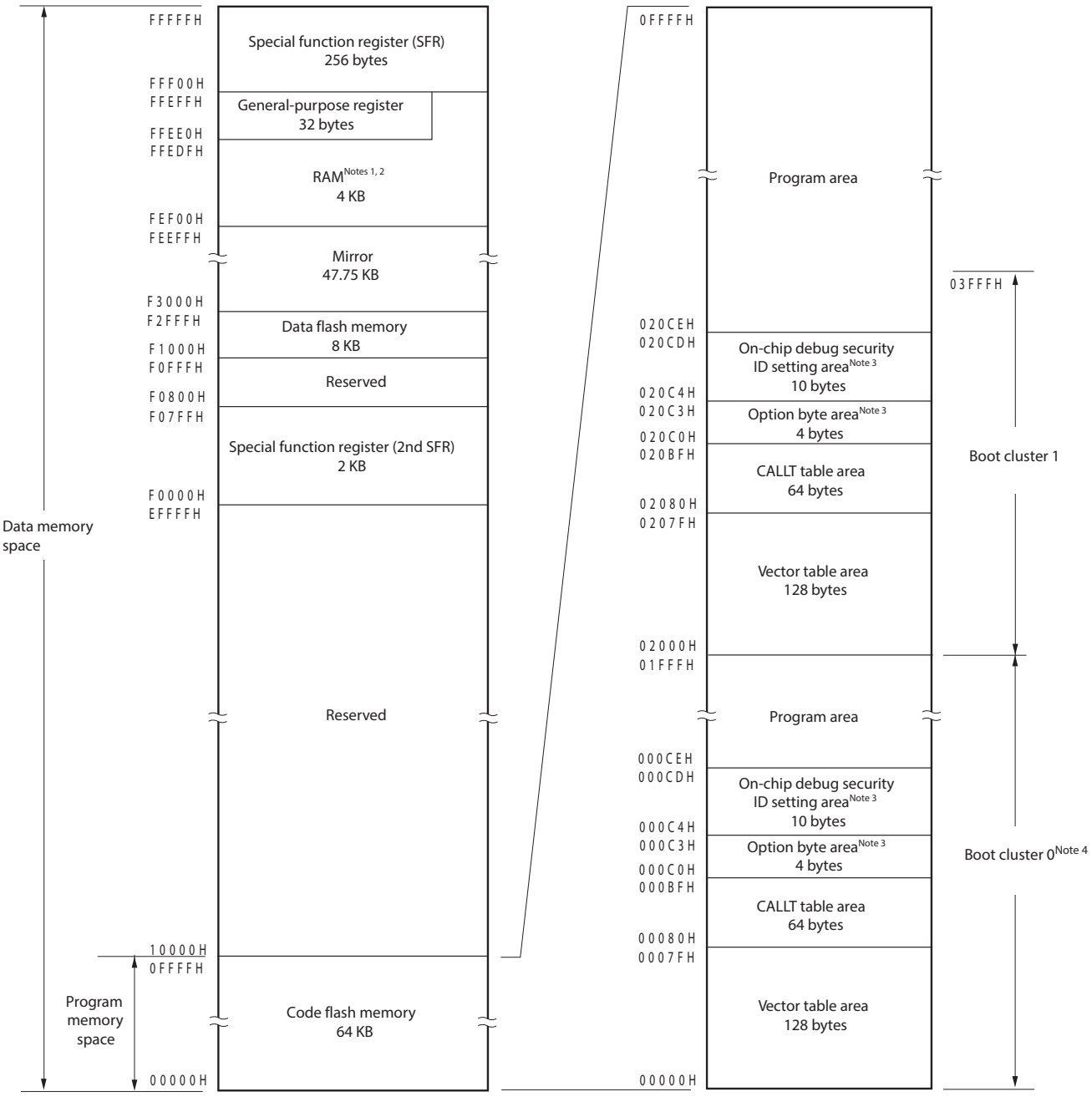
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-4. Memory Map (R5F10DGExFB, R5F10DLExFB, R5F10CMExFB, R5F10DMExFB, R5F10DPExFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

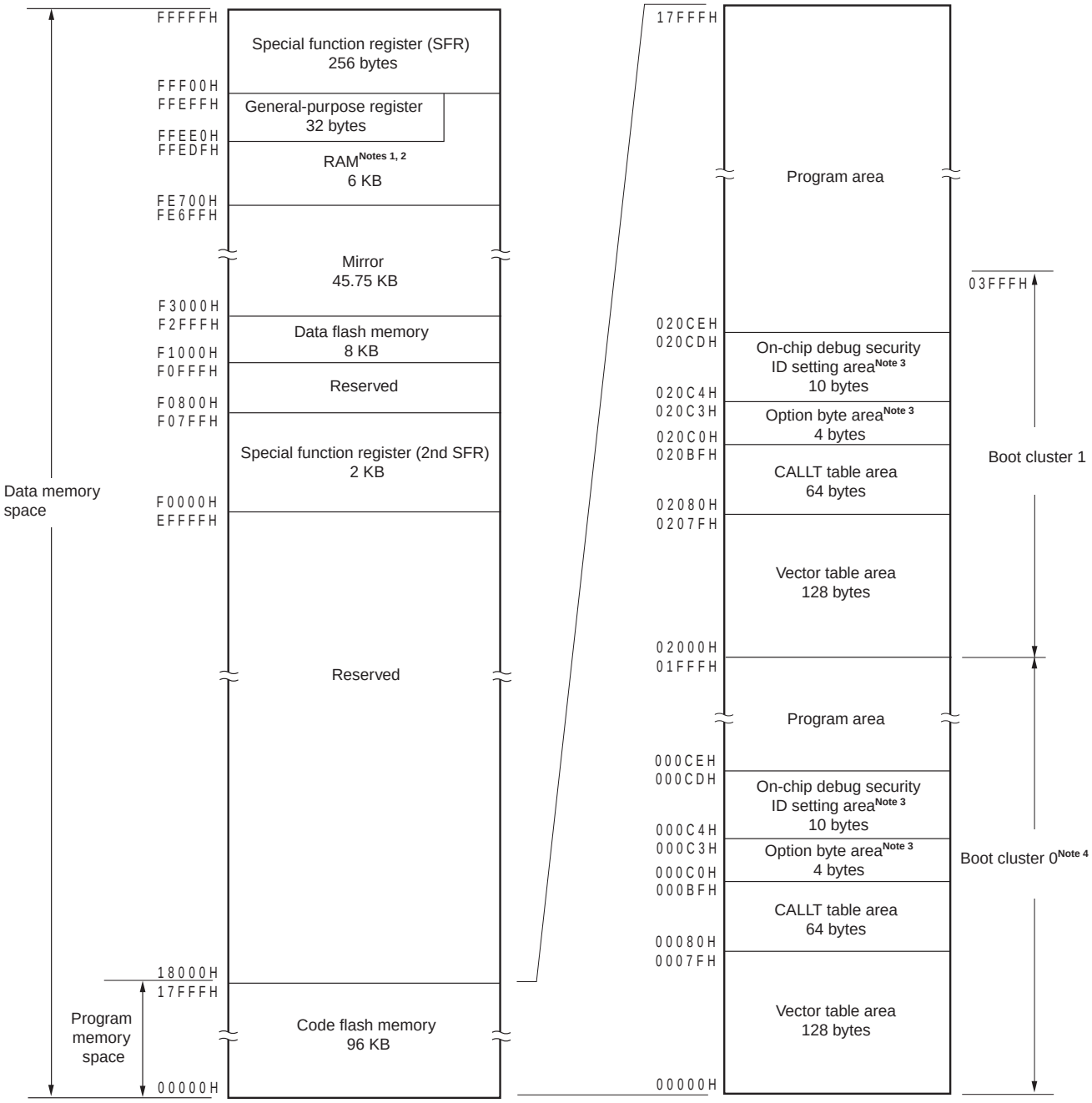
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

3. 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 and 020C4H to 020CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-5. Memory Map (R5F10DMFxFB, R5F10DPFxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

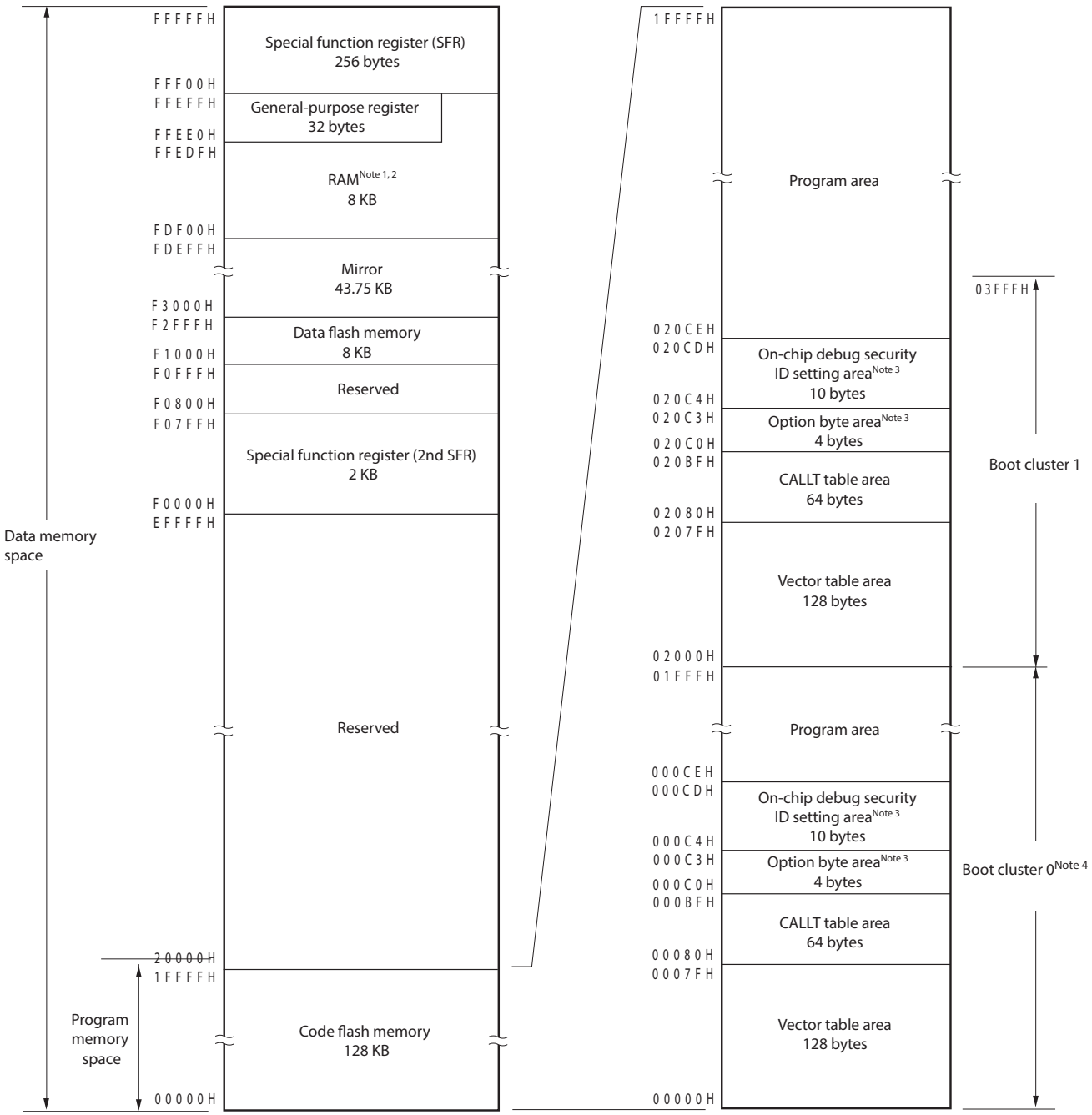
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-6. Memory Map (R5F10DMGxFB, R5F10DPGxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

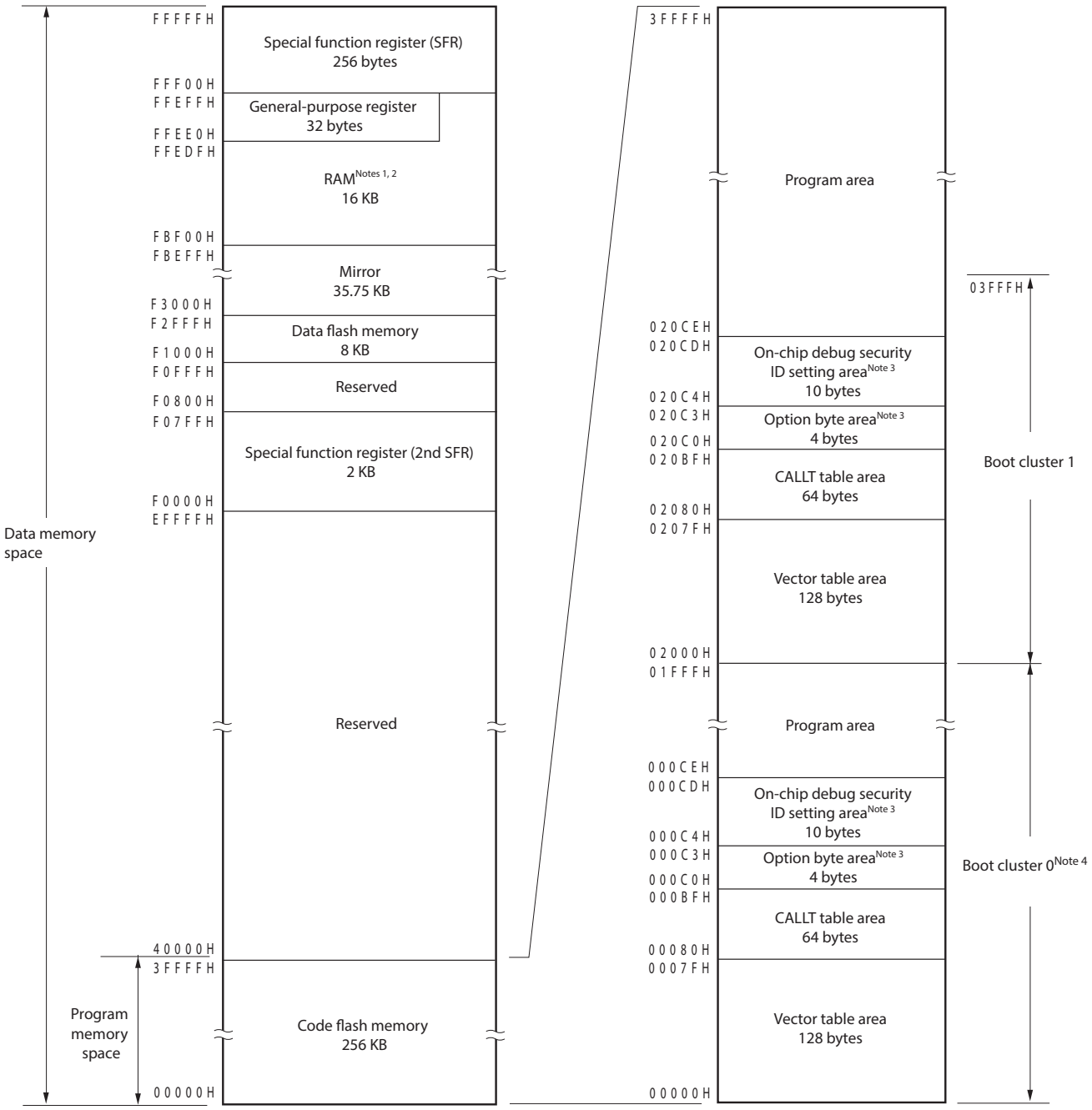
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

<R> **Figure 3-7. Memory Map (R5F10DMJxFB, R5F10TPJxFB, R5F10DPJxFB, R5F10DSJxFB)**



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

3. 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.

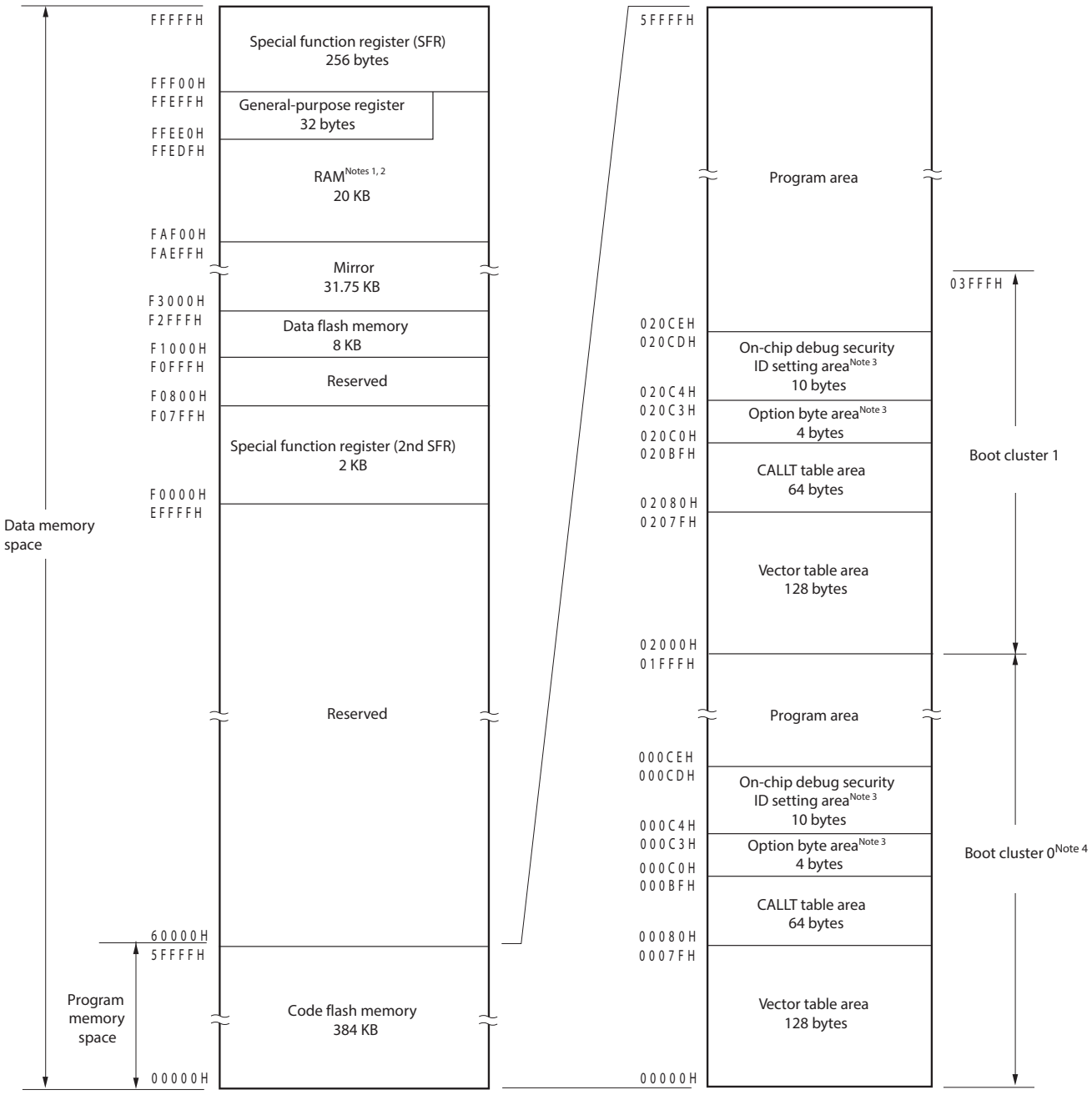
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

<R>

Figure3-8. Memory Map (R5F10DSKxFB, R5F10DPKxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

3. 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.

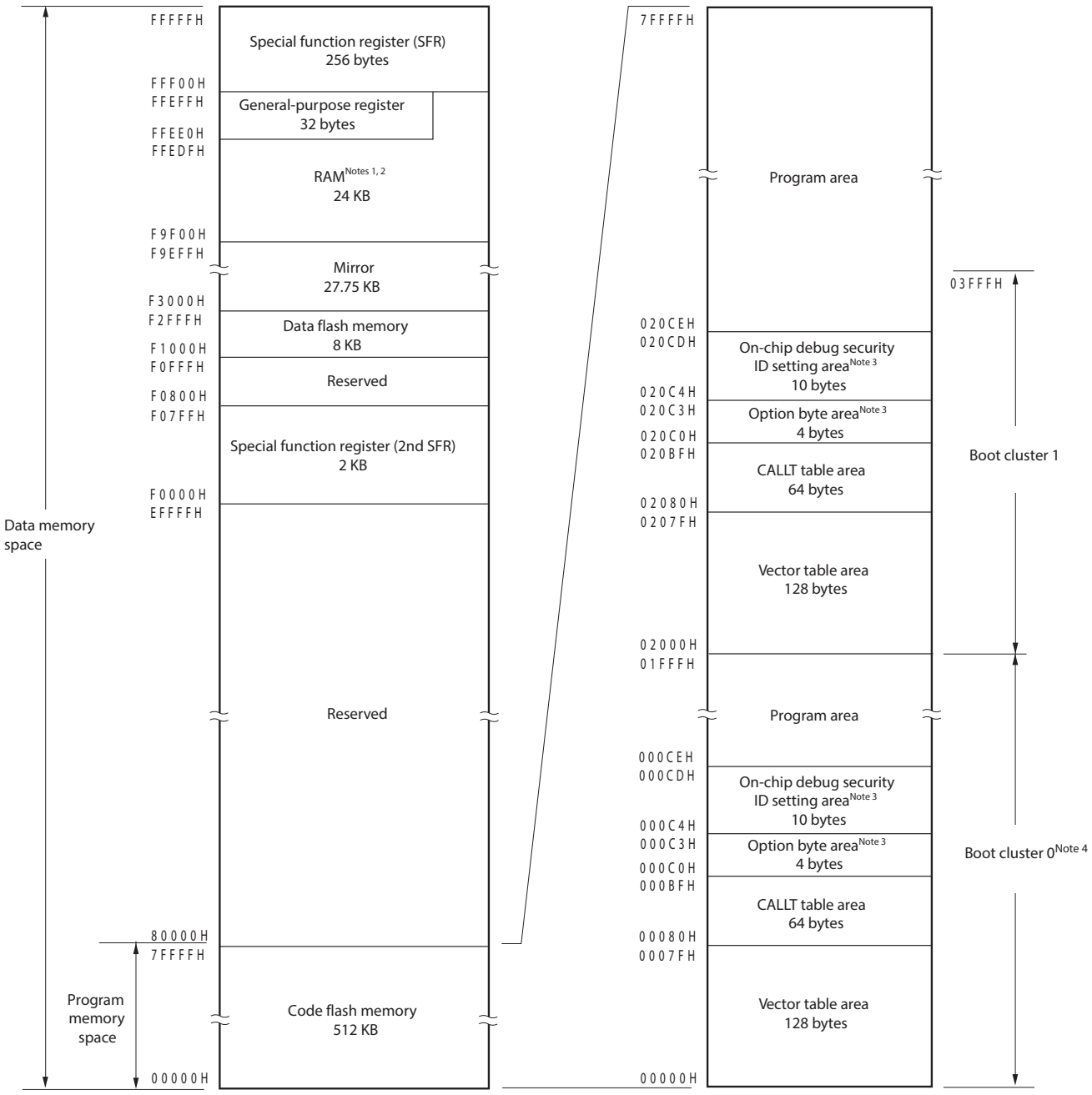
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see 29.**6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

<R>

Figure 3-9. Memory Map (R5F10DSLxFB, R5F10DPLxFB)



Notes 1. Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

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

3. 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.

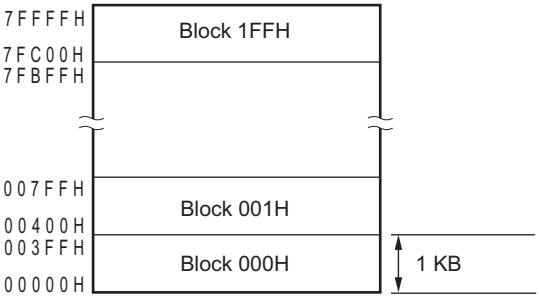
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **29.6 Security Setting**).

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

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

<R>



(In case of R5F113TLL)

(In case of that Flash size is 512KB.)

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

Table 3-1. Correspondence between Address Values and Block Numbers in Flash Memory (1/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	000H	08000H to 083FFH	020H	10000H to 103FFH	040H	18000H to 183FFH	060H
00400H to 007FFH	001H	08400H to 087FFH	021H	10400H to 107FFH	041H	18400H to 187FFH	061H
00800H to 00BFFH	002H	08800H to 08BFFH	022H	10800H to 10BFFH	042H	18800H to 18BFFH	062H
00C00H to 00FFFH	003H	08C00H to 08FFFH	023H	10C00H to 10FFFH	043H	18C00H to 18FFFH	063H
01000H to 013FFH	004H	09000H to 093FFH	024H	11000H to 113FFH	044H	19000H to 193FFH	064H
01400H to 017FFH	005H	09400H to 097FFH	025H	11400H to 117FFH	045H	19400H to 197FFH	065H
01800H to 01BFFH	006H	09800H to 09BFFH	026H	11800H to 11BFFH	046H	19800H to 19BFFH	066H
01C00H to 01FFFH	007H	09C00H to 09FFFH	027H	11C00H to 11FFFH	047H	19C00H to 19FFFH	067H
02000H to 023FFH	008H	0A000H to 0A3FFH	028H	12000H to 123FFH	048H	1A000H to 1A3FFH	068H
02400H to 027FFH	009H	0A400H to 0A7FFH	029H	12400H to 127FFH	049H	1A400H to 1A7FFH	069H
02800H to 02BFFH	00AH	0A800H to 0ABFFH	02AH	12800H to 12BFFH	04AH	1A800H to 1ABFFH	06AH
02C00H to 02FFFH	00BH	0AC00H to 0AFFFH	02BH	12C00H to 12FFFH	04BH	1AC00H to 1AFFFH	06BH
03000H to 033FFH	00CH	0B000H to 0B3FFH	02CH	13000H to 133FFH	04CH	1B000H to 1B3FFH	06CH
03400H to 037FFH	00DH	0B400H to 0B7FFH	02DH	13400H to 137FFH	04DH	1B400H to 1B7FFH	06DH
03800H to 03BFFH	00EH	0B800H to 0BBFFH	02EH	13800H to 13BFFH	04EH	1B800H to 1BBFFH	06EH
03C00H to 03FFFH	00FH	0BC00H to 0BFFFH	02FH	13C00H to 13FFFH	04FH	1BC00H to 1BFFFH	06FH
04000H to 043FFH	010H	0C000H to 0C3FFH	030H	14000H to 143FFH	050H	1C000H to 1C3FFH	070H
04400H to 047FFH	011H	0C400H to 0C7FFH	031H	14400H to 147FFH	051H	1C400H to 1C7FFH	071H
04800H to 04BFFH	012H	0C800H to 0CBFFH	032H	14800H to 14BFFH	052H	1C800H to 1CBFFH	072H
04C00H to 04FFFH	013H	0CC00H to 0CFFFH	033H	14C00H to 14FFFH	053H	1CC00H to 1CFFFH	073H
05000H to 053FFH	014H	0D000H to 0D3FFH	034H	15000H to 153FFH	054H	1D000H to 1D3FFH	074H
05400H to 057FFH	015H	0D400H to 0D7FFH	035H	15400H to 157FFH	055H	1D400H to 1D7FFH	075H
05800H to 05BFFH	016H	0D800H to 0DBFFH	036H	15800H to 15BFFH	056H	1D800H to 1DBFFH	076H
05C00H to 05FFFH	017H	0DC00H to 0DFFFH	037H	15C00H to 15FFFH	057H	1DC00H to 1DFFFH	077H
06000H to 063FFH	018H	0E000H to 0E3FFH	038H	16000H to 163FFH	058H	1E000H to 1E3FFH	078H
06400H to 067FFH	019H	0E400H to 0E7FFH	039H	16400H to 167FFH	059H	1E400H to 1E7FFH	079H
06800H to 06BFFH	01AH	0E800H to 0EBFFH	03AH	16800H to 16BFFH	05AH	1E800H to 1EBFFH	07AH
06C00H to 06FFFH	01BH	0EC00H to 0EFFFH	03BH	16C00H to 16FFFH	05BH	1EC00H to 1EFFFH	07BH
07000H to 073FFH	01CH	0F000H to 0F3FFH	03CH	17000H to 173FFH	05CH	1F000H to 1F3FFH	07CH
07400H to 077FFH	01DH	0F400H to 0F7FFH	03DH	17400H to 177FFH	05DH	1F400H to 1F7FFH	07DH
07800H to 07BFFH	01EH	0F800H to 0FBFFH	03EH	17800H to 17BFFH	05EH	1F800H to 1FBFFH	07EH
07C00H to 07FFFH	01FH	0FC00H to 0FFFFH	03FH	17C00H to 17FFFH	05FH	1FC00H to 1FFFFH	07FH

Table 3-1. Correspondence between Address Values and Block Numbers in Flash Memory (2/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
20000H to 203FFH	080H	28000H to 283FFH	0A0H	30000H to 303FFH	0C0H	38000H to 383FFH	0E0H
20400H to 207FFH	081H	28400H to 287FFH	0A1H	30400H to 307FFH	0C1H	38400H to 387FFH	0E1H
20800H to 20BFFH	082H	28800H to 28BFFH	0A2H	30800H to 30BFFH	0C2H	38800H to 38BFFH	0E2H
20C00H to 20FFFH	083H	28C00H to 28FFFH	0A3H	30C00H to 30FFFH	0C3H	38C00H to 38FFFH	0E3H
21000H to 213FFH	084H	29000H to 293FFH	0A4H	31000H to 313FFH	0C4H	39000H to 393FFH	0E4H
21400H to 217FFH	085H	29400H to 297FFH	0A5H	31400H to 317FFH	0C5H	39400H to 397FFH	0E5H
21800H to 21BFFH	086H	29800H to 29BFFH	0A6H	31800H to 31BFFH	0C6H	39800H to 39BFFH	0E6H
21C00H to 21FFFH	087H	29C00H to 29FFFH	0A7H	31C00H to 31FFFH	0C7H	39C00H to 39FFFH	0E7H
22000H to 223FFH	088H	2A000H to 2A3FFH	0A8H	32000H to 323FFH	0C8H	3A000H to 3A3FFH	0E8H
22400H to 227FFH	089H	2A400H to 2A7FFH	0A9H	32400H to 327FFH	0C9H	3A400H to 3A7FFH	0E9H
22800H to 22BFFH	08AH	2A800H to 2ABFFH	0AAH	32800H to 32BFFH	0CAH	3A800H to 3ABFFH	0EAH
22C00H to 22FFFH	08BH	2AC00H to 2AFFFH	0ABH	32C00H to 32FFFH	0CBH	3AC00H to 3AFFFH	0EBH
23000H to 233FFH	08CH	2B000H to 2B3FFH	0ACH	33000H to 333FFH	0CCH	3B000H to 3B3FFH	0ECH
23400H to 237FFH	08DH	2B400H to 2B7FFH	0ADH	33400H to 337FFH	0CDH	3B400H to 3B7FFH	0EDH
23800H to 23BFFH	08EH	2B800H to 2BBFFH	0AEH	33800H to 33BFFH	0CEH	3B800H to 3BBFFH	0EEH
23C00H to 23FFFH	08FH	2BC00H to 2BFFFH	0AFH	33C00H to 33FFFH	0CFH	3BC00H to 3BFFFH	0EFH
24000H to 243FFH	090H	2C000H to 2C3FFH	0B0H	34000H to 343FFH	0D0H	3C000H to 3C3FFH	0F0H
24400H to 247FFH	091H	2C400H to 2C7FFH	0B1H	34400H to 347FFH	0D1H	3C400H to 3C7FFH	0F1H
24800H to 24BFFH	092H	2C800H to 2CBFFH	0B2H	34800H to 34BFFH	0D2H	3C800H to 3CBFFH	0F2H
24C00H to 24FFFH	093H	2CC00H to 2CFFFH	0B3H	34C00H to 34FFFH	0D3H	3CC00H to 3CFFFH	0F3H
25000H to 253FFH	094H	2D000H to 2D3FFH	0B4H	35000H to 353FFH	0D4H	3D000H to 3D3FFH	0F4H
25400H to 257FFH	095H	2D400H to 2D7FFH	0B5H	35400H to 357FFH	0D5H	3D400H to 3D7FFH	0F5H
25800H to 25BFFH	096H	2D800H to 2DBFFH	0B6H	35800H to 35BFFH	0D6H	3D800H to 3DBFFH	0F6H
25C00H to 25FFFH	097H	2DC00H to 2DFFFH	0B7H	35C00H to 35FFFH	0D7H	3DC00H to 3DFFFH	0F7H
26000H to 263FFH	098H	2E000H to 2E3FFH	0B8H	36000H to 363FFH	0D8H	3E000H to 3E3FFH	0F8H
26400H to 267FFH	099H	2E400H to 2E7FFH	0B9H	36400H to 367FFH	0D9H	3E400H to 3E7FFH	0F9H
26800H to 26BFFH	09AH	2E800H to 2EBFFH	0BAH	36800H to 36BFFH	0DAH	3E800H to 3EBFFH	0FAH
26C00H to 26FFFH	09BH	2EC00H to 2EFFFH	0BBH	36C00H to 36FFFH	0DBH	3EC00H to 3EFFFH	0FBH
27000H to 273FFH	09CH	2F000H to 2F3FFH	0BCH	37000H to 373FFH	0DCH	3F000H to 3F3FFH	0FCH
27400H to 277FFH	09DH	2F400H to 2F7FFH	0BDH	37400H to 377FFH	0DDH	3F400H to 3F7FFH	0FDH
27800H to 27BFFH	09EH	2F800H to 2FBFFH	0BEH	37800H to 37BFFH	0DEH	3F800H to 3FBFFH	0FEH
27C00H to 27FFFH	09FH	2FC00H to 2FFFFH	0BFH	37C00H to 37FFFH	0DFH	3FC00H to 3FFFFH	0FFH

<R> **Table 3-1. Correspondence between Address Values and Block Numbers in Flash Memory (3/4)**

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
40000H-403FFH	100H	48000H-483FFH	120H	50000H-503FFH	140H	58000H-583FFH	160H
40400H-407FFH	101H	48400H-487FFH	121H	50400H-507FFH	141H	58400H-587FFH	161H
40800H-40BFFH	102H	48800H-48BFFH	122H	50800H-50BFFH	142H	58800H-58BFFH	162H
40C00H-40FFFH	103H	48C00H-48FFFH	123H	50C00H-50FFFH	143H	58C00H-58FFFH	163H
41000H-413FFH	104H	49000H-493FFH	124H	51000H-513FFH	144H	59000H-593FFH	164H
41400H-417FFH	105H	49400H-497FFH	125H	51400H-517FFH	145H	59400H-597FFH	165H
41800H-41BFFH	106H	49800H-49BFFH	126H	51800H-51BFFH	146H	59800H-59BFFH	166H
41C00H-41FFFH	107H	49C00H-49FFFH	127H	51C00H-51FFFH	147H	59C00H-59FFFH	167H
42000H-423FFH	108H	4A000H-4A3FFH	128H	52000H-523FFH	148H	5A000H-5A3FFH	168H
42400H-427FFH	109H	4A400H-4A7FFH	129H	52400H-527FFH	149H	5A400H-5A7FFH	169H
42800H-42BFFH	10AH	4A800H-4ABFFH	12AH	52800H-52BFFH	14AH	5A800H-5ABFFH	16AH
42C00H-42FFFH	10BH	4AC00H-4AFFFH	12BH	52C00H-52FFFH	14BH	5AC00H-5AFFFH	16BH
43000H-433FFH	10CH	4B000H-4B3FFH	12CH	53000H-533FFH	14CH	5B000H-5B3FFH	16CH
43400H-437FFH	10DH	4B400H-4B7FFH	12DH	53400H-537FFH	14DH	5B400H-5B7FFH	16DH
43800H-43BFFH	10EH	4B800H-4BBFFH	12EH	53800H-53BFFH	14EH	5B800H-5BBFFH	16EH
43C00H-43FFFH	10FH	4BC00H-4BFFFH	12FH	53C00H-53FFFH	14FH	5BC00H-5BFFFH	16FH
44000H-443FFH	110H	4C000H-4C3FFH	130H	54000H-543FFH	150H	5C000H-5C3FFH	170H
44400H-447FFH	111H	4C400H-4C7FFH	131H	54400H-547FFH	151H	5C400H-5C7FFH	171H
44800H-44BFFH	112H	4C800H-4CBFFH	132H	54800H-54BFFH	152H	5C800H-5CBFFH	172H
44C00H-44FFFH	113H	4CC00H-4CFFFH	133H	54C00H-54FFFH	153H	5CC00H-5CFFFH	173H
45000H-453FFH	114H	4D000H-4D3FFH	134H	55000H-553FFH	154H	5D000H-5D3FFH	174H
45400H-457FFH	115H	4D400H-4D7FFH	135H	55400H-557FFH	155H	5D400H-5D7FFH	175H
45800H-45BFFH	116H	4D800H-4DBFFH	136H	55800H-55BFFH	156H	5D800H-5DBFFH	176H
45C00H-45FFFH	117H	4DC00H-4DFFFH	137H	55C00H-55FFFH	157H	5DC00H-5DFFFH	177H
46000H-463FFH	118H	4E000H-4E3FFH	138H	56000H-563FFH	158H	5E000H-5E3FFH	178H
46400H-467FFH	119H	4E400H-4E7FFH	139H	56400H-567FFH	159H	5E400H-5E7FFH	179H
46800H-46BFFH	11AH	4E800H-4EBFFH	13AH	56800H-56BFFH	15AH	5E800H-5EBFFH	17AH
46C00H-46FFFH	11BH	4EC00H-4EFFFH	13BH	56C00H-56FFFH	15BH	5EC00H-5EFFFH	17BH
47000H-473FFH	11CH	4F000H-4F3FFH	13CH	57000H-573FFH	15CH	5F000H-5F3FFH	17CH
47400H-477FFH	11DH	4F400H-4F7FFH	13DH	57400H-577FFH	15DH	5F400H-5F7FFH	17DH
47800H-47BFFH	11EH	4F800H-4FBFFH	13EH	57800H-57BFFH	15EH	5F800H-5FBFFH	17EH
47C00H-47FFFH	11FH	4FC00H-4FFFFH	13FH	57C00H-57FFFH	15FH	5FC00H-5FFFFH	17FH

<R> **Table 3-1. Correspondence between Address Values and Block Numbers in Flash Memory (4/4)**

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
60000H-603FFH	180H	68000H-683FFH	1A0H	70000H-703FFH	1C0H	78000H-783FFH	1E0H
60400H-607FFH	181H	68400H-687FFH	1A1H	70400H-707FFH	1C1H	78400H-787FFH	1E1H
60800H-60BFFH	182H	68800H-68BFFH	1A2H	70800H-70BFFH	1C2H	78800H-78BFFH	1E2H
60C00H-60FFFH	183H	68C00H-68FFFH	1A3H	70C00H-70FFFH	1C3H	78C00H-78FFFH	1E3H
61000H-613FFH	184H	69000H-693FFH	1A4H	71000H-713FFH	1C4H	79000H-793FFH	1E4H
61400H-617FFH	185H	69400H-697FFH	1A5H	71400H-717FFH	1C5H	79400H-797FFH	1E5H
61800H-61BFFH	186H	69800H-69BFFH	1A6H	71800H-71BFFH	1C6H	79800H-79BFFH	1E6H
61C00H-61FFFH	187H	69C00H-69FFFH	1A7H	71C00H-71FFFH	1C7H	79C00H-79FFFH	1E7H
62000H-623FFH	188H	6A000H-6A3FFH	1A8H	72000H-723FFH	1C8H	7A000H-7A3FFH	1E8H
62400H-627FFH	189H	6A400H-6A7FFH	1A9H	72400H-727FFH	1C9H	7A400H-7A7FFH	1E9H
62800H-62BFFH	18AH	6A800H-6ABFFH	1AAH	72800H-72BFFH	1CAH	7A800H-7ABFFH	1EAH
62C00H-62FFFH	18BH	6AC00H-6AFFFH	1ABH	72C00H-72FFFH	1CBH	7AC00H-7AFFFH	1EBH
63000H-633FFH	18CH	6B000H-6B3FFH	1ACH	73000H-733FFH	1CCH	7B000H-7B3FFH	1ECH
63400H-637FFH	18DH	6B400H-6B7FFH	1ADH	73400H-737FFH	1CDH	7B400H-7B7FFH	1EDH
63800H-63BFFH	18EH	6B800H-6BBFFH	1AEH	73800H-73BFFH	1CEH	7B800H-7BBFFH	1EEH
63C00H-63FFFH	18FH	6BC00H-6BFFFH	1AFH	73C00H-73FFFH	1CFH	7BC00H-7BFFFH	1EFH
64000H-643FFH	190H	6C000H-6C3FFH	1B0H	74000H-743FFH	1D0H	7C000H-7C3FFH	1F0H
64400H-647FFH	191H	6C400H-6C7FFH	1B1H	74400H-747FFH	1D1H	7C400H-7C7FFH	1F1H
64800H-64BFFH	192H	6C800H-6CBFFH	1B2H	74800H-74BFFH	1D2H	7C800H-7CBFFH	1F2H
64C00H-64FFFH	193H	6CC00H-6CFFFH	1B3H	74C00H-74FFFH	1D3H	7CC00H-7CFFFH	1F3H
65000H-653FFH	194H	6D000H-6D3FFH	1B4H	75000H-753FFH	1D4H	7D000H-7D3FFH	1F4H
65400H-657FFH	195H	6D400H-6D7FFH	1B5H	75400H-757FFH	1D5H	7D400H-7D7FFH	1F5H
65800H-65BFFH	196H	6D800H-6DBFFH	1B6H	75800H-75BFFH	1D6H	7D800H-7DBFFH	1F6H
65C00H-65FFFH	197H	6DC00H-6DFFFH	1B7H	75C00H-75FFFH	1D7H	7DC00H-7DFFFH	1F7H
66000H-663FFH	198H	6E000H-6E3FFH	1B8H	76000H-763FFH	1D8H	7E000H-7E3FFH	1F8H
66400H-667FFH	199H	6E400H-6E7FFH	1B9H	76400H-767FFH	1D9H	7E400H-7E7FFH	1F9H
66800H-66BFFH	19AH	6E800H-6EBFFH	1BAH	76800H-76BFFH	1DAH	7E800H-7EBFFH	1FAH
66C00H-66FFFH	19BH	6EC00H-6EFFFH	1BBH	76C00H-76FFFH	1DBH	7EC00H-7EFFFH	1FBH
67000H-673FFH	19CH	6F000H-6F3FFH	1BCH	77000H-773FFH	1DCH	7F000H-7F3FFH	1FCH
67400H-677FFH	19DH	6F400H-6F7FFH	1BDH	77400H-777FFH	1DDH	7F400H-7F7FFH	1FDH
67800H-67BFFH	19EH	6F800H-6FBFFH	1BEH	77800H-77BFFH	1DEH	7F800H-7FBFFH	1FEH
67C00H-67FFFH	19FH	6FC00H-6FFFFH	1BFH	77C00H-77FFFH	1DFH	7FC00H-7FFFFH	1FFH

3.1.1 Internal program memory space

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

The RL78/D1A products incorporate internal ROM (flash memory), as shown below.

<R>

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F10CGBxFB	Flash memory	24576 × 8 bits (00000H to 05FFFH)
R5F10CGCxFB, R5F10DGCxFB		32768 × 8 bits (00000H to 07FFFH)
R5F10CGDxFB, R5F10DGDxFB, R5F10CLDxFB, R5F10DLDxFB, R5F10CMDxFB, R5F10DMDxFB		49152 × 8 bits (00000H to 0BFFFH)
R5F10DGExFB, R5F10DLExFB, R5F10CMExFB, R5F10DMExFB, R5F10DPExFB		65536 × 8 bits (00000H to 0FFFFH)
R5F10DMFxFB, R5F10DPFxFB		98304 × 8 bits (00000H to 17FFFH)
R5F10DMGxFB, R5F10DPGxFB		131072 × 8 bits (00000H to 1FFFFH)
R5F10DMJxFB, R5F10TPJxFB, R5F10DPJxFB, R5F10DSJxFB		262144 × 8 bits (00000H to 3FFFFH)
R5F10DSKxFB, R5F10DPKxFB		393216 × 8 bits (00000H to 5FFFFH)
R5F10DSLxFB, R5F10DPLxFB		524288 × 8 bits (00000H to 7FFFFH)

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.

To use the boot swap function, set a vector table also at 02000H to 0207FH.

<R>

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

Vector Table Address	Interrupt Source	48-pin		64-pin		80-pin		100-pin		128-pin
		R5F10CGx	R5F10DGx	R5F10CLx	R5F10DLx	R5F10CMx	R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DPJ	R5F10DSx
0000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE, CLKM	√	√	√	√	√	√	√	√	√
0004H	INTWDTI	√	√	√	√	√	√	√	√	√
0006H	INTLVI	√	√	√	√	√	√	√	√	√
0008H	INTP0	–	–	√	√	√	√	√	√	√
000AH	INTP1	√	√	√	√	√	√	√	√	√
000CH	INTP2	√	√	√	√	√	√	√	√	√
000EH	INTP3	√	√	√	√	√	√	√	√	√
0010H	INTP4	√	√	√	√	√	√	√	√	√
0012H	INTP5	√	√	√	√	√	√	√	√	√
0014H	INTCLM	√	√	√	√	√	√	√	√	√
0016H	INTCSI00	√	√	√	√	√	√	√	√	√
	INTST0	–	–	–	–	–	–	–	–	√
0018H	INTCSI01	√	√	√	√	√	√	√	√	√
	INTSR0	–	–	–	–	–	–	–	–	√
001AH	INTDMA0	√	√	√	√	√	√	√	√	√
001CH	INTDMA1	√	√	√	√	√	√	√	√	√
001EH	INTRTC	√	√	√	√	√	√	√	√	√
0020H	INTIT	√	√	√	√	√	√	√	√	√
0022H	INTLT0	–	–	√	√	√	√	√	√	√
0024H	INTLR0	–	–	√	√	√	√	√	√	√
0026H	INTLS0	–	–	√	√	√	√	√	√	√
0028H	INTPLR0	–	–	√	√	√	√	√	√	√
002AH	INTSG	√	√	√	√	√	√	√	√	√
002CH	INTTM00	√	√	√	√	√	√	√	√	√
002EH	INTTM01	√	√	√	√	√	√	√	√	√
0030H	INTTM02	√	√	√	√	√	√	√	√	√
0032H	INTTM03	√	√	√	√	√	√	√	√	√
0034H	INTAD	√	√	√	√	√	√	√	√	√
0036H	INTLT1	√	√	√	√	√	√	√	√	√
0038H	INTLR1	√	√	√	√	√	√	√	√	√
003AH	INTLS1	√	√	√	√	√	√	√	√	√
003CH	INTPLR1	√	√	√	√	√	√	√	√	√
003EH	INTCSI10	–	–	–	–	–	–	√	√	√
0040H	INTIIC11	√	√	√	√	√	√	√	√	√

<R>

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

Vector Table Address	Interrupt Source	48-pin		64-pin		80-pin		100-pin		128-pin
		R5F10CGx	R5F10DGx	R5F10CLx	R5F10DLx	R5F10DSx	R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DPJ	R5F10DSx
0042H	INTTM04	√	√	√	√	√	√	√	√	√
0044H	INTTM05	√	√	√	√	√	√	√	√	√
0046H	INTTM06	√	√	√	√	√	√	√	√	√
0048H	INTTM07	√	√	√	√	√	√	√	√	√
004AH	INTC1ERR	–	–	–	–	–	–	–	√	√
004CH	INTC1WUP	–	–	–	–	–	–	–	√	√
004EH	INTC0ERR	–	√	–	√	–	√	√	√	√
0050H	INTC0WUP	–	√	–	√	–	√	√	√	√
0052H	INTC0REC	–	√	–	√	–	√	√	√	√
0054H	INTC0TRX	–	√	–	√	–	√	√	√	√
0056H	INTTM10	√	√	√	√	√	√	√	√	√
0058H	INTTM11	√	√	√	√	√	√	√	√	√
005AH	INTTM12	√	√	√	√	√	√	√	√	√
005CH	INTTM13	√	√	√	√	√	√	√	√	√
005EH	INTMD	√	√	√	√	√	√	√	√	√
0060H	INTC1REC	–	–	–	–	–	–	–	√	√
0062H	INTFL	√	√	√	√	√	√	√	√	√
0064H	INTC1TRX	–	–	–	–	–	–	–	√	√
0066H	INTTM14	√	√	√	√	√	√	√	√	√
0068H	INTTM15	√	√	√	√	√	√	√	√	√
006AH	INTTM16	√	√	√	√	√	√	√	√	√
006CH	INTTM17	√	√	√	√	√	√	√	√	√
006EH	INTTM20	√	√	√	√	√	√	√	√	√
00070H	INTTM21	√	√	√	√	√	√	√	√	√
00072H	INTTM22	√	√	√	√	√	√	√	√	√
0074H	INTTM23	√	√	√	√	√	√	√	√	√
0076H	INTTM24	√	√	√	√	√	√	√	√	√
0078H	INTTM26	√	√	√	√	√	√	√	√	√
007AH	INTDMA2	–	–	–	–	–	–	√	√	√
007CH	INTDMA3	–	–	–	–	–	–	√	√	√

(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 02080H to 020BFH.

(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 020C0H to 020C3H when the boot swap is used. For details, see **CHAPTER 28 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 020C4H to 020CDH 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 020C4H to 020CDH when the boot swap is used. For details, see **CHAPTER 30 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/D1A mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH. The products with 96 KB or more flash memory mirror the code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

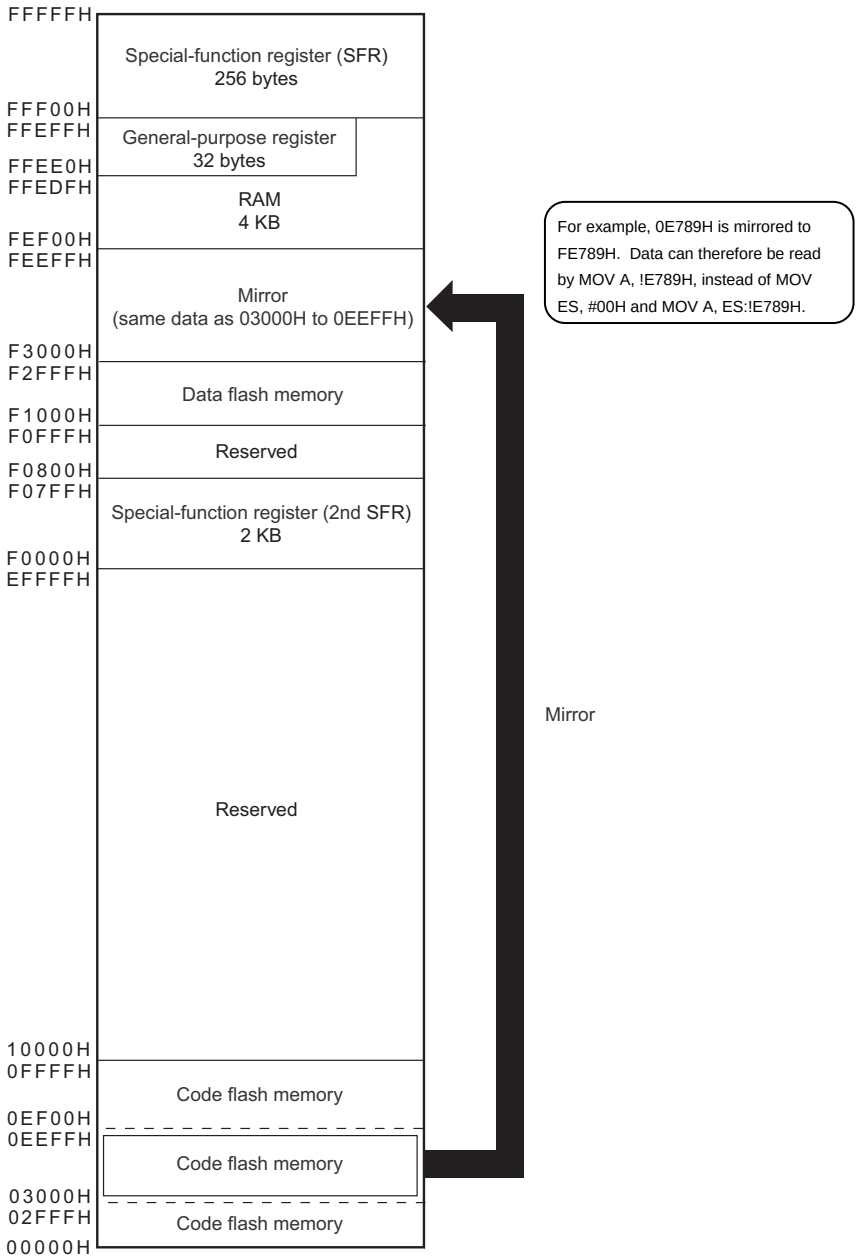
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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 R5F10DMExFB (Flash memory: 64 KB, RAM: 4 KB)



The PMC register is described below.

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

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

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

Reset signal generation sets this register to 00H.

Figure 3-10. Format of **Configuration of Processor Mode Control Register (PMC)**

Address: FFFFEH After reset: 00H R/W

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

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

Cautions 1. In products with 64 KB or less flash memory, be sure to clear bit 0 (MAA) of this register to 0 (default value).

<R> **2. After setting the PMC register, wait for at least one instruction and access the mirror area.**

3.1.3 Internal data memory space

The RL78/D1A products incorporate the following RAMs.

<R>

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R5F10CGBxFB, R5F10CGCxFB, R5F10DGCxFB	2048 × 8 bits (FF700H to FFEFFH)
R5F10CGDxFB, R5F10DGDxFB, R5F10CLDxFB, R5F10DLDxFB, R5F10CMDxFB, R5F10DMDxFB	3072 × 8 bits (FF300H to FFEFFH)
R5F10DGExFB, R5F10DLExFB, R5F10CMExFB, R5F10DMExFB, R5F10DPExFB	4096 × 8 bits (FEF00H to FFEFFH)
R5F10DMFxFB, R5F10DPFxFB	6144 × 8 bits (FE700H to FFEFFH)
R5F10DMGxFB, R5F10DPGxFB	8192 × 8 bits (FDF00H to FFEFFH)
R5F10DMJxFB, R5F10TPJxFB, R5F10DPJxFB, R5F10DSJxFB	16384 × 8 bits (FBF00H to FFEFFH)
R5F10DSKxFB, R5F10DPKxFB	20480 × 8 bits (FAF00H to FFEFFH)
R5F10DSLxFB, R5F10DPLxFB	24576 × 8 bits (F9F00H to FFEFFH)

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

The internal RAM is used as stack memory.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching or as a
<R> **stack area.**

2. Do not allocate the stack area, data buffers for use by the flash library, arguments of library functions, branch destinations in the processing of vectored interrupts, or destinations or sources for DMA transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the
<R> **data flash memory.**

3. The flash library uses RAM in self-programming and rewriting of the data flash memory.
The target products and start address of the RAM areas used by the flash library are shown below.
R5F10CGDxFB, R5F10DGDxFB, R5F10CLDxFB, R5F10DLDxFB, R5F10CMDxFB, R5F10DMDxFB: Start address FF300H
R5F10DGExFB, R5F10DLExFB, R5F10CMExFB, R5F10DMExFB, R5F10DPExFB: Start address FEF00H
R5F10DSKxFB, R5F10DPKxFB: Start address FAF00H
R5F10DSLxFB, R5F10DPLxFB: Start address F9F00H
For the RAM areas used by the flash library, see Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944).

3.1.4 Special function register (SFR) area

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

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

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

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

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

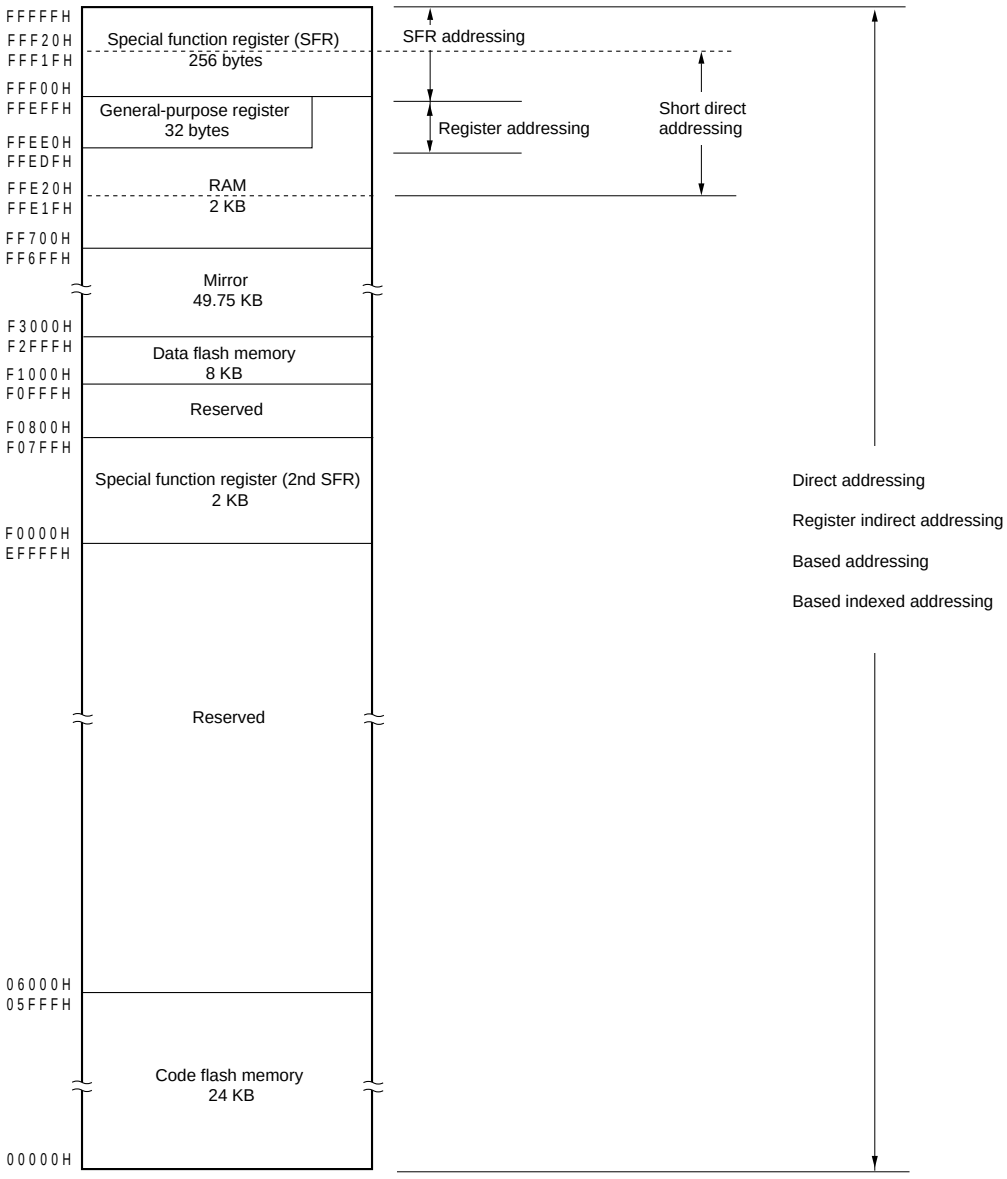
Caution Do not access addresses to which extended SFRs are not assigned.

3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/D1A, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figures 3-9 to 3-15 show correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

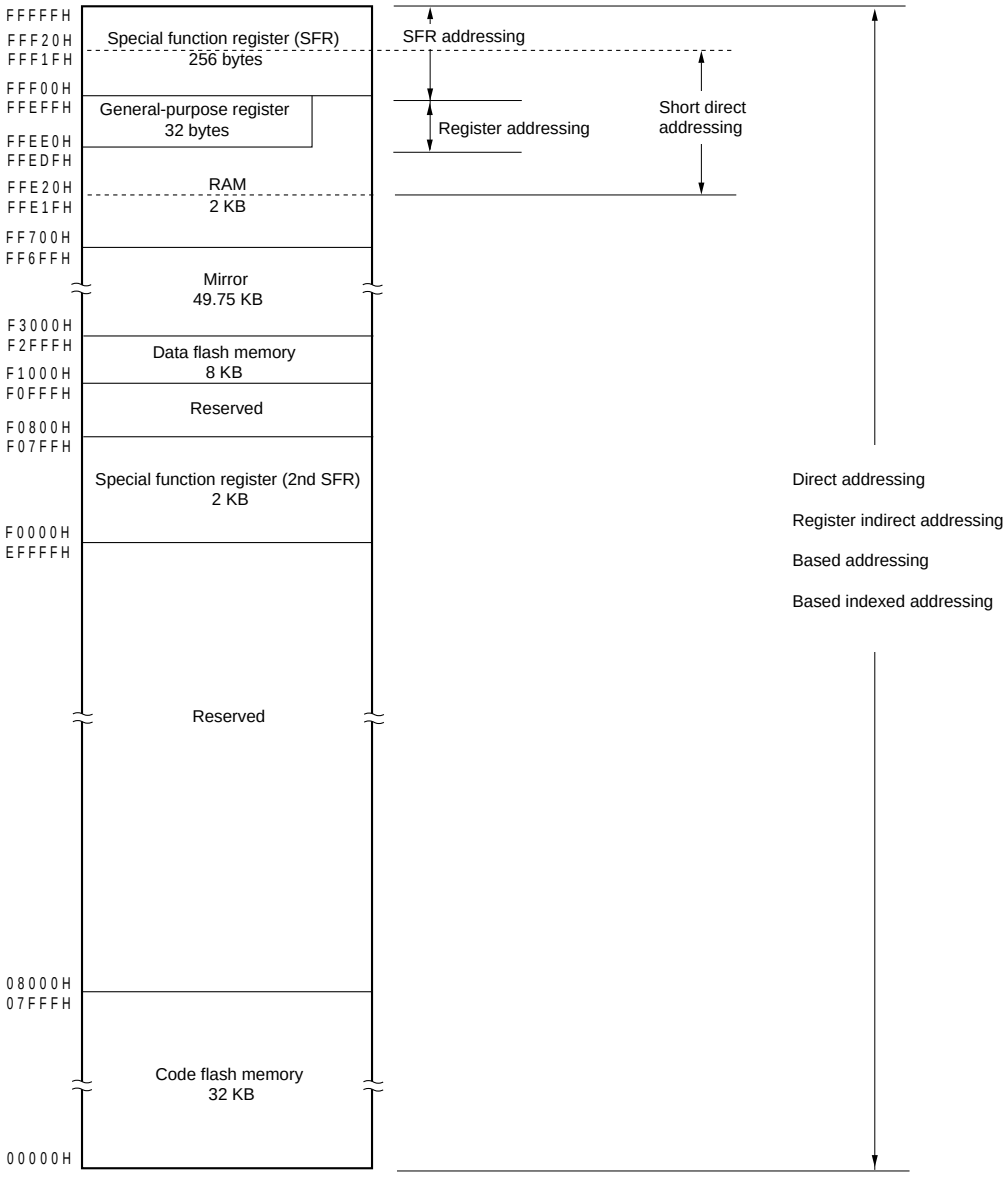
Figure 3-11. Correspondence Between Data Memory and Addressing (R5F10CGD)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

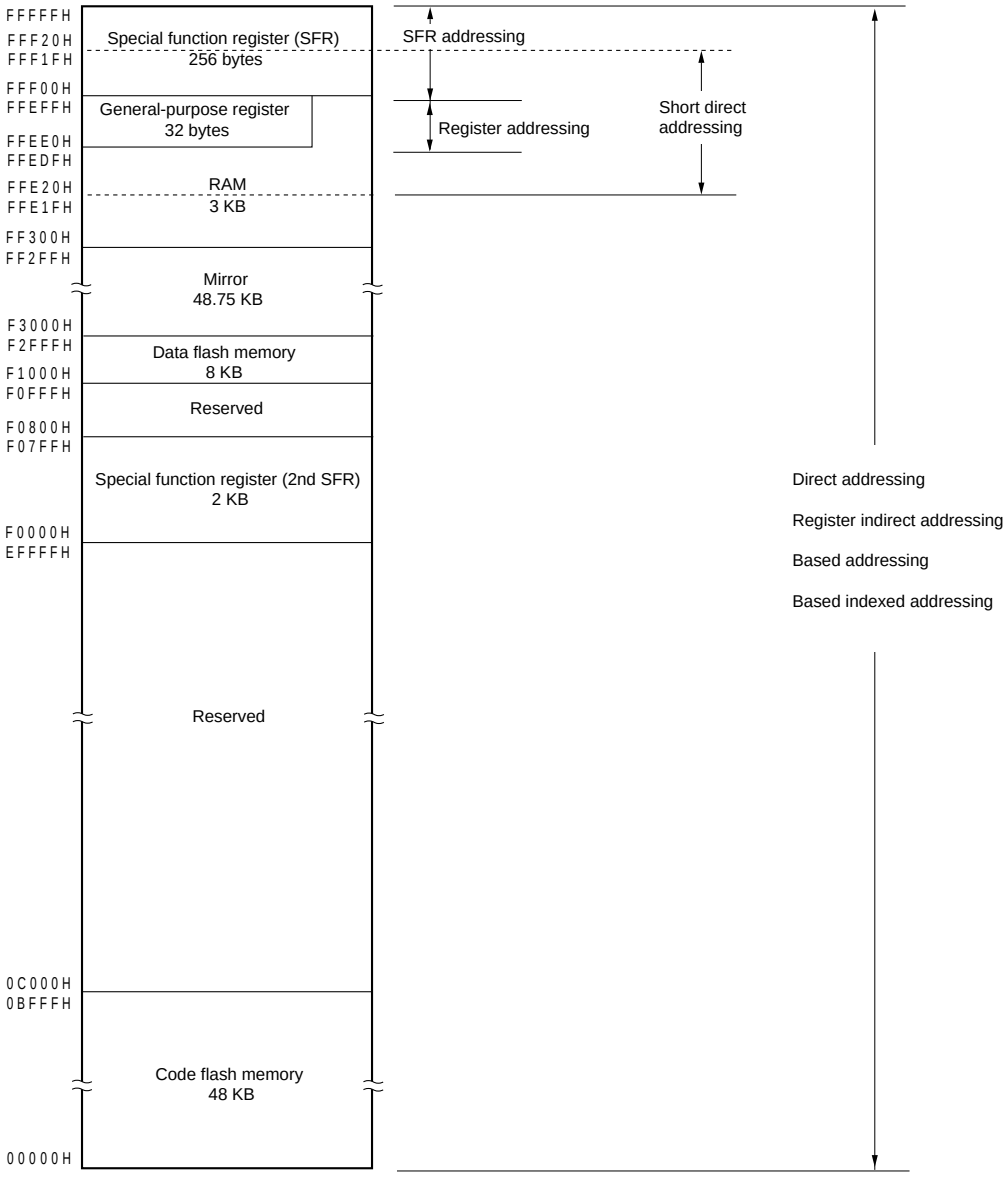
Figure 3-12. Correspondence Between Data Memory and Addressing (R5F10CGC, R5F10DGC)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

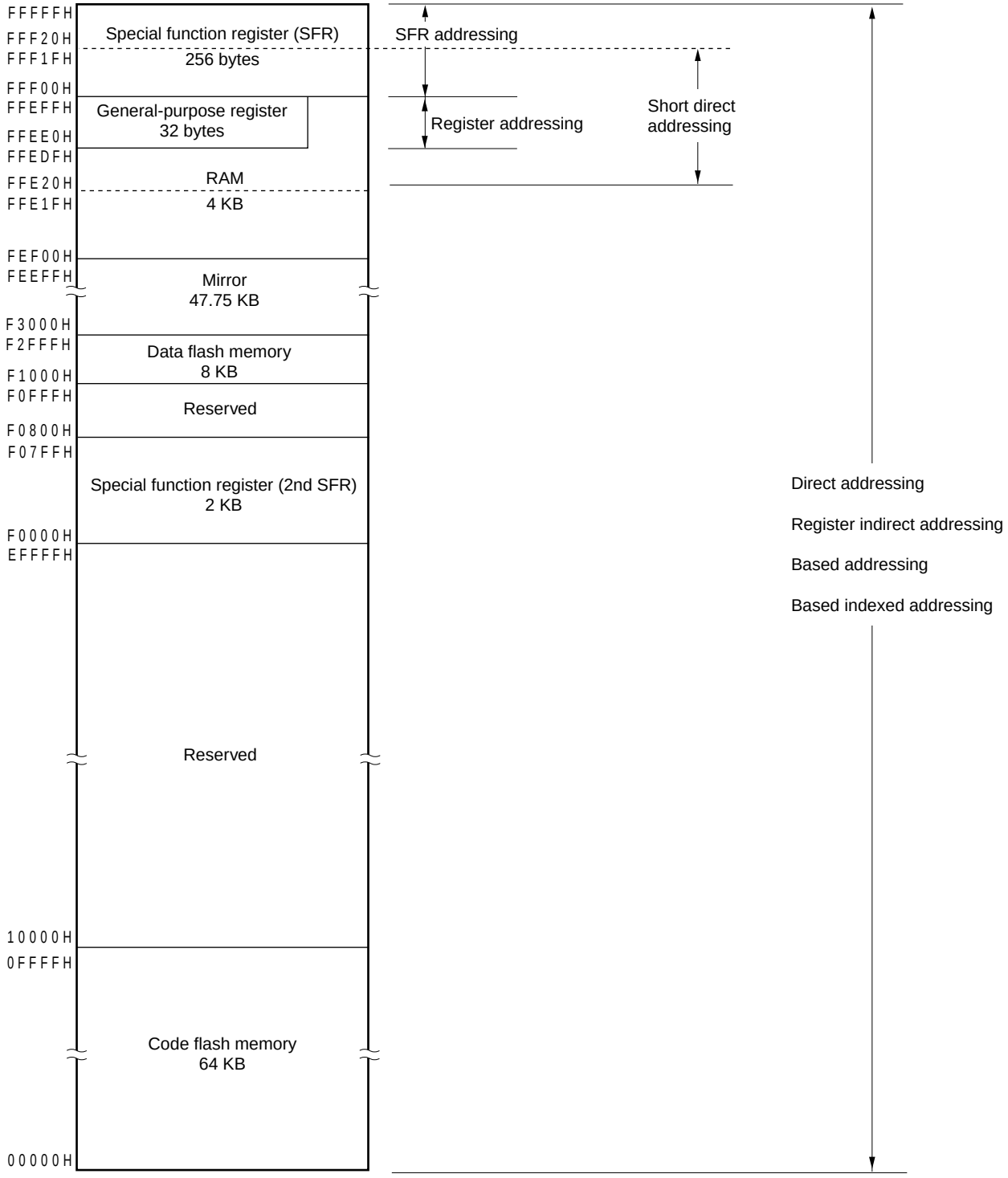
Figure 3-13. Correspondence Between Data Memory and Addressing
(R5F10CGD, R5F10DGD, R5F10CLD, R5F10DLD, R5F10CMD, R5F10DMD)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

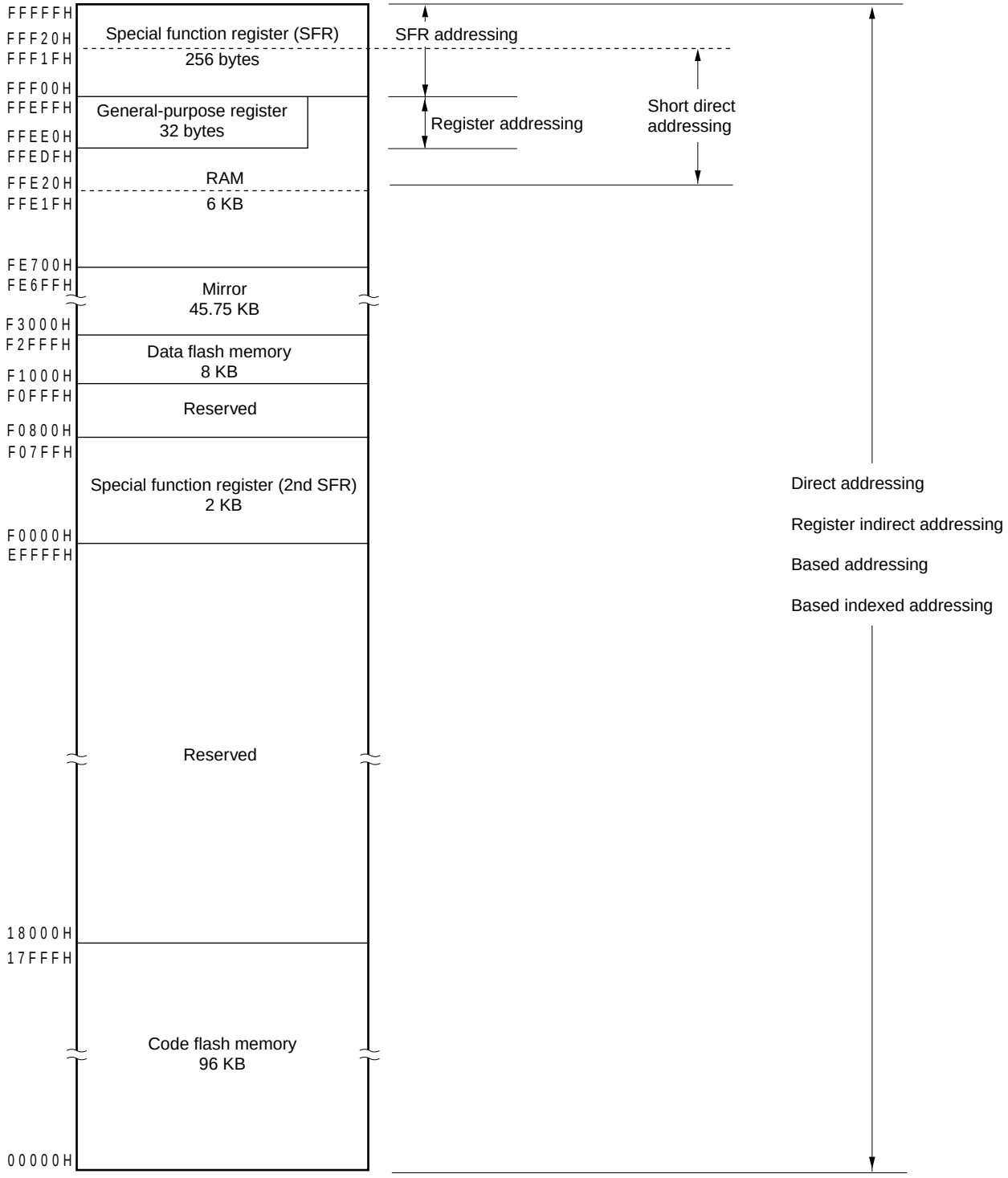
Figure 3-14. Correspondence Between Data Memory and Addressing
(R5F10DGE, R5F10DLE, R5F10CME, R5F10DME, R5F10DPE)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

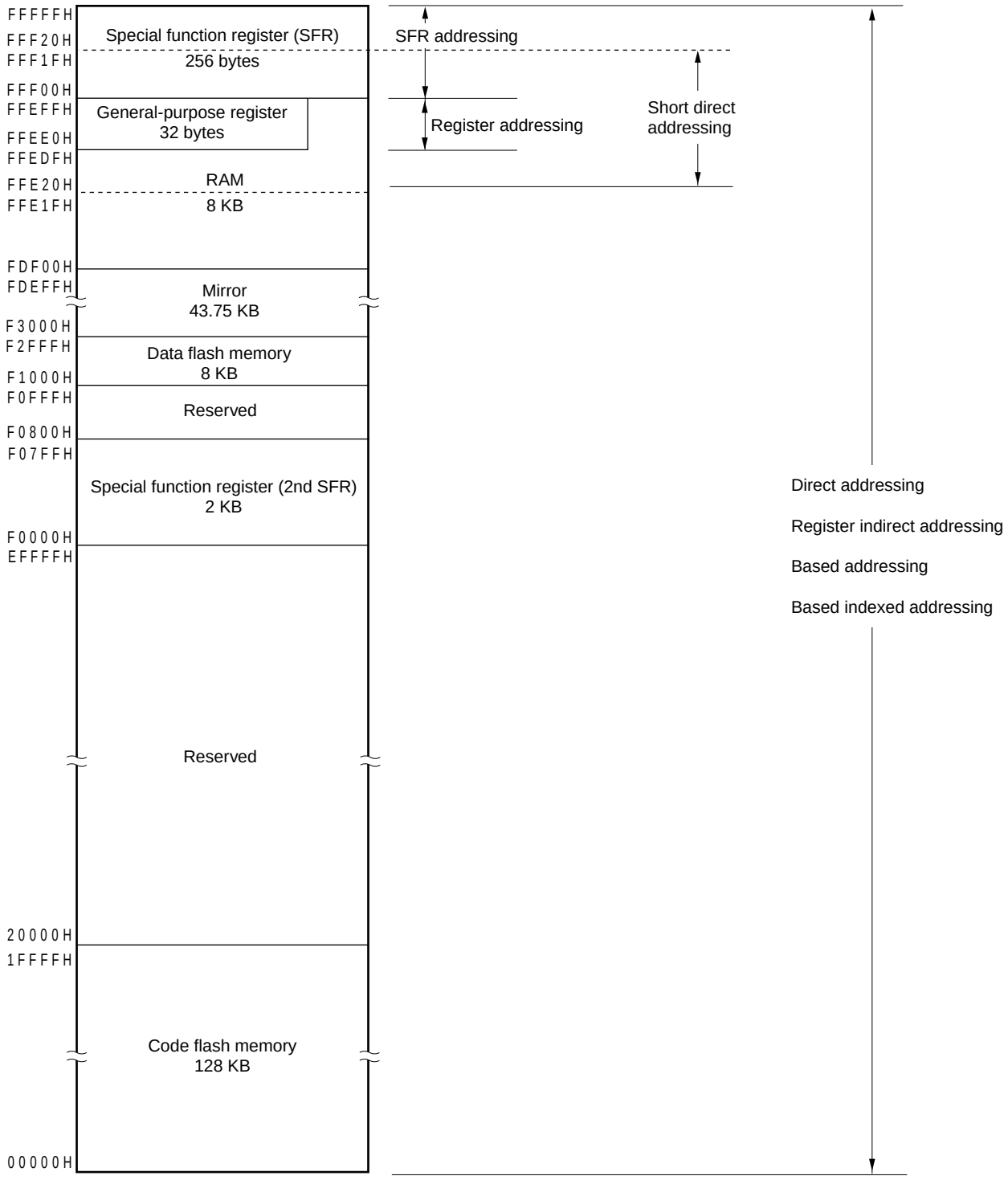
Figure 3-15. Correspondence Between Data Memory and Addressing
(R5F10DMF, R5F10DPF)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

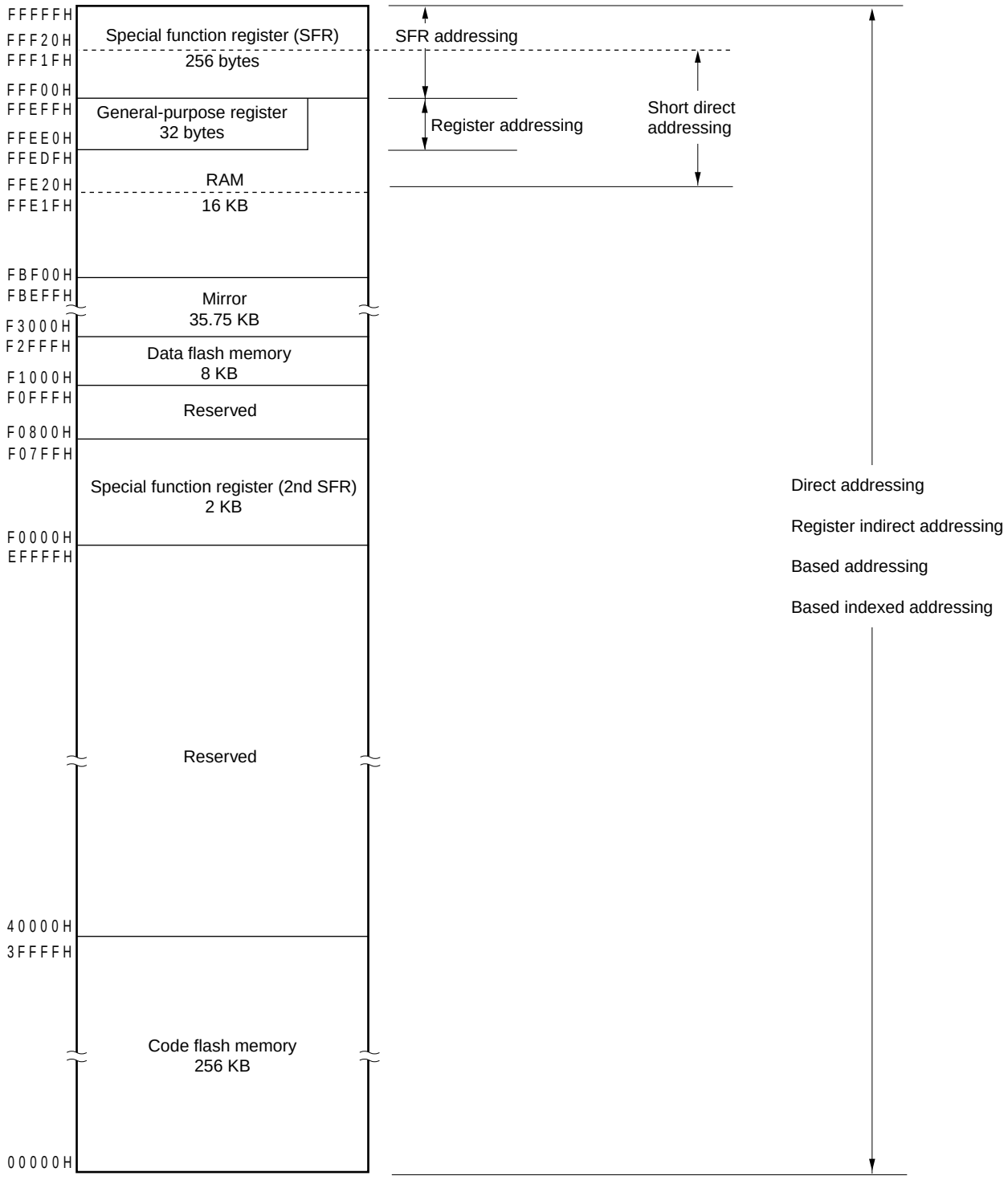
Figure 3-16. Correspondence Between Data Memory and Addressing (R5F10DMG, R5F10DPG)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-17. Correspondence Between Data Memory and Addressing
<R> **(R5F10DMJ, R5F10TPJ, R5F10DPJ, R5F10DSJ)**

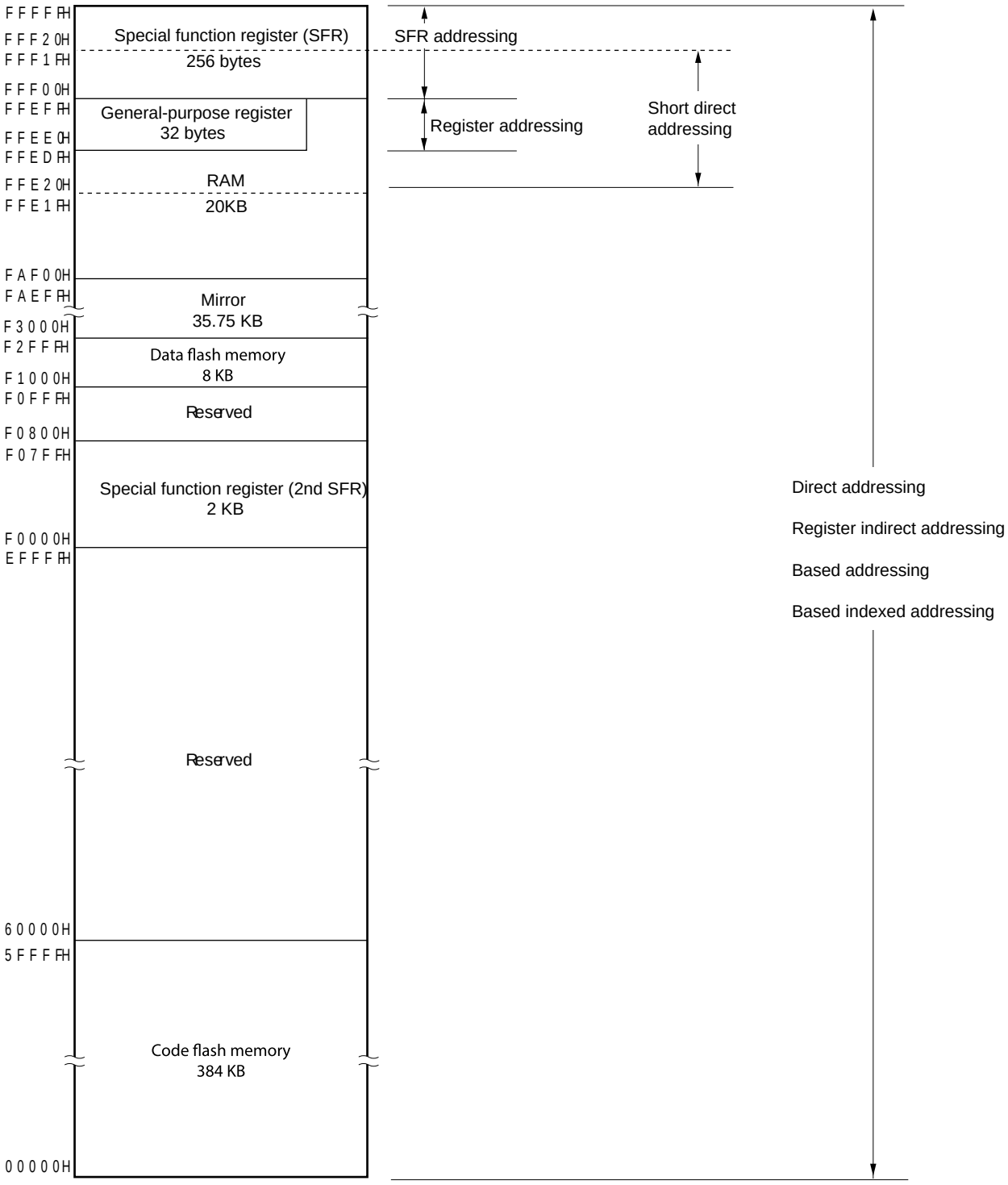


Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

<R>

Figure 3-18. Correspondence Between Data Memory and Addressing
(R5F10DSK, R5F10DPK)

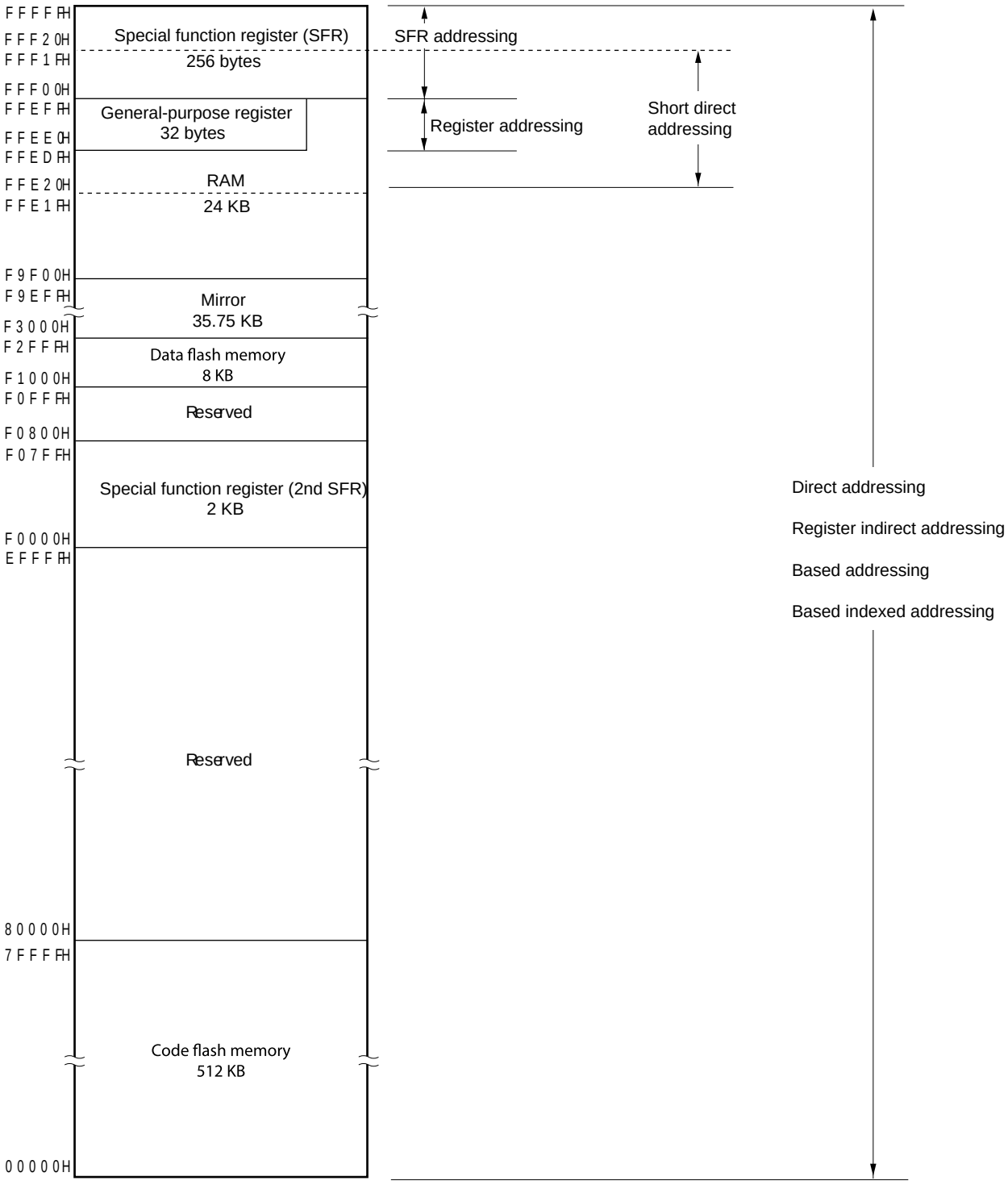


Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

<R>

Figure 3-19. Correspondence Between Data Memory and Addressing
(R5F10DSL, R5F10DP)



Note Use of part of this area is prohibited by libraries, when using the self-programming function and data flash function.

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

3.2 Processor Registers

The RL78/D1A 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)

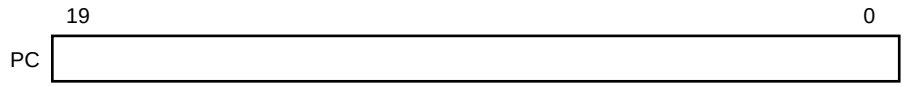
The program counter is a 20-bit register that holds the address information of the next program to be executed.

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 3-20. Format of Program Counter



(2) Program status word (PSW)

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

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

Figure 3-21. 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 maskable interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

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

(b) Zero flag (Z)

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

(c) Register bank select flags (RBS0, RBS1)

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

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

(d) Auxiliary carry flag (AC)

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

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

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **21.3 (3)**) cannot 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 3-22. 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 3-18.

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

<R> **2. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or a stack area.**

<R> **3. Do not allocate the stack area, data buffers for use by the flash library, arguments of library functions, branch destinations in the processing of vectored interrupts, or destinations or sources for DMA transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.**

<R> **4. The flash library uses RAM in self-programming and rewriting of the data flash memory.**
The target products and start address of the RAM areas used by the flash library are shown below.
R5F10CGDxFB, R5F10DGDxFB, R5F10CLDxFB, R5F10DLDxFB, R5F10CMDxFB, R5F10DMDxFB: Start address FF300H
R5F10DGExFB, R5F10DLExFB, R5F10CMExFB, R5F10DMExFB, R5F10DPExFB: Start address FEF00H
R5F10DSKxFB, R5F10DPKxFB: Start address FAF00H
R5F10DSLxFB, R5F10DPLxFB: Start address F9F00H
For the RAM areas used by the flash library, see Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944).

<R>

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).

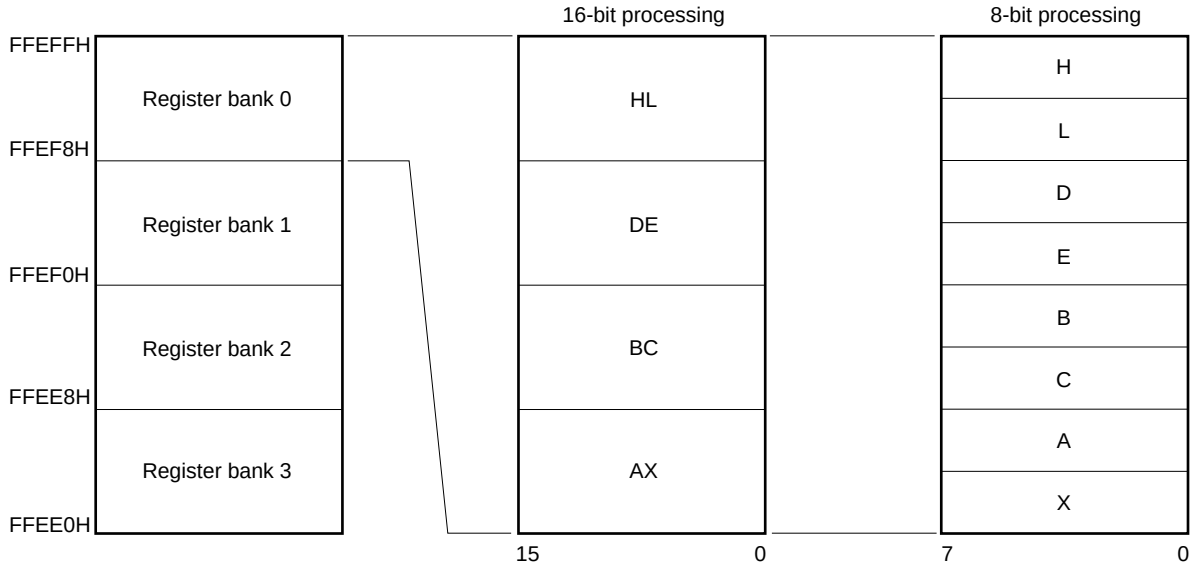
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

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 3-23. Configuration of General-Purpose Registers

(a) Function name



<R>

<R> 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.

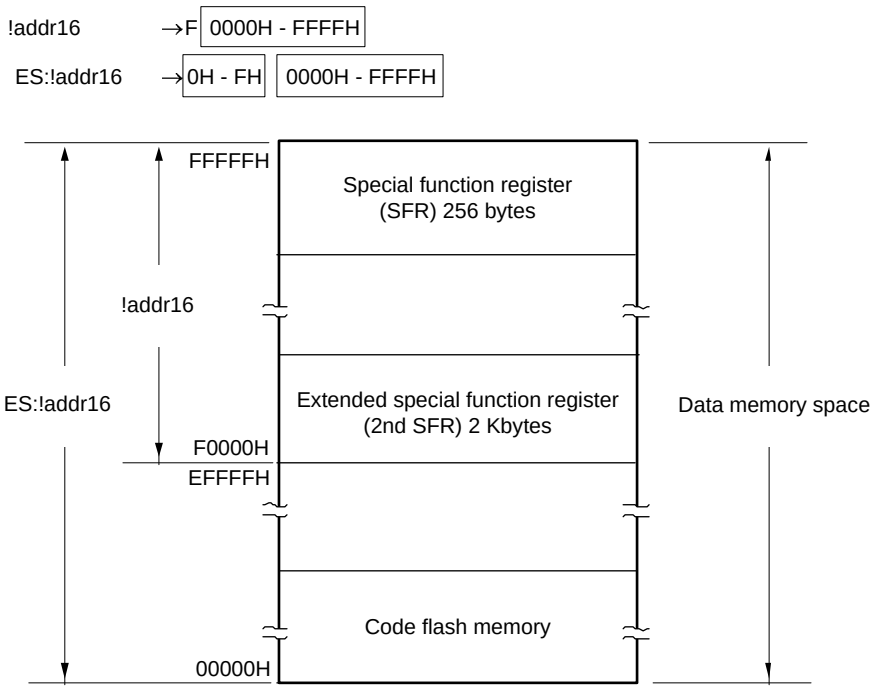
The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

Figure 3-24. Configuration of ES and CS Registers

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0
CS	0	0	0	0	CS3	CS2	CS1	CS0

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

Figure 3-25. Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

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

SFRs are allocated to the FFF00H to FFFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

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

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

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

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

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

	Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
	FFF00H	Port register 0	P0	R/W	√	√	–	00H
	FFF01H	Port register 1	P1	R/W	√	√	–	00H
	FFF02H	Port register 2	P2	R/W	√	√	–	00H
	FFF03H	Port register 3	P3	R/W	√	√	–	00H
	FFF04H	Port register 4	P4	R/W	√	√	–	00H
	FFF05H	Port register 5	P5	R/W	√	√	–	00H
	FFF06H	Port register 6	P6	R/W	√	√	–	00H
	FFF07H	Port register 7	P7	R/W	√	√	–	00H
	FFF08H	Port register 8	P8	R/W	√	√	–	00H
	FFF09H	Port register 9	P9	R/W	√	√	–	00H
<R>	FFF0AH	Port register 10 Note	P10	R/W	√	√	–	00H
<R>	FFF0BH	Port register 11 Note	P11	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	SDR00	R/W	–	–	√	0000H
	FFF11H		SDR00L	R/W	–	√	–	
	FFF12H	Serial data register 01	SDR01	R/W	–	–	√	0000H
	FFF13H		SDR01L	R/W	–	√	–	
	FFF14H	Serial data register 10	SDR10	R/W	–	–	√	0000H
	FFF15H		SDR10L	R/W	–	√	–	
	FFF16H	Serial data register 11	SDR11	R/W	–	–	√	0000H
	FFF17H		SDR11L	R/W	–	√	–	
	FFF18H	Timer data register 00	TDR00	R/W	–	–	√	0000H
	FFF19H							
	FFF1AH	Timer data register 01	TDR01	R/W	–	–	√	0000H
	FFF1BH							
	FFF1EH	10-bit A/D conversion result register	ADCR	R	–	–	√	0000H
	FFF1FH	8-bit A/D conversion result register	ADCRH	R	–	√	–	00H
	FFF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
	FFF21H	Port mode register 1	PM1	R/W	√	√	–	FFH
	FFF22H	Port mode register 2	PM2	R/W	√	√	–	FFH
	FFF23H	Port mode register 3	PM3	R/W	√	√	–	FFH
	FFF24H	Port mode register 4	PM4	R/W	√	√	–	FFH
	FFF25H	Port mode register 5	PM5	R/W	√	√	–	FFH
	FFF26H	Port mode register 6	PM6	R/W	√	√	–	FFH
	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
<R>	FFF2AH	Port mode register 10 Note	PM10	R/W	√	√	–	FFH
<R>	FFF2BH	Port mode register 11 Note	PM11	R/W	√	√	–	FFH
<R>	FFF2CH	Port mode register 12 Note	PM12	R/W	√	√	–	FFH

<R> **Note** 128-pin products only.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF2DH	Port mode register 13	PM13	R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14	R/W	√	√	–	FFH
FFF2FH	Port mode register 15	PM15	R/W	√	√	–	FFH
FFF30H	A/D converter mode register 0	ADM0	R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS	R/W	√	√	–	00H
FFF32H	A/D converter mode register 1	ADM1	R/W	√	√	–	00H
FFF34H	Watch error correction register	SUBCUDW	R/W	–	–	√	0000H
FFF36H	RTC1Hz pin select register	RTCSEL	R/W	√	√	–	00H
FFF37H	Stepper motor port mode control register	SMPC	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
FFF3CH	Serial communication pin select register 0	STSEL0	R/W	√	√	–	00H
FFF3DH	Serial communication pin select register 1	STSEL1	R/W	√	√	–	00H
FFF3EH	Timer input select else register	TISELSE	R/W	√	√	–	00H
FFF3FH	Sound generator pin select register	SGSEL	R/W	√	√	–	00H
FFF40H	LCD mode register	LCDMD	R/W	√	√	–	00H
FFF41H	LCD display mode register	LCDM	R/W	√	√	–	00H
FFF42H	LCD clock control register	LCDC0	R/W	√	√	–	00H
<R> FFF44H	LCD Bus Interface data register Note	LBDATA	R/W	–	–	√	0000H
		LBDATAL	R/W	–	√	–	00H
<R> FFF46H	LCD Bus Interface read data register Note	LBDATAR	R/W	–	–	√	0000H
		LBDATARL	R/W	–	√	–	00H
FFF48H	LIN-UART0 transmit data register	UF0TX	R/W	–	–	√	0000H
	LIN-UART0 8-bit transmit data register	UF0TXB	R/W	–	√	–	00H
FFF4AH	LIN-UART0 receive data register	UF0RX	R	–	–	√	0000H
		UF0RXB	R	–	√	–	00H
FFF4CH	LIN-UART1 transmit data register	UF1TX	R/W	–	–	√	0000H
	LIN-UART1 8-bit transmit data register	UF1TXB	R/W	–	√	–	00H
FFF4EH	LIN-UART1 receive data register	UF1RX	R	–	–	√	0000H
		UF1RXB	R	–	√	–	00H

<R> **Note** 128-pin products only.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF50H	Interval timer control register	ITMC	R/W	–	–	√	7FFFH
FFF52H	Second count register	SEC	R/W	–	√	–	00H
FFF53H	Minute count register	MIN	R/W	–	√	–	00H
FFF54H	Hour count register	HOUR	R/W	–	√	–	12H
FFF55H	Week count register	WEEK	R/W	–	√	–	00H
FFF56H	Day count register	DAY	R/W	–	√	–	01H
FFF57H	Month count register	MONTH	R/W	–	√	–	01H
FFF58H	Year count register	YEAR	R/W	–	√	–	00H
FFF59H	Watch error correction register	SUBCUD	R/W	–	√	–	00H
FFF5AH	Alarm minute register	ALARMWM	R/W	–	√	–	00H
FFF5BH	Alarm hour register	ALARMWH	R/W	–	√	–	12H[Note]
FFF5CH	Alarm week register	ALARMWW	R/W	–	√	–	00H
FFF5DH	Real time counter control register 0	RTCC0	R/W	√	√	–	00H
FFF5EH	Real time counter control register 1	RTCC1	R/W	√	√	–	00H
FFF64H	Timer data register 02	TDR02	R/W	–	–	√	0000H
FFF66H	Timer data register 03	TDR03	R/W	–	–	√	0000H
FFF68H	Timer data register 04	TDR04	R/W	–	–	√	0000H
FFF6AH	Timer data register 05	TDR05	R/W	–	–	√	0000H
FFF6CH	Timer data register 06	TDR06	R/W	–	–	√	0000H
FFF6EH	Timer data register 07	TDR07	R/W	–	–	√	0000H
FFF70H	Timer data register 10	TDR10	R/W	–	–	√	0000H
FFF72H	Timer data register 11	TDR11	R/W	–	–	√	0000H
FFF74H	Timer data register 12	TDR12	R/W	–	–	√	0000H
FFF76H	Timer data register 13	TDR13	R/W	–	–	√	0000H
FFF78H	Timer data register 14	TDR14	R/W	–	–	√	0000H
FFF7AH	Timer data register 15	TDR15	R/W	–	–	√	0000H
FFF7CH	Timer data register 16	TDR16	R/W	–	–	√	0000H
FFF7EH	Timer data register 17	TDR17	R/W	–	–	√	0000H
FFF90H	Timer data register 20	TDR20	R/W	–	–	√	0000H
FFF92H	Timer data register 21	TDR21	R/W	–	–	√	0000H
FFF94H	Timer data register 22	TDR22	R/W	–	–	√	0000H
FFF96H	Timer data register 23	TDR23	R/W	–	–	√	0000H
FFF98H	Timer data register 24	TDR24	R/W	–	–	√	0000H
FFF9AH	Timer data register 25	TDR25	R/W	–	–	√	0000H
FFF9CH	Timer data register 26	TDR26	R/W	–	–	√	0000H
FFF9EH	Timer data register 27	TDR27	R/W	–	–	√	0000H

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA0H	Clock operation mode control register	CMC		R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	07H
FFFA4H	System clock control register	CKC		R/W	√	√	–	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined Note1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	–	00H Note2
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	–	00H/01H/81H Note3
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1AH/9AH Note4
FFFACH	CRC input register	CRCIN		R/W	–	√	–	00H
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0L	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H	DMA RAM address register 0H	DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1L	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H	DMA RAM address register 1H	DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0L	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H	DMA byte count register 0H	DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1L	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H	DMA byte count register 1H	DBC1H		R/W	–	√		00H
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
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

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

2. The reset value of the LVIM register varies depending on the reset source.

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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFDAH	Priority specification flag register 02L	PR03L	PR03	R/W	√	√	√	FFH
FFFDBH	Priority specification flag register 03H	PR03H			√	√		
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
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/division data register A (L)	MDAL/MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH/MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH/MULOH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL/MULOL		R/W	–	–	√	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.

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

Table 3-6 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding extended SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

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

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

Table 3-6. Extended SFR (2nd SFR) List (1/21)

	Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
	F0010H	A/D converter mode register 2	ADM2	R/W	√	√	–	00H
	F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	–	√	–	FFH
	F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	–	√	–	00H
	F0013H	A/D test register	ADTES	R/W	–	√	–	00H
<R>	F0014H	RESET output control register Note	RESOC	R/W	√	√	–	00H
<R>	F0016H	STOP status output control register Note	STPSTC	R/W	√	√	–	00H
<R>	F0018H	LCD Bus Interface mode register Note	LBCTL	R/W	√	√	–	00H
<R>	F0019H	LCB Bus Interface cycle control register Note	LBCYC	R/W	√	√	–	02H
<R>	F001AH	LCB Bus Interface wait control register Note	LBWST	R/W	√	√	–	00H
<R>	F001FH	DMA trigger selection register Note	DMATSEL	R/W	√	√	–	00H
	F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
	F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
	F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
	F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	01H
	F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
	F0036H	Pull-up resistor option register 6	PU6	R/W	√	√	–	00H
	F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
	F0038H	Pull-up resistor option register 8	PU8	R/W	√	√	–	00H
	F0039H	Pull-up resistor option register 9	PU9	R/W	√	√	–	00H
	F003DH	Pull-up resistor option register 13	PU13	R/W	√	√	–	00H
	F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
	F0040H	Port input mode register 0	PIM0	R/W	√	√	–	00H
<R>	F003AH	Pull-up resistor option register 10 Note	PU10	R/W	√	√	–	00H
<R>	F003BH	Pull-up resistor option register 11 Note	PU11	R/W	√	√	–	00H
<R>	F003CH	Pull-up resistor option register 12 Note	PU12	R/W	√	√	–	00H
	F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
	F0043H	Port input mode register 3	PIM3	R/W	√	√	–	00H
	F0045H	Port input mode register 5	PIM5	R/W	√	√	–	00H
	F0046H	Port input mode register 6	PIM6	R/W	√	√	–	00H
	F0047H	Port input mode register 7	PIM7	R/W	√	√	–	00H
<R>	F004BH	Port input mode register 11 Note	PIM11	R/W	√	√	–	00H
	F004DH	Port input mode register 13	PIM13	R/W	√	√	–	00H
	F0050H	LCD port function register 0	LCDPF0	R/W	√	√	–	00H
	F0051H	LCD port function register 1	LCDPF1	R/W	√	√	–	00H
	F0053H	LCD port function register 3	LCDPF3	R/W	√	√	–	00H
<R>	F0054H	LCD port function register 4 Note	LCDPF4	R/W	√	√	–	00H
	F0055H	LCD port function register 5	LCDPF5	R/W	√	√	–	00H
	F0057H	LCD port function register 7	LCDPF7	R/W	√	√	–	00H
	F0058H	LCD port function register 8	LCDPF8	R/W	√	√	–	00H
	F0059H	LCD port function register 9	LCDPF9	R/W	√	√	–	00H
<R>	F005AH	LCD port function register 10 Note	LCDPF10	R/W	√	√	–	00H
<R>	F005BH	LCD port function register 11 Note	LCDPF11	R/W	√	√	–	00H
<R>	F005CH	LCD port function register 12 Note	LCDPF12	R/W	√	√	–	00H
	F005DH	LCD port function register 13	LCDPF13	R/W	√	√	–	00H
	F0060H	Noise filter enable register for each channel of TAU unit0	TNFEN0	R/W	√	√	–	00H
	F0061H	Sampling clock select of noise filter for unit0 (2set)	TNFSMP0	R/W	√	√	–	00H

<R> **Note** 128-pin products only.

Table 3-6. Extended SFR (2nd SFR) List (2/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0062H	Noise filter clock select register for each channel of TAU unit0	TNFCS0	R/W	√	√	–	00H
F0064H	Noise filter enable register for each channel of TAU unit1	TNFEN1	R/W	√	√	–	00H
F0065H	Sampling clock select of noise filter for unit1 (2set)	TNFSMP1	R/W	√	√	–	00H
F0066H	Noise filter clock select register for each channel of TAU unit1	TNFCS1	R/W	√	√	–	00H
F0068H	Noise filter enable register for each channel of TAU unit2	TNFEN2	R/W	√	√	–	00H
F0069H	Sampling clock select of noise filter for unit2(2set)	TNFSMP2	R/W	√	√	–	00H
F006AH	Noise filter clock select register for each channel of TAU unit2	TNFCS2	R/W	√	√	–	00H
F006EH	A/D port configuration register	ADPC	R/W	–	√	–	00H
F006FH	Port output mode register	POM	R/W	√	√	–	00H
F0070H	Timer input select register 00	TIS00	R/W	√	√	–	00H
F0071H	Timer input select register 01	TIS01	R/W	√	√	–	00H
F0072H	Timer input select register 10	TIS10	R/W	√	√	–	00H
F0073H	Timer input select register 11	TIS11	R/W	√	√	–	00H
F0074H	Timer input select register 20	TIS20	R/W	√	√	–	00H
F0075H	Timer input select register 21	TIS21	R/W	√	√	–	00H
F0076H	Timer output select register 00	TOS00	R/W	√	√	–	00H
F0077H	Timer output select register 01	TOS01	R/W	√	√	–	00H
F0078H	Illegal-memory access detection control register	IAWCTL	R/W	–	√	–	00H
F0079H	Timer output select register 10	TOS10	R/W	√	√	–	00H
F007AH	Timer output select register 11	TOS11	R/W	√	√	–	00H
F007BH	Timer output select register 20	TOS20	R/W	√	√	–	00H
F007CH	Timer output select register 21	TOS21	R/W	√	√	–	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	–	√	–	Undefined[Note]
F00E0H	Multiplication/division data register C (L)	MDCL	R/W	–	–	√	0000H
F00E2H	Multiplication/division data register C (H)	MDCH	R/W	–	–	√	0000H
F00E8H	Multiplication/division control register	MDUC	R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	–	00H
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
F00F5H	RAM parity error control register	RPECTL	R/W	√	√	–	00H
F00F8H	FMP clock division selction register	MDIV	R/W	–	√	–	00H
F00F9H	RTC clock selection register	RTCCL	R/W	√	√	–	00H
F00FAH	CLM reset control flag register	RESFCLM	R	–	√	–	Undefined
F00FBH	POC reset confirm register	POCRES	R/W	√	√	–	00H
F00FCH	Specific register manipulation protection register	GUARD	R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ	R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00	R	–	–	√	0000H
		SSR00L	R	–	√	–	
F0102H	Serial status register 01	SSR01	R	–	–	√	0000H
		SSR01L	R	–	√	–	

Note The reset value differs for each chip.

Table 3-6. Extended SFR (2nd SFR) List (3/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0104H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0106H	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F0108H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F010AH	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F010CH	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F010EH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F0110H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0112H	Serial channel start trigger register 0	SS0L	SS0	R/W	√	√	√	0000H
F0114H	Serial channel stop trigger register 0	ST0L	ST0	R/W	√	√	√	0000H
F0116H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0118H	Serial output register 0	SO0		R/W	–	–	√	0303H
F011AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F0120H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0128H	PLL status register	PLLSTS		R	√	√	–	00H
F0129H	PLL control register	PLLCTL		R/W	√	√	–	00H
F0130H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0132H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0134H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0136H	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F0138H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F013AH	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F013CH	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F013EH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F0140H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0142H	Serial channel start trigger register 1	SS1L	SS1	R/W	√	√	√	0000H
F0144H	Serial channel stop trigger register 1	ST1L	ST1	R/W	√	√	√	0000H
F0146H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0148H	Serial output register 1	SO1L	SO1	R/W	–	√	√	0303H
F014AH	Serial output enabel register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F0150H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0158H	Compare control register 4	MCMPC4		R/W	√	√	–	00H
F015CH	ZPD detection voltage setting register0/ZPD analog input control	ZPDS0		R/W	√	√	–	00H
F015DH	ZPD detection voltage setting register1/ZPD analog input control	ZPDS1		R/W	√	√	–	00H
F015EH	ZPD flag detection clock setting register	CMPCTL		R/W	√	√	–	00H
F015FH	ZPD operation control register	ZPDEN		R/W	√	√	–	00H
F0160H	Compare control register 0	MCNTC0		R/W	√	√	–	00H
F0162H	Compare register 1HW	MCMP1HW		R/W	–	–	√	0000H
	Compare register 10	MCMP10		R/W	–	√	–	00H
F0163H	Compare register 11	MCMP11		R/W	–	√	–	00H
F0164H	Compare register 2HW	MCMP2HW		R/W	–	–	√	0000H
	Compare register 20	MCMP20		R/W	–	√	–	00H
F0165H	Compare register 21	MCMP21		R/W	–	√	–	00H
F0166H	Compare register 3HW	MCMP3HW		R/W	–	–	√	0000H
	Compare register 30	MCMP30		R/W	–	√	–	00H
F0167H	Compare register 31	MCMP31		R/W	–	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (4/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0168H	Compare register 4HW	MCMP4HW	R/W	–	–	√	0000H
	Compare register 40	MCMP40	R/W	–	√	–	00H
F0169H	Compare register 41	MCMP41	R/W	–	√	–	00H
F016AH	Compare control register 1	MCMPC1	R/W	√	√	–	00H
F016CH	Compare control register 2	MCMPC2	R/W	√	√	–	00H
F016EH	Compare control register 3	MCMPC3	R/W	√	√	–	00H
F0170H	DMA SFR address register 2	DSA2	R/W	–	√	–	00H
F0171H	DMA SFR address register 3	DSA3	R/W	–	√	–	00H
F0172H	DMA RAM address register 2	DRA2	R/W	–	–	√	0000H
	DMA RAM address register 2L	DRA2L	R/W	–	√	–	00H
F0173H	DMA RAM address register 2H	DRA2H	R/W	–	√	–	00H
F0174H	DMA RAM address register 3	DRA3	R/W	–	–	√	0000H
	DMA RAM address register 3L	DRA3L	R/W	–	√	–	00H
F0175H	DMA RAM address register 3H	DRA3H	R/W	–	√	–	00H
F0176H	DMA byte count register 2	DBC2	R/W	–	–	√	0000H
	DMA byte count register 2L	DBC2L	R/W	–	√	–	00H
F0177H	DMA byte count register 2H	DBC2H	R/W	–	√	–	00H
F0178H	DMA byte count register 3	DBC3	R/W	–	–	√	0000H
	DMA byte count register 3L	DBC3L	R/W	–	√	–	00H
F0179H	DMA byte count register 3H	DBC3H	R/W	–	√	–	00H
F017AH	DMA mode control register 2	DMC2	R/W	√	√	–	00H
F017BH	DMA mode control register 3	DMC3	R/W	√	√	–	00H
F017CH	DMA operation control register 2	DRC2	R/W	√	√	–	00H
F017DH	DMA operation control register 3	DRC3	R/W	√	√	–	00H
F017FH	DMA all-channel forced wait register	DWAITALL	R/W	√	√	–	00H
F0180H	Timer counter register 00	TCR00	R	–	–	√	FFFFH
F0182H	Timer counter register 01	TCR01	R	–	–	√	FFFFH
F0184H	Timer counter register 02	TCR02	R	–	–	√	FFFFH
F0186H	Timer counter register 03	TCR03	R	–	–	√	FFFFH
F0188H	Timer counter register 04	TCR04	R	–	–	√	FFFFH
F018AH	Timer counter register 05	TCR05	R	–	–	√	FFFFH
F018CH	Timer counter register 06	TCR06	R	–	–	√	FFFFH
F018EH	Timer counter register 07	TCR07	R	–	–	√	FFFFH
F0190H	Timer mode register 00	TMR00	R/W	–	–	√	0000H
F0192H	Timer mode register 01	TMR01	R/W	–	–	√	0000H
F0194H	Timer mode register 02	TMR02	R/W	–	–	√	0000H
F0196H	Timer mode register 03	TMR03	R/W	–	–	√	0000H
F0198H	Timer mode register 04	TMR04	R/W	–	–	√	0000H
F019AH	Timer mode register 05	TMR05	R/W	–	–	√	0000H
F019CH	Timer mode register 06	TMR06	R/W	–	–	√	0000H
F019EH	Timer mode register 07	TMR07	R/W	–	–	√	0000H

Table 3-6. Extended SFR (2nd SFR) List (5/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B6H	Timer clock select register 0	TPS0		R/W	–	–	√	0000H
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01C0H	Timer counter register 10	TCR10		R	–	–	√	FFFFH
F01C2H	Timer counter register 11	TCR11		R	–	–	√	FFFFH
F01C4H	Timer counter register 12	TCR12		R	–	–	√	FFFFH
F01C6H	Timer counter register 13	TCR13		R	–	–	√	FFFFH
F01C8H	Timer counter register 14	TCR14		R	–	–	√	FFFFH
F01CAH	Timer counter register 15	TCR15		R	–	–	√	FFFFH
F01CCH	Timer counter register 16	TCR16		R	–	–	√	FFFFH
F01CEH	Timer counter register 17	TCR17		R	–	–	√	FFFFH
F01D0H	Timer mode register 10	TMR10		R/W	–	–	√	0000H
F01D2H	Timer mode register 11	TMR11		R/W	–	–	√	0000H
F01D4H	Timer mode register 12	TMR12		R/W	–	–	√	0000H
F01D6H	Timer mode register 13	TMR13		R/W	–	–	√	0000H
F01D8H	Timer mode register 14	TMR14		R/W	–	–	√	0000H
F01DAH	Timer mode register 15	TMR15		R/W	–	–	√	0000H
F01DCH	Timer mode register 16	TMR16		R/W	–	–	√	0000H
F01DEH	Timer mode register 17	TMR17		R/W	–	–	√	0000H
F01E0H	Timer status register 10	TSR10L	TSR10	R	–	√	√	0000H
F01E2H	Timer status register 11	TSR11L	TSR11	R	–	√	√	0000H
F01E4H	Timer status register 12	TSR12L	TSR12	R	–	√	√	0000H
F01E6H	Timer status register 13	TSR13L	TSR13	R	–	√	√	0000H
F01E8H	Timer status register 14	TSR14L	TSR14	R	–	√	√	0000H
F01EAH	Timer status register 15	TSR15L	TSR15	R	–	√	√	0000H
F01ECH	Timer status register 16	TSR16L	TSR16	R	–	√	√	0000H
F01EEH	Timer status register 17	TSR17L	TSR17	R	–	√	√	0000H

Table 3-6. Extended SFR (2nd SFR) List (6/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01F0H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H
F01F2H	Timer channel start register 1	TS1L	TS1	R/W	√	√	√	0000H
F01F4H	Timer channel stop register 1	TT1L	TT1	R/W	√	√	√	0000H
F01F6H	Timer clock select register 1	TPS1		R/W	–	–	√	0000H
F01F8H	Timer output register 1	TO1L	TO1	R/W	–	√	√	0000H
F01FAH	Timer output enable register 1	TOE1L	TOE1	R/W	√	√	√	0000H
F01FCH	Timer output level register 1	TOL1L	TOL1	R/W	–	√	√	0000H
F01FEH	Timer output mode register 1	TOM1L	TOM1	R/W	–	√	√	0000H
F0200H	Timer counter register 20	TCR20		R	–	–	√	FFFFH
F0202H	Timer counter register 21	TCR21		R	–	–	√	FFFFH
F0204H	Timer counter register 22	TCR22		R	–	–	√	FFFFH
F0206H	Timer counter register 23	TCR23		R	–	–	√	FFFFH
F0208H	Timer counter register 24	TCR24		R	–	–	√	FFFFH
F020AH	Timer counter register 25	TCR25		R	–	–	√	FFFFH
F020CH	Timer counter register 26	TCR26		R	–	–	√	FFFFH
F020EH	Timer counter register 27	TCR27		R	–	–	√	FFFFH
F0210H	Timer mode register 20	TMR20		R/W	–	–	√	0000H
F0212H	Timer mode register 21	TMR21		R/W	–	–	√	0000H
F0214H	Timer mode register 22	TMR22		R/W	–	–	√	0000H
F0216H	Timer mode register 23	TMR23		R/W	–	–	√	0000H
F0218H	Timer mode register 24	TMR24		R/W	–	–	√	0000H
F021AH	Timer mode register 25	TMR25		R/W	–	–	√	0000H
F021CH	Timer mode register 26	TMR26		R/W	–	–	√	0000H
F021EH	Timer mode register 27	TMR27		R/W	–	–	√	0000H
F0220H	Timer status register 20	TSR20L	TSR20	R	–	√	√	0000H
F0222H	Timer status register 21	TSR21L	TSR21	R	–	√	√	0000H
F0224H	Timer status register 22	TSR22L	TSR22	R	–	√	√	0000H
F0226H	Timer status register 23	TSR23L	TSR23	R	–	√	√	0000H
F0228H	Timer status register 24	TSR24L	TSR24	R	–	√	√	0000H
F022AH	Timer status register 25	TSR25L	TSR25	R	–	√	√	0000H
F022CH	Timer status register 26	TSR26L	TSR26	R	–	√	√	0000H
F022EH	Timer status register 27	TSR27L	TSR27	R	–	√	√	0000H
F0230H	Timer channel enable status register 2	TE2L	TE2	R	√	√	√	0000H
F0232H	Timer channel start trigger register 2	TS2L	TS2	R/W	√	√	√	0000H
F0234H	Timer channel stop trigger register 2	TT2L	TT2	R/W	√	√	√	0000H
F0236H	Timer clock select register 2	TPS2		R/W	–	–	√	0000H
F0238H	Timer output register 2	TO2L	TO2	R/W	–	√	√	0000H
F023AH	Timer output enable register 2	TOE2L	TOE2	R/W	√	√	√	0000H
F023CH	Timer output level register 2	TOL2L	TOL2	R/W	–	√	√	0000H
F023EH	Timer output mode register 2	TOM2L	TOM2	R/W	–	√	√	0000H

Table 3-6. Extended SFR (2nd SFR) List (7/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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
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
F0248H	LIN-UART0 status clear register	UF0STC	R/W	–	–	√	0000H
F024AH	LIN-UART0 wait transmit data register	UF0WTX	R/W	–	–	√	0000H
	LIN-UART0 8-bit wait transmit data register	UF0WTXB	R/W	–	√	–	00H
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
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
F0264H	LIN-UART1 option control register 1	UF1OPT1	R/W	√	√	–	00H
F0265H	LIN-UART0 option control register 2	UF1OPT2	R/W	√	√	–	00H
F0266H	LIN-UART1 status register	UF1STR	R	–	–	√	0000H
F0268H	LIN-UART1 status clear register	UF1STC	R/W	–	–	√	0000H
F026AH	LIN-UART1 wait transmit data register	UF1WTX	R/W	–	–	√	0000H
	LIN-UART1 8-bit wait transmit data register	UF1WTXB	R/W	–	√	–	00H
F026EH	LIN-UART1 ID setting register	UF1ID	R/W	–	√	–	00H
F026FH	LIN-UART1 buffer register 0	UF1BUF0	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

Table 3-6. Extended SFR (2nd SFR) List (8/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0280H	Frequency register SG0FL	SG0FL	R/W	–	–	√	0000H
F0282H	Frequency register SG0FH	SG0FH	R/W	–	–	√	0000H
F0284H	Amplitude register	SG0PWM	R/W	–	–	√	0000H
F0286H	Duration factor register	SG0SDF	R/W	–	√	–	00H
F0287H	Control register	SG0CTL	R/W	√	√	–	0000H
F0288H	Interrupt threshold register	SG0ITH	R/W	–	–	√	0000H
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	√	√	–	00H
F02F2H	Flash memory CRC operation result register	PGCRCL	R/W	–	–	√	0000H
F02FAH	CRC data register	CRCD	R/W	–	–	√	0000H
F0300H	LCD display data memory0	SEG0	R/W	–	√	–	00H
F0301H	LCD display data memory1	SEG1	R/W	–	√	–	00H
F0302H	LCD display data memory2	SEG2	R/W	–	√	–	00H
F0303H	LCD display data memory3	SEG3	R/W	–	√	–	00H
F0304H	LCD display data memory4	SEG4	R/W	–	√	–	00H
F0305H	LCD display data memory5	SEG5	R/W	–	√	–	00H
F0306H	LCD display data memory6	SEG6	R/W	–	√	–	00H
F0307H	LCD display data memory7	SEG7	R/W	–	√	–	00H
F0308H	LCD display data memory8	SEG8	R/W	–	√	–	00H
F0309H	LCD display data memory9	SEG9	R/W	–	√	–	00H
F030AH	LCD display data memory10	SEG10	R/W	–	√	–	00H
F030BH	LCD display data memory11	SEG11	R/W	–	√	–	00H
F030CH	LCD display data memory12	SEG12	R/W	–	√	–	00H
F030DH	LCD display data memory13	SEG13	R/W	–	√	–	00H
F030EH	LCD display data memory14	SEG14	R/W	–	√	–	00H
F030FH	LCD display data memory15	SEG15	R/W	–	√	–	00H
F0310H	LCD display data memory16	SEG16	R/W	–	√	–	00H
F0311H	LCD display data memory17	SEG17	R/W	–	√	–	00H
F0312H	LCD display data memory18	SEG18	R/W	–	√	–	00H
F0313H	LCD display data memory19	SEG19	R/W	–	√	–	00H
F0314H	LCD display data memory20	SEG20	R/W	–	√	–	00H
F0315H	LCD display data memory21	SEG21	R/W	–	√	–	00H
F0316H	LCD display data memory22	SEG22	R/W	–	√	–	00H
F0317H	LCD display data memory23	SEG23	R/W	–	√	–	00H
F0318H	LCD display data memory24	SEG24	R/W	–	√	–	00H
F0319H	LCD display data memory25	SEG25	R/W	–	√	–	00H
F031AH	LCD display data memory26	SEG26	R/W	–	√	–	00H
F031BH	LCD display data memory27	SEG27	R/W	–	√	–	00H
F031CH	LCD display data memory28	SEG28	R/W	–	√	–	00H
F031DH	LCD display data memory29	SEG29	R/W	–	√	–	00H
F031EH	LCD display data memory30	SEG30	R/W	–	√	–	00H
F031FH	LCD display data memory31	SEG31	R/W	–	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (9/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0320H	LCD display data memory32	SEG32	R/W	–	√	–	00H
F0321H	LCD display data memory33	SEG33	R/W	–	√	–	00H
F0322H	LCD display data memory34	SEG34	R/W	–	√	–	00H
F0323H	LCD display data memory35	SEG35	R/W	–	√	–	00H
F0324H	LCD display data memory36	SEG36	R/W	–	√	–	00H
F0325H	LCD display data memory37	SEG37	R/W	–	√	–	00H
F0326H	LCD display data memory38	SEG38	R/W	–	√	–	00H
F0327H	LCD display data memory39	SEG39	R/W	–	√	–	00H
F0328H	LCD display data memory40	SEG40	R/W	–	√	–	00H
F0329H	LCD display data memory41	SEG41	R/W	–	√	–	00H
F032AH	LCD display data memory42	SEG42	R/W	–	√	–	00H
F032BH	LCD display data memory43	SEG43	R/W	–	√	–	00H
F032CH	LCD display data memory44	SEG44	R/W	–	√	–	00H
F032DH	LCD display data memory45	SEG45	R/W	–	√	–	00H
F032EH	LCD display data memory46	SEG46	R/W	–	√	–	00H
F032FH	LCD display data memory47	SEG47	R/W	–	√	–	00H
F0330H	LCD display data memory48	SEG48	R/W	–	√	–	00H
F0331H	LCD display data memory49	SEG49	R/W	–	√	–	00H
F0332H	LCD display data memory50	SEG50	R/W	–	√	–	00H
F0333H	LCD display data memory51	SEG51	R/W	–	√	–	00H
F0334H	LCD display data memory52	SEG52	R/W	–	√	–	00H
<R> F0335H	LCD display data memory53 Note	SEG53	R/W	–	√	–	00H
F0340H	CAN1 global module control register	C1GMCTRL	R/W	–	–	√	0000H
F0342H	CAN1 global module clock select register	C1GMCS	R/W	–	√	–	0FH
F0346H	CAN1 global block transmission control register	C1GMABT	R/W	–	–	√	0000H
F0348H	CAN1 global block transmission delay setting register	C1GMABTD	R/W	–	√	–	00H
F0380H	CAN1 module mask 1 register L	C1MASK1L	R/W	–	–	√	Undefined
F0382H	CAN1 module mask 1 register H	C1MASK1H	R/W	–	–	√	Undefined
F0384H	CAN1 module mask 2 register L	C1MASK2L	R/W	–	–	√	Undefined
F0386H	CAN1 module mask 2 register H	C1MASK2H	R/W	–	–	√	Undefined
F0388H	CAN1 module mask 3 register L)	C1MASK3L	R/W	–	–	√	Undefined
F038AH	CAN1 module mask 3 register H	C1MASK3H	R/W	–	–	√	Undefined
F038CH	CAN1 module mask 4 register L	C1MASK4L	R/W	–	–	√	Undefined
F038EH	CAN1 module mask 4 register H	C1MASK4H	R/W	–	–	√	Undefined
F0390H	CAN1 module control register	C1CTRL	R/W	–	–	√	0000H
F0392H	CAN1 module last error information register	C1LEC	R/W	–	√	–	00H
F0393H	CAN1 module information register	C1INFO	R	–	√	–	00H
F0394H	CAN1 module error counter register	C1ERC	R	–	–	√	0000H
F0396H	CAN1 module interrupt enable register	C1IE	R/W	–	–	√	0000H
F0398H	CAN1 module interrupt status register	C1INTS	R/W	–	–	√	0000H

<R> **Note** 128-pin products only.

Table 3-6. Extended SFR (2nd SFR) List (10/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F039AH	CAN1 module bit rate prescaler register	C1BRP	R/W	–	√	–	FFH
F039CH	CAN1 module bit rate register	C1BTR	R/W	–	–	√	370FH
F039EH	CAN1 module last in-pointer register	C1LIPT	R	–	√	–	Undefined
F03A0H	CAN1 module receive history list register	C1RGPT	R/W	–	–	√	xx02H
F03A2H	CAN1 module last out-pointer register	C1LOPT	R	–	√	–	Undefined
F03A4H	CAN1 module transmit history list register	C1TGPT	R/W	–	–	√	xx02H
F03A6H	CAN1 module time stamp register	C1TS	R/W	–	–	√	0000H
F0400H	CAN1 message data byte 01 register 00	C1MDB0100	R/W	–	–	√	Undefined
F0400H	CAN1 message data byte 0 register 00	C1MDB000	R/W	–	√	–	Undefined
F0401H	CAN1 message data byte 1 register 00	C1MDB100	R/W	–	√	–	Undefined
F0402H	CAN1 message data byte 23 register 00	C1MDB2300	R/W	–	–	√	Undefined
F0402H	CAN1 message data byte 2 register 00	C1MDB200	R/W	–	√	–	Undefined
F0403H	CAN1 message data byte 3 register 00	C1MDB300	R/W	–	√	–	Undefined
F0404H	CAN1 message data byte 45 register 00	C1MDB4500	R/W	–	–	√	Undefined
F0404H	CAN1 message data byte 4 register 00	C1MDB400	R/W	–	√	–	Undefined
F0405H	CAN1 message data byte 5 register 00	C1MDB500	R/W	–	√	–	Undefined
F0406H	CAN1 message data byte 67 register 00	C1MDB6700	R/W	–	–	√	Undefined
F0406H	CAN1 message data byte 6 register 00	C1MDB600	R/W	–	√	–	Undefined
F0407H	CAN1 message data byte 7 register 00	C1MDB700	R/W	–	√	–	Undefined
F0408H	CAN1 message data length register 00	C1MDLC00	R/W	–	√	–	0xH
F0409H	CAN1 message Configuration register 00	C1MCONF00	R/W	–	√	–	Undefined
F040AH	CAN1 message ID register 00L	C1MIDL00	R/W	–	–	√	Undefined
F040CH	CAN1 message ID register 00H	C1MIDH00	R/W	–	–	√	Undefined
F040EH	CAN1 message control register 00	C1MCTRL00	R/W	–	–	√	Undefined
F0410H	CAN1 message data byte 01 register 01	C1MDB0101	R/W	–	–	√	Undefined
F0410H	CAN1 message data byte 0 register 01	C1MDB001	R/W	–	√	–	Undefined
F0411H	CAN1 message data byte 1 register 01	C1MDB101	R/W	–	√	–	Undefined
F0412H	CAN1 message data byte 23 register 01	C1MDB2301	R/W	–	–	√	Undefined
F0412H	CAN1 message data byte 2 register 01	C1MDB201	R/W	–	√	–	Undefined
F0413H	CAN1 message data byte 3 register 01	C1MDB301	R/W	–	√	–	Undefined
F0414H	CAN1 message data byte 45 register 01	C1MDB4501	R/W	–	–	√	Undefined
F0414H	CAN1 message data byte 4 register 01	C1MDB401	R/W	–	√	–	Undefined
F0415H	CAN1 message data byte 5 register 01	C1MDB501	R/W	–	√	–	Undefined
F0416H	CAN1 message data byte 67 register 01	C1MDB6701	R/W	–	–	√	Undefined
F0416H	CAN1 message data byte 6 register 01	C1MDB601	R/W	–	√	–	Undefined
F0417H	CAN1 message data byte 7 register 01	C1MDB701	R/W	–	√	–	Undefined
F0418H	CAN1 message data length register 01	C1MDLC01	R/W	–	√	–	0xH
F0419H	CAN1 message Configuration register 01	C1MCONF01	R/W	–	√	–	Undefined
F041AH	CAN1 message ID register 01L	C1MIDL01	R/W	–	–	√	Undefined
F041CH	CAN1 message ID register 01H	C1MIDH01	R/W	–	–	√	Undefined
F041EH	CAN1 message control register 01	C1MCTRL01	R/W	–	–	√	Undefined

Table 3-6. Extended SFR (2nd SFR) List (11/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0420H	CAN1 message data byte 01 register 02	C1MDB0102	R/W	–	–	√	Undefined
F0420H	CAN1 message data byte 0 register 02	C1MDB002	R/W	–	√	–	Undefined
F0421H	CAN1 message data byte 1 register 02	C1MDB102	R/W	–	√	–	Undefined
F0422H	CAN1 message data byte 23 register 02	C1MDB2302	R/W	–	–	√	Undefined
F0422H	CAN1 message data byte 2 register 02	C1MDB202	R/W	–	√	–	Undefined
F0423H	CAN1 message data byte 3 register 02	C1MDB302	R/W	–	√	–	Undefined
F0424H	CAN1 message data byte 45 register 02	C1MDB4502	R/W	–	–	√	Undefined
F0424H	CAN1 message data byte 4 register 02	C1MDB402	R/W	–	√	–	Undefined
F0425H	CAN1 message data byte 5 register 02	C1MDB502	R/W	–	√	–	Undefined
F0426H	CAN1 message data byte 67 register 02	C1MDB6702	R/W	–	–	√	Undefined
F0426H	CAN1 message data byte 6 register 02	C1MDB602	R/W	–	√	–	Undefined
F0427H	CAN1 message data byte 7 register 02	C1MDB702	R/W	–	√	–	Undefined
F0428H	CAN1 message data length register 02	C1MDLC02	R/W	–	√	–	0xH
F0429H	CAN1 message Configuration register 02	C1MCONF02	R/W	–	√	–	Undefined
F042AH	CAN1 message ID register 02L	C1MIDL02	R/W	–	–	√	Undefined
F042CH	CAN1 message ID register 02H	C1MIDH02	R/W	–	–	√	Undefined
F042EH	CAN1 message control register 02	C1MCTRL02	R/W	–	–	√	Undefined
F0430H	CAN1 message data byte 01 register 03	C1MDB0103	R/W	–	–	√	Undefined
F0430H	CAN1 message data byte 0 register 03	C1MDB003	R/W	–	√	–	Undefined
F0431H	CAN1 message data byte 1 register 03	C1MDB103	R/W	–	√	–	Undefined
F0432H	CAN1 message data byte 23 register 03	C1MDB2303	R/W	–	–	√	Undefined
F0432H	CAN1 message data byte 2 register 03	C1MDB203	R/W	–	√	–	Undefined
F0433H	CAN1 message data byte 3 register 03	C1MDB303	R/W	–	√	–	Undefined
F0434H	CAN1 message data byte 45 register 03	C1MDB4503	R/W	–	–	√	Undefined
F0434H	CAN1 message data byte 4 register 03	C1MDB403	R/W	–	√	–	Undefined
F0435H	CAN1 message data byte 5 register 03	C1MDB503	R/W	–	√	–	Undefined
F0436H	CAN1 message data byte 67 register 03	C1MDB6703	R/W	–	–	√	Undefined
F0436H	CAN1 message data byte 6 register 03	C1MDB603	R/W	–	√	–	Undefined
F0437H	CAN1 message data byte 7 register 03	C1MDB703	R/W	–	√	–	Undefined
F0438H	CAN1 message data length register 03	C1MDLC03	R/W	–	√	–	0xH
F0439H	CAN1 message Configuration register 03	C1MCONF03	R/W	–	√	–	Undefined
F043AH	CAN1 message ID register 03L	C1MIDL03	R/W	–	–	√	Undefined
F043CH	CAN1 message ID register 03H	C1MIDH03	R/W	–	–	√	Undefined
F043EH	CAN1 message control register 03	C1MCTRL03	R/W	–	–	√	Undefined
F0440H	CAN1 message data byte 01 register 04	C1MDB0104	R/W	–	–	√	Undefined
F0440H	CAN1 message data byte 0 register 04	C1MDB004	R/W	–	√	–	Undefined
F0441H	CAN1 message data byte 1 register 04	C1MDB104	R/W	–	√	–	Undefined
F0442H	CAN1 message data byte 23 register 04	C1MDB2304	R/W	–	–	√	Undefined
F0442H	CAN1 message data byte 2 register 04	C1MDB204	R/W	–	√	–	Undefined
F0443H	CAN1 message data byte 3 register 04	C1MDB304	R/W	–	√	–	Undefined

Table 3-6. Extended SFR (2nd SFR) List (12/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0444H	CAN1 message data byte 45 register 04	C1MDB4504	R/W	–	–	√	Undefined
F0444H	CAN1 message data byte 4 register 04	C1MDB404	R/W	–	√	–	Undefined
F0445H	CAN1 message data byte 5 register 04	C1MDB504	R/W	–	√	–	Undefined
F0446H	CAN1 message data byte 67 register 04	C1MDB6704	R/W	–	–	√	Undefined
F0446H	CAN1 message data byte 6 register 04	C1MDB604	R/W	–	√	–	Undefined
F0447H	CAN1 message data byte 7 register 04	C1MDB704	R/W	–	√	–	Undefined
F0448H	CAN1 message data length register 04	C1MDLC04	R/W	–	√	–	0xH
F0449H	CAN1 message Configuration register 04	C1MCONF04	R/W	–	√	–	Undefined
F044AH	CAN1 message ID register 04L	C1MIDL04	R/W	–	–	√	Undefined
F044CH	CAN1 message ID register 04H	C1MIDH04	R/W	–	–	√	Undefined
F044EH	CAN1 message control register 04	C1MCTRL04	R/W	–	–	√	Undefined
F0450H	CAN1 message data byte 01 register 05	C1MDB0105	R/W	–	–	√	Undefined
F0450H	CAN1 message data byte 0 register 05	C1MDB005	R/W	–	√	–	Undefined
F0451H	CAN1 message data byte 1 register 05	C1MDB105	R/W	–	√	–	Undefined
F0452H	CAN1 message data byte 23 register 05	C1MDB2305	R/W	–	–	√	Undefined
F0452H	CAN1 message data byte 2 register 05	C1MDB205	R/W	–	√	–	Undefined
F0453H	CAN1 message data byte 3 register 05	C1MDB305	R/W	–	√	–	Undefined
F0454H	CAN1 message data byte 45 register 05	C1MDB4505	R/W	–	–	√	Undefined
F0454H	CAN1 message data byte 4 register 05	C1MDB405	R/W	–	√	–	Undefined
F0455H	CAN1 message data byte 5 register 05	C1MDB505	R/W	–	√	–	Undefined
F0456H	CAN1 message data byte 67 register 05	C1MDB6705	R/W	–	–	√	Undefined
F0456H	CAN1 message data byte 6 register 05	C1MDB605	R/W	–	√	–	Undefined
F0457H	CAN1 message data byte 7 register 05	C1MDB705	R/W	–	√	–	Undefined
F0458H	CAN1 message data length register 05	C1MDLC05	R/W	–	√	–	0xH
F0459H	CAN1 message Configuration register 05	C1MCONF05	R/W	–	√	–	Undefined
F045AH	CAN1 message ID register 05L	C1MIDL05	R/W	–	–	√	Undefined
F045CH	CAN1 message ID register 05H	C1MIDH05	R/W	–	–	√	Undefined
F045EH	CAN1 message control register 05	C1MCTRL05	R/W	–	–	√	Undefined
F0460H	CAN1 message data byte 01 register 06	C1MDB0106	R/W	–	–	√	Undefined
F0460H	CAN1 message data byte 0 register 06	C1MDB006	R/W	–	√	–	Undefined
F0461H	CAN1 message data byte 1 register 06	C1MDB106	R/W	–	√	–	Undefined
F0462H	CAN1 message data byte 23 register 06	C1MDB2306	R/W	–	–	√	Undefined
F0462H	CAN1 message data byte 2 register 06	C1MDB206	R/W	–	√	–	Undefined
F0463H	CAN1 message data byte 3 register 06	C1MDB306	R/W	–	√	–	Undefined
F0464H	CAN1 message data byte 45 register 06	C1MDB4506	R/W	–	–	√	Undefined
F0464H	CAN1 message data byte 4 register 06	C1MDB406	R/W	–	√	–	Undefined
F0465H	CAN1 message data byte 5 register 06	C1MDB506	R/W	–	√	–	Undefined
F0466H	CAN1 message data byte 67 register 06	C1MDB6706	R/W	–	–	√	Undefined
F0466H	CAN1 message data byte 6 register 06	C1MDB606	R/W	–	√	–	Undefined
F0467H	CAN1 message data byte 7 register 06	C1MDB706	R/W	–	√	–	Undefined
F0468H	CAN1 message data length register 06	C1MDLC06	R/W	–	√	–	0xH
F0469H	CAN1 message Configuration register 06	C1MCONF06	R/W	–	√	–	Undefined

Table 3-6. Extended SFR (2nd SFR) List (13/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F046AH	CAN1 message ID register 06L	C1MIDL06	R/W	–	–	√	Undefined
F046CH	CAN1 message ID register 06H	C1MIDH06	R/W	–	–	√	Undefined
F046EH	CAN1 message control register 06	C1MCTRL06	R/W	–	–	√	Undefined
F0470H	CAN1 message data byte 01 register 07	C1MDB0107	R/W	–	–	√	Undefined
F0470H	CAN1 message data byte 0 register 07	C1MDB007	R/W	–	√	–	Undefined
F0471H	CAN1 message data byte 1 register 07	C1MDB107	R/W	–	√	–	Undefined
F0472H	CAN1 message data byte 23 register 07	C1MDB2307	R/W	–	–	√	Undefined
F0472H	CAN1 message data byte 2 register 07	C1MDB207	R/W	–	√	–	Undefined
F0473H	CAN1 message data byte 3 register 07	C1MDB307	R/W	–	√	–	Undefined
F0474H	CAN1 message data byte 45 register 07	C1MDB4507	R/W	–	–	√	Undefined
F0474H	CAN1 message data byte 4 register 07	C1MDB407	R/W	–	√	–	Undefined
F0475H	CAN1 message data byte 5 register 07	C1MDB507	R/W	–	√	–	Undefined
F0476H	CAN1 message data byte 67 register 07	C1MDB6707	R/W	–	–	√	Undefined
F0476H	CAN1 message data byte 6 register 07	C1MDB607	R/W	–	√	–	Undefined
F0477H	CAN1 message data byte 7 register 07	C1MDB707	R/W	–	√	–	Undefined
F0478H	CAN1 message data length register 07	C1MDLC07	R/W	–	√	–	0xH
F0479H	CAN1 message Configuration register 07	C1MCONF07	R/W	–	√	–	Undefined
F047AH	CAN1 message ID register 07L	C1MIDL07	R/W	–	–	√	Undefined
F047CH	CAN1 message ID register 07H	C1MIDH07	R/W	–	–	√	Undefined
F047EH	CAN1 message control register 07	C1MCTRL07	R/W	–	–	√	Undefined
F0480H	CAN1 message data byte 01 register 08	C1MDB0108	R/W	–	–	√	Undefined
F0480H	CAN1 message data byte 0 register 08	C1MDB008	R/W	–	√	–	Undefined
F0481H	CAN1 message data byte 1 register 08	C1MDB108	R/W	–	√	–	Undefined
F0482H	CAN1 message data byte 23 register 08	C1MDB2308	R/W	–	–	√	Undefined
F0482H	CAN1 message data byte 2 register 08	C1MDB208	R/W	–	√	–	Undefined
F0483H	CAN1 message data byte 3 register 08	C1MDB308	R/W	–	√	–	Undefined
F0484H	CAN1 message data byte 45 register 08	C1MDB4508	R/W	–	–	√	Undefined
F0484H	CAN1 message data byte 4 register 08	C1MDB408	R/W	–	√	–	Undefined
F0485H	CAN1 message data byte 5 register 08	C1MDB508	R/W	–	√	–	Undefined
F0486H	CAN1 message data byte 67 register 08	C1MDB6708	R/W	–	–	√	Undefined
F0486H	CAN1 message data byte 6 register 08	C1MDB608	R/W	–	√	–	Undefined
F0487H	CAN1 message data byte 7 register 08	C1MDB708	R/W	–	√	–	Undefined
F0488H	CAN1 message data length register 08	C1MDLC08	R/W	–	√	–	0xH
F0489H	CAN1 message Configuration register 08	C1MCONF08	R/W	–	√	–	Undefined
F048AH	CAN1 message ID register 08L	C1MIDL08	R/W	–	–	√	Undefined
F048CH	CAN1 message ID register 08H	C1MIDH08	R/W	–	–	√	Undefined
F048EH	CAN1 message control register 08	C1MCTRL08	R/W	–	–	√	Undefined
F0490H	CAN1 message data byte 01 register 09	C1MDB0109	R/W	–	–	√	Undefined
F0490H	CAN1 message data byte 0 register 09	C1MDB009	R/W	–	√	–	Undefined
F0491H	CAN1 message data byte 1 register 09	C1MDB109	R/W	–	√	–	Undefined

Table 3-6. Extended SFR (2nd SFR) List (14/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0492H	CAN1 message data byte 23 register 09	C1MDB2309	R/W	–	–	√	Undefined
F0492H	CAN1 message data byte 2 register 09	C1MDB209	R/W	–	√	–	Undefined
F0493H	CAN1 message data byte 3 register 09	C1MDB309	R/W	–	√	–	Undefined
F0494H	CAN1 message data byte 45 register 09	C1MDB4509	R/W	–	–	√	Undefined
F0494H	CAN1 message data byte 4 register 09	C1MDB409	R/W	–	√	–	Undefined
F0495H	CAN1 message data byte 5 register 09	C1MDB509	R/W	–	√	–	Undefined
F0496H	CAN1 message data byte 67 register 09	C1MDB6709	R/W	–	–	√	Undefined
F0496H	CAN1 message data byte 6 register 09	C1MDB609	R/W	–	√	–	Undefined
F0497H	CAN1 message data byte 7 register 09	C1MDB709	R/W	–	√	–	Undefined
F0498H	CAN1 message data length register 09	C1MDLC09	R/W	–	√	–	0xH
F0499H	CAN1 message Configuration register 09	C1MCONF09	R/W	–	√	–	Undefined
F049AH	CAN1 message ID register 09L	C1MIDL09	R/W	–	–	√	Undefined
F049CH	CAN1 message ID register 09H	C1MIDH09	R/W	–	–	√	Undefined
F049EH	CAN1 message control register 09	C1MCTRL09	R/W	–	–	√	Undefined
F04A0H	CAN1 message data byte 01 register 10	C1MDB0110	R/W	–	–	√	Undefined
F04A0H	CAN1 message data byte 0 register 10	C1MDB010	R/W	–	√	–	Undefined
F04A1H	CAN1 message data byte 1 register 10	C1MDB110	R/W	–	√	–	Undefined
F04A2H	CAN1 message data byte 23 register 10	C1MDB2310	R/W	–	–	√	Undefined
F04A2H	CAN1 message data byte 2 register 10	C1MDB210	R/W	–	√	–	Undefined
F04A3H	CAN1 message data byte 3 register 10	C1MDB310	R/W	–	√	–	Undefined
F04A4H	CAN1 message data byte 45 register 10	C1MDB4510	R/W	–	–	√	Undefined
F04A4H	CAN1 message data byte 4 register 10	C1MDB410	R/W	–	√	–	Undefined
F04A5H	CAN1 message data byte 5 register 10	C1MDB510	R/W	–	√	–	Undefined
F04A6H	CAN1 message data byte 67 register 10	C1MDB6710	R/W	–	–	√	Undefined
F04A6H	CAN1 message data byte 6 register 10	C1MDB610	R/W	–	√	–	Undefined
F04A7H	CAN1 message data byte 7 register 10	C1MDB710	R/W	–	√	–	Undefined
F04A8H	CAN1 message data length register 10	C1MDLC10	R/W	–	√	–	0xH
F04A9H	CAN1 message Configuration register 10	C1MCONF10	R/W	–	√	–	Undefined
F04AAH	CAN1 message ID register 10L	C1MIDL10	R/W	–	–	√	Undefined
F04ACH	CAN1 message ID register 10H	C1MIDH10	R/W	–	–	√	Undefined
F04AEH	CAN1 message control register 10	C1MCTRL10	R/W	–	–	√	Undefined
F04B0H	CAN1 message data byte 01 register 11	C1MDB0111	R/W	–	–	√	Undefined
F04B0H	CAN1 message data byte 0 register 11	C1MDB011	R/W	–	√	–	Undefined
F04B1H	CAN1 message data byte 1 register 11	C1MDB111	R/W	–	√	–	Undefined
F04B2H	CAN1 message data byte 23 register 11	C1MDB2311	R/W	–	–	√	Undefined
F04B2H	CAN1 message data byte 2 register 11	C1MDB211	R/W	–	√	–	Undefined
F04B3H	CAN1 message data byte 3 register 11	C1MDB311	R/W	–	√	–	Undefined
F04B4H	CAN1 message data byte 45 register 11	C1MDB4511	R/W	–	–	√	Undefined
F04B4H	CAN1 message data byte 4 register 11	C1MDB411	R/W	–	√	–	Undefined
F04B5H	CAN1 message data byte 5 register 11	C1MDB511	R/W	–	√	–	Undefined

Table 3-6. Extended SFR (2nd SFR) List (15/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F04B6H	CAN1 message data byte 67 register 11	C1MDB6711	R/W	–	–	√	Undefined
F04B6H	CAN1 message data byte 6 register 11	C1MDB611	R/W	–	√	–	Undefined
F04B7H	CAN1 message data byte 7 register 11	C1MDB711	R/W	–	√	–	Undefined
F04B8H	CAN1 message data length register 11	C1MDLC11	R/W	–	√	–	0xH
F04B9H	CAN1 message Configuration register 11	C1MCONF11	R/W	–	√	–	Undefined
F04BAH	CAN1 message ID register 11L	C1MIDL11	R/W	–	–	√	Undefined
F04BCH	CAN1 message ID register 11H	C1MIDH11	R/W	–	–	√	Undefined
F04BEH	CAN1 message control register 11	C1MCTRL11	R/W	–	–	√	Undefined
F04C0H	CAN1 message data byte 01 register 12	C1MDB0112	R/W	–	–	√	Undefined
F04C0H	CAN1 message data byte 0 register 12	C1MDB012	R/W	–	√	–	Undefined
F04C1H	CAN1 message data byte 1 register 12	C1MDB112	R/W	–	√	–	Undefined
F04C2H	CAN1 message data byte 23 register 12	C1MDB2312	R/W	–	–	√	Undefined
F04C2H	CAN1 message data byte 2 register 12	C1MDB212	R/W	–	√	–	Undefined
F04C3H	CAN1 message data byte 3 register 12	C1MDB312	R/W	–	√	–	Undefined
F04C4H	CAN1 message data byte 45 register 12	C1MDB4512	R/W	–	–	√	Undefined
F04C4H	CAN1 message data byte 4 register 12	C1MDB412	R/W	–	√	–	Undefined
F04C5H	CAN1 message data byte 5 register 12	C1MDB512	R/W	–	√	–	Undefined
F04C6H	CAN1 message data byte 67 register 12	C1MDB6712	R/W	–	–	√	Undefined
F04C6H	CAN1 message data byte 6 register 12	C1MDB612	R/W	–	√	–	Undefined
F04C7H	CAN1 message data byte 7 register 12	C1MDB712	R/W	–	√	–	Undefined
F04C8H	CAN1 message data length register 12	C1MDLC12	R/W	–	√	–	0xH
F04C9H	CAN1 message Configuration register 12	C1MCONF12	R/W	–	√	–	Undefined
F04CAH	CAN1 message ID register 12L	C1MIDL12	R/W	–	–	√	Undefined
F04CCH	CAN1 message ID register 12H	C1MIDH12	R/W	–	–	√	Undefined
F04CEH	CAN1 message control register 12	C1MCTRL12	R/W	–	–	√	Undefined
F04D0H	CAN1 message data byte 01 register 13	C1MDB0113	R/W	–	–	√	Undefined
F04D0H	CAN1 message data byte 0 register 13	C1MDB013	R/W	–	√	–	Undefined
F04D1H	CAN1 message data byte 1 register 13	C1MDB113	R/W	–	√	–	Undefined
F04D2H	CAN1 message data byte 23 register 13	C1MDB2313	R/W	–	–	√	Undefined
F04D2H	CAN1 message data byte 2 register 13	C1MDB213	R/W	–	√	–	Undefined
F04D3H	CAN1 message data byte 3 register 13	C1MDB313	R/W	–	√	–	Undefined
F04D4H	CAN1 message data byte 45 register 13	C1MDB4513	R/W	–	–	√	Undefined
F04D4H	CAN1 message data byte 4 register 13	C1MDB413	R/W	–	√	–	Undefined
F04D5H	CAN1 message data byte 5 register 13	C1MDB513	R/W	–	√	–	Undefined
F04D6H	CAN1 message data byte 67 register 13	C1MDB6713	R/W	–	–	√	Undefined
F04D6H	CAN1 message data byte 6 register 13	C1MDB613	R/W	–	√	–	Undefined
F04D7H	CAN1 message data byte 7 register 13	C1MDB713	R/W	–	√	–	Undefined
F04D8H	CAN1 message data length register 13	C1MDLC13	R/W	–	√	–	0xH
F04D9H	CAN1 message Configuration register 13	C1MCONF13	R/W	–	√	–	Undefined
F04DAH	CAN1 message ID register 13L	C1MIDL13	R/W	–	–	√	Undefined
F04DCH	CAN1 message ID register 13H	C1MIDH13	R/W	–	–	√	Undefined
F04DEH	CAN1 message control register 13	C1MCTRL13	R/W	–	–	√	Undefined

Table 3-6. Extended SFR (2nd SFR) List (16/21)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F04E0H	CAN1 message data byte 01 register 14	C1MDB0114	R/W	–	–	√	Undefined
F04E0H	CAN1 message data byte 0 register 14	C1MDB014	R/W	–	√	–	Undefined
F04E1H	CAN1 message data byte 1 register 14	C1MDB114	R/W	–	√	–	Undefined
F04E2H	CAN1 message data byte 23 register 14	C1MDB2314	R/W	–	–	√	Undefined
F04E2H	CAN1 message data byte 2 register 14	C1MDB214	R/W	–	√	–	Undefined
F04E3H	CAN1 message data byte 3 register 14	C1MDB314	R/W	–	√	–	Undefined
F04E4H	CAN1 message data byte 45 register 14	C1MDB4514	R/W	–	–	√	Undefined
F04E4H	CAN1 message data byte 4 register 14	C1MDB414	R/W	–	√	–	Undefined
F04E5H	CAN1 message data byte 5 register 14	C1MDB514	R/W	–	√	–	Undefined
F04E6H	CAN1 message data byte 67 register 14	C1MDB6714	R/W	–	–	√	Undefined
F04E6H	CAN1 message data byte 6 register 14	C1MDB614	R/W	–	√	–	Undefined
F04E7H	CAN1 message data byte 7 register 14	C1MDB714	R/W	–	√	–	Undefined
F04E8H	CAN1 message data length register 14	C1MDLC14	R/W	–	√	–	0xH
F04E9H	CAN1 message Configuration register 14	C1MCONF14	R/W	–	√	–	Undefined
F04EAH	CAN1 message ID register 14L	C1MIDL14	R/W	–	–	√	Undefined
F04ECH	CAN1 message ID register 14H	C1MIDH14	R/W	–	–	√	Undefined
F04EEH	CAN1 message control register 14	C1MCTRL14	R/W	–	–	√	Undefined
F04F0H	CAN1 message data byte 01 register 15	C1MDB0115	R/W	–	–	√	Undefined
F04F0H	CAN1 message data byte 0 register 15	C1MDB015	R/W	–	√	–	Undefined
F04F1H	CAN1 message data byte 1 register 15	C1MDB115	R/W	–	√	–	Undefined
F04F2H	CAN1 message data byte 23 register 15	C1MDB2315	R/W	–	–	√	Undefined
F04F2H	CAN1 message data byte 2 register 15	C1MDB215	R/W	–	√	–	Undefined
F04F3H	CAN1 message data byte 3 register 15	C1MDB315	R/W	–	√	–	Undefined
F04F4H	CAN1 message data byte 45 register 15	C1MDB4515	R/W	–	–	√	Undefined
F04F4H	CAN1 message data byte 4 register 15	C1MDB415	R/W	–	√	–	Undefined
F04F5H	CAN1 message data byte 5 register 15	C1MDB515	R/W	–	√	–	Undefined
F04F6H	CAN1 message data byte 67 register 15	C1MDB6715	R/W	–	–	√	Undefined
F04F6H	CAN1 message data byte 6 register 15	C1MDB615	R/W	–	√	–	Undefined
F04F7H	CAN1 message data byte 7 register 15	C1MDB715	R/W	–	√	–	Undefined
F04F8H	CAN1 message data length register 15	C1MDLC15	R/W	–	√	–	0xH
F04F9H	CAN1 message Configuration register 15	C1MCONF15	R/W	–	√	–	Undefined
F04FAH	CAN1 message ID register 15L	C1MIDL15	R/W	–	–	√	Undefined
F04FCH	CAN1 message ID register 15H	C1MIDH15	R/W	–	–	√	Undefined
F04FEH	CAN1 message control register 15	C1MCTRL15	R/W	–	–	√	Undefined
F05C0H	CAN0 global module control register	C0GMCTRL	R/W	–	–	√	0000H
F05C6H	CAN0 global block transmission control register	C0GMABT	R/W	–	–	√	0000H
F05C8H	CAN0 global block transmission delay setting register	C0GMABTD	R/W	–	√	–	00H
F05CEH	CAN0 global module clock select register	C0GMCS	R/W	–	√	–	0FH

Table 3-6. Extended SFR (2nd SFR) List (17/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F05D0H	CAN0 module mask 1 register L	C0MASK1L		R/W	–	–	√	Undefined
F05D2H	CAN0 module mask 1 register H	C0MASK1H		R/W	–	–	√	Undefined
F05D4H	CAN0 module mask 2 register L	C0MASK2L		R/W	–	–	√	Undefined
F05D6H	CAN0 module mask 2 register H	C0MASK2H		R/W	–	–	√	Undefined
F05D8H	CAN0 module mask 3 register L	C0MASK3L		R/W	–	–	√	Undefined
F05DAH	CAN0 module mask 3 register H	C0MASK3H		R/W	–	–	√	Undefined
F05DCH	CAN0 module mask 4 register L	C0MASK4L		R/W	–	–	√	Undefined
F05DEH	CAN0 module mask 4 register H	C0MASK4H		R/W	–	–	√	Undefined
F05E0H	CAN0 module control register	C0CTRL		R/W	–	–	√	0000H
F05E2H	CAN0 module last error information register	C0LEC		R/W	–	√	–	00H
F05E3H	CAN0 module information register	C0INFO		R	–	√	–	00H
F05E4H	CAN0 module error counter register	C0ERC		R	–	–	√	0000H
F05E6H	CAN0 module interrupt enable register	C0IE		R/W	–	–	√	0000H
F05E8H	CAN0 module interrupt status register	C0INTS		R/W	–	–	√	0000H
F05EAH	CAN0 module bit rate prescaler register	C0BRP		R/W	–	√	–	FFH
F05ECH	CAN0 module bit rate register	C0BTR		R/W	–	–	√	370FH
F05EEH	CAN0 module last in-pointer register	C0LIPT		R	–	√	–	Undefined
F05F0H	CAN0 module receive history list register	C0RGPT		R/W	–	–	√	xx02H
F05F2H	CAN0 module last out-pointer register	C0LOPT		R	–	√	–	Undefined
F05F4H	CAN0 module transmit history list register	C0TGPT		R/W	–	–	√	xx2H
F05F6H	CAN0 module time stamp register	C0TS		R/W	–	–	√	0000H
F0600H	CAN0 message data byte 01 register 00	C0MDB000	C0MDB0100	R/W	–	√	√	Undefined
F0601H		C0MDB100						
F0602H	CAN0 message data byte 23 register 00	C0MDB200	C0MDB2300	R/W	–	√	√	Undefined
F0603H		C0MDB300						
F0604H	CAN0 message data Byte 45 register 00	C0MDB400	C0MDB4500	R/W	–	√	√	Undefined
F0605H		C0MDB500						
F0606H	CAN0 message data byte 67 register 00	C0MDB600	C0MDB6700	R/W	–	√	√	Undefined
F0607H		C0MDB700						
F0608H	CAN0 message data length register 00	C0MDLC00		R/W	–	√	–	0xH
F0609H	CAN0 message configuration register 00	C0MCONF00		R/W	–	√	–	Undefined
F060AH	CAN0 message ID register 00L	C0MIDL00		R/W	–	–	√	Undefined
F060CH	CAN0 message ID register 00H	C0MIDH00		R/W	–	–	√	Undefined
F060EH	CAN0 message control register 00	C0MCTRL00		R/W	–	–	√	Undefined
F0610H	CAN0 message data byte 01 register 01	C0MDB001	C0MDB0101	R/W	–	√	√	Undefined
F0611H		C0MDB101						
F0612H	CAN0 message data byte 23 register 01	C0MDB201	C0MDB2301	R/W	–	√	√	Undefined
F0613H		C0MDB301						
F0614H	CAN0 message data byte 45 register 01	C0MDB401	C0MDB4501	R/W	–	√	√	Undefined
F0615H		C0MDB501						
F0616H	CAN0 message data byte 67 register 01	C0MDB601	C0MDB6701	R/W	–	√	√	Undefined
F0617H		C0MDB701						
F0618H	CAN0 message data length register 01	C0MDLC01		R/W	–	√	–	0xH
F0619H	CAN0 message configuration register 01	C0MCONF01		R/W	–	√	–	Undefined
F061AH	CAN0 message ID register 01L	C0MIDL01		R/W	–	–	√	Undefined
F061CH	CAN0 message ID register 01H	C0MIDH01		R/W	–	–	√	Undefined

Table 3-6. Extended SFR (2nd SFR) List (18/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F061EH	CAN0 message control register 01	C0MCTRL01		R/W	–	–	√	Undefined
F0620H	CAN0 message data byte 01 register 02	C0MDB002	C0MDB0102	R/W	–	√	√	Undefined
F0621H		C0MDB102						
F0622H	CAN0 message data byte 23 register 02	C0MDB202	C0MDB2302	R/W	–	√	√	Undefined
F0623H		C0MDB302						
F0624H	CAN0 message data byte 45 register 02	C0MDB402	C0MDB4502	R/W	–	√	√	Undefined
F0625H		C0MDB502						
F0626H	CAN0 message data byte 67 register 02	C0MDB602	C0MDB6702	R/W	–	√	√	Undefined
F0627H		C0MDB702						
F0628H	CAN0 message data length register 02	C0MDLC02		R/W	–	√	–	0xH
F0629H	CAN0 message configuration register 02	C0MCONF02		R/W	–	√	–	Undefined
F062AH	CAN0 message ID register 02L	C0MIDL02		R/W	–	–	√	Undefined
F062CH	CAN0 message ID register 02H	C0MIDH02		R/W	–	–	√	Undefined
F062EH	CAN0 message control register 02	C0MCTRL02		R/W	–	–	√	Undefined
F0630H	CAN0 message data byte 01 register 03	C0MDB003	C0MDB0103	R/W	–	√	√	Undefined
F0631H		C0MDB103						
F0632H	CAN0 message data byte 23 register 03	C0MDB203	C0MDB2303	R/W	–	√	√	Undefined
F0633H		C0MDB303						
F0634H	CAN0 message data byte 45 register 03	C0MDB403	C0MDB4503	R/W	–	√	√	Undefined
F0635H		C0MDB503						
F0636H	CAN0 message data byte 67 register 03	C0MDB603	C0MDB6703	R/W	–	√	√	Undefined
F0637H		C0MDB703						
F0638H	CAN0 message data length register 03	C0MDLC03		R/W	–	√	–	0xH
F0639H	CAN0 message Configuration register 03	C0MCONF03		R/W	–	√	–	Undefined
F063AH	CAN0 message ID register 03L	C0MIDL03		R/W	–	–	√	Undefined
F063CH	CAN0 message ID register 03H	C0MIDH03		R/W	–	–	√	Undefined
F063EH	CAN0 message control register 03	C0MCTRL03		R/W	–	–	√	Undefined
F0640H	CAN0 message data byte 01 register 04	C0MDB004	C0MDB0104	R/W	–	√	√	Undefined
F0641H		C0MDB104						
F0642H	CAN0 message data byte 23 register 04	C0MDB204	C0MDB2304	R/W	–	√	√	Undefined
F0643H		C0MDB304						
F0644H	CAN0 message data byte 45 register 04	C0MDB404	C0MDB4504	R/W	–	√	√	Undefined
F0645H		C0MDB504						
F0646H	CAN0 message data byte 67 register 04	C0MDB604	C0MDB6704	R/W	–	√	√	Undefined
F0647H		C0MDB704						
F0648H	CAN0 message data length register 04	C0MDLC04		R/W	–	√	–	0xH
F0649H	CAN0 message Configuration register 04	C0MCONF04		R/W	–	√	–	Undefined
F064AH	CAN0 message ID register 04L	C0MIDL04		R/W	–	–	√	Undefined
F064CH	CAN0 message ID register 04H	C0MIDH04		R/W	–	–	√	Undefined
F064EH	CAN0 message control register 04	C0MCTRL04		R/W	–	–	√	Undefined
F0650H	CAN0 message data byte 01 register 05	C0MDB005	C0MDB0105	R/W	–	√	√	Undefined
F0651H		C0MDB105						
F0652H	CAN0 message data byte 23 register 05	C0MDB205	C0MDB2305	R/W	–	√	√	Undefined
F0653H		C0MDB305						

Table 3-6. Extended SFR (2nd SFR) List (19/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0654H	CAN0 message data byte 45 register 05	C0MDB405	C0MDB4505	R/W	–	√	√	Undefined
F0655H		C0MDB505						
F0656H	CAN0 message data byte 67 register 05	C0MDB605	C0MDB6705	R/W	–	√	√	Undefined
F0657H		C0MDB705						
F0658H	CAN0 message data length register 05	C0MDLC05		R/W	–	√	–	0xH
F0659H	CAN0 message Configuration register 05	C0MCONF05		R/W	–	√	–	Undefined
F065AH	CAN0 message ID register 05L	C0MIDL05		R/W	–	–	√	Undefined
F065CH	CAN0 message ID register 05H	C0MIDH05		R/W	–	–	√	Undefined
F065EH	CAN0 message control register 05	C0MCTRL05		R/W	–	–	√	Undefined
F0660H	CAN0 message data byte 01 register 06	C0MDB006	C0MDB0106	R/W	–	√	√	Undefined
F0661H		C0MDB106						
F0662H	CAN0 message data byte 23 register 06	C0MDB206	C0MDB2306	R/W	–	√	√	Undefined
F0663H		C0MDB306						
F0664H	CAN0 message data byte 45 register 06	C0MDB406	C0MDB4506	R/W	–	√	√	Undefined
F0665H		C0MDB506						
F0666H	CAN0 message data byte 67 register 06	C0MDB606	C0MDB6706	R/W	–	√	√	Undefined
F0667H		C0MDB706						
F0668H	CAN0 message data length register 06	C0MDLC06		R/W	–	√	–	0xH
F0669H	CAN0 message Configuration register 06	C0MCONF06		R/W	–	√	–	Undefined
F066AH	CAN0 message ID register 06L	C0MIDL06		R/W	–	–	√	Undefined
F066CH	CAN0 message ID register 06H	C0MIDH06		R/W	–	–	√	Undefined
F066EH	CAN0 message control register 06	C0MCTRL06		R/W	–	–	√	Undefined
F0670H	CAN0 message data byte 01 register 07	C0MDB007	C0MDB0107	R/W	–	√	√	Undefined
F0671H		C0MDB107						
F0672H	CAN0 message data byte 23 register 07	C0MDB207	C0MDB2307	R/W	–	√	√	Undefined
F0673H		C0MDB307						
F0674H	CAN0 message data byte 45 register 07	C0MDB407	C0MDB4507	R/W	–	√	√	Undefined
F0675H		C0MDB507						
F0676H	CAN0 message data byte 67 register 07	C0MDB607	C0MDB6707	R/W	–	√	√	Undefined
F0677H		C0MDB707						
F0678H	CAN0 message data length register 07	C0MDLC07		R/W	–	√	–	0xH
F0679H	CAN0 message Configuration register 07	C0MCONF07		R/W	–	√	–	Undefined
F067AH	CAN0 message ID register 07L	C0MIDL07		R/W	–	–	√	Undefined
F067CH	CAN0 message ID register 07H	C0MIDH07		R/W	–	–	√	Undefined
F067EH	CAN0 message control register 07	C0MCTRL07		R/W	–	–	√	Undefined
F0680H	CAN0 message data byte 01 register 08	C0MDB008	C0MDB0108	R/W	–	√	√	Undefined
F0681H		C0MDB108						
F0682H	CAN0 message data byte 23 register 08	C0MDB208	C0MDB2308	R/W	–	√	√	Undefined
F0683H		C0MDB308						
F0684H	CAN0 message data byte 45 register 08	C0MDB408	C0MDB4508	R/W	–	√	√	Undefined
F0685H		C0MDB508						
F0686H	CAN0 message data byte 67 register 08	C0MDB608	C0MDB6708	R/W	–	√	√	Undefined
F0687H		C0MDB708						
F0688H	CAN0 message data length register 08	C0MDLC08		R/W	–	√	–	0xH
F0689H	CAN0 message Configuration register 08	C0MCONF08		R/W	–	√	–	Undefined
F068AH	CAN0 message ID register 08L	C0MIDL08		R/W	–	–	√	Undefined
F068CH	CAN0 message ID register 08H	C0MIDH08		R/W	–	–	√	Undefined

Table 3-6. Extended SFR (2nd SFR) List (20/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F068EH	CAN0 message control register 08	C0MCTRL08		R/W	–	–	√	Undefined
F0690H	CAN0 message data byte 01 register 09	C0MDB009	C0MDB0109	R/W	–	√	√	Undefined
F0691H		C0MDB109						
F0692H	CAN0 message data byte 23 register 09	C0MDB209	C0MDB2309	R/W	–	√	√	Undefined
F0693H		C0MDB309						
F0694H	CAN0 message data byte 45 register 09	C0MDB409	C0MDB4509	R/W	–	√	√	Undefined
F0695H		C0MDB509						
F0696H	CAN0 message data byte 67 register 09	C0MDB609	C0MDB6709	R/W	–	√	√	Undefined
F0697H		C0MDB709						
F0698H	CAN0 message data length register 09	C0MDLC09		R/W	–	√	–	0xH
F0699H	CAN0 message Configuration register 09	C0MCONF09		R/W	–	√	–	Undefined
F069AH	CAN0 message ID register 09L	C0MIDL09		R/W	–	–	√	Undefined
F069CH	CAN0 message ID register 09H	C0MIDH09		R/W	–	–	√	Undefined
F069EH	CAN0 message control register 09	C0MCTRL09		R/W	–	–	√	Undefined
F06A0H	CAN0 message data byte 01 register 10	C0MDB010	C0MDB0110	R/W	–	√	√	Undefined
F06A1H		C0MDB110						
F06A2H	CAN0 message data byte 23 register 10	C0MDB210	C0MDB2310	R/W	–	√	√	Undefined
F06A3H		C0MDB310						
F06A4H	CAN0 message data byte 45 register 10	C0MDB410	C0MDB4510	R/W	–	√	√	Undefined
F06A5H		C0MDB510						
F06A6H	CAN0 message data byte 67 register 10	C0MDB610	C0MDB6710	R/W	–	√	√	Undefined
F06A7H		C0MDB710						
F06A8H	CAN0 message data length register 10	C0MDLC10		R/W	–	√	–	0xH
F06A9H	CAN0 message Configuration register 10	C0MCONF10		R/W	–	√	–	Undefined
F06AAH	CAN0 message ID register 10L	C0MIDL10		R/W	–	–	√	Undefined
F06ACH	CAN0 message ID register 10H	C0MIDH10		R/W	–	–	√	Undefined
F06AEH	CAN0 message control register 10	C0MCTRL10		R/W	–	–	√	Undefined
F06B0H	CAN0 message data byte 01 register 11	C0MDB011	C0MDB0111	R/W	–	√	√	Undefined
F06B1H		C0MDB111						
F06B2H	CAN0 message data byte 23 register 11	C0MDB211	C0MDB2311	R/W	–	√	√	Undefined
F06B3H		C0MDB311						
F06B4H	CAN0 message data byte 45 register 11	C0MDB411	C0MDB4511	R/W	–	√	√	Undefined
F06B5H		C0MDB511						
F06B6H	CAN0 message data byte 67 register 11	C0MDB611	C0MDB6711	R/W	–	√	√	Undefined
F06B7H		C0MDB711						
F06B8H	CAN0 message data length register 11	C0MDLC11		R/W	–	√	–	0xH
F06B9H	CAN0 message Configuration register 11	C0MCONF11		R/W	–	√	–	Undefined
F06BAH	CAN0 message ID register 11L	C0MIDL11		R/W	–	–	√	Undefined
F06BCH	CAN0 message ID register 11H	C0MIDH11		R/W	–	–	√	Undefined
F06BEH	CAN0 message control register 11	C0MCTRL11		R/W	–	–	√	Undefined
F06C0H	CAN0 message data byte 01 register 12	C0MDB012	C0MDB0112	R/W	–	√	√	Undefined
F06C1H		C0MDB112						
F06C2H	CAN0 message data byte 23 register 12	C0MDB212	C0MDB2312	R/W	–	√	√	Undefined
F06C3H		C0MDB312						
F06C4H	CAN0 message data byte 45 register 12	C0MDB412	C0MDB4512	R/W	–	√	√	Undefined
F06C5H		C0MDB512						

Table 3-6. Extended SFR (2nd SFR) List (21/21)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F06C6H	CAN0 message data byte 67 register 12	C0MDB612	C0MDB6712	R/W	–	√	√	Undefined
F06C7H		C0MDB712						
F06C8H	CAN0 message data length register 12	C0MDLC12		R/W	–	√	–	0XH
F06C9H	CAN0 message Configuration register 12	C0MCONF12		R/W	–	√	–	Undefined
F06CAH	CAN0 message ID register 12L	C0MIDL12		R/W	–	–	√	Undefined
F06CCH	CAN0 message ID register 12H	C0MIDH12		R/W	–	–	√	Undefined
F06CEH	CAN0 message control register 12	C0MCTRL12		R/W	–	–	√	Undefined
F06D0H	CAN0 message data byte 01 register 13	C0MDB013	C0MDB0113	R/W	–	√	√	Undefined
F06D1H		C0MDB113						
F06D2H	CAN0 message data byte 23 register 13	C0MDB213	C0MDB2313	R/W	–	√	√	Undefined
F06D3H		C0MDB313						
F06D4H	CAN0 message data byte 45 register 13	C0MDB413	C0MDB4513	R/W	–	√	√	Undefined
F06D5H		C0MDB513						
F06D6H	CAN0 message data byte 67 register 13	C0MDB613	C0MDB6713	R/W	–	√	√	Undefined
F06D7H		C0MDB713						
F06D8H	CAN0 message data length register 13	C0MDLC13		R/W	–	√	–	0XH
F06D9H	CAN0 message Configuration register 13	C0MCONF13		R/W	–	√	–	Undefined
F06DAH	CAN0 message ID register 13L	C0MIDL13		R/W	–	–	√	Undefined
F06DCH	CAN0 message ID register 13H	C0MIDH13		R/W	–	–	√	Undefined
F06DEH	CAN0 message control register 13	C0MCTRL13		R/W	–	–	√	Undefined
F06E0H	CAN0 message data byte 01 register 14	C0MDB014	C0MDB0114	R/W	–	√	√	Undefined
F06E1H		C0MDB114						
F06E2H	CAN0 message data byte 23 register 14	C0MDB214	C0MDB2314	R/W	–	√	√	Undefined
F06E3H		C0MDB314						
F06E4H	CAN0 message data byte 45 register 14	C0MDB414	C0MDB4514	R/W	–	√	√	Undefined
F06E5H		C0MDB514						
F06E6H	CAN0 message data byte 67 register 14	C0MDB614	C0MDB6714	R/W	–	√	√	Undefined
F06E7H		C0MDB714						
F06E8H	CAN0 message data length register 14	C0MDLC14		R/W	–	√	–	0XH
F06E9H	CAN0 message Configuration register 14	C0MCONF14		R/W	–	√	–	Undefined
F06EAH	CAN0 message ID register 14L	C0MIDL14		R/W	–	–	√	Undefined
F06ECH	CAN0 message ID register 14H	C0MIDH14		R/W	–	–	√	Undefined
F06EEH	CAN0 message control register 14	C0MCTRL14		R/W	–	–	√	Undefined
F06F0H	CAN0 message data byte 01 register 15	C0MDB015	C0MDB0115	R/W	–	√	√	Undefined
F06F1H		C0MDB115						
F06F2H	CAN0 message data byte 23 register 15	C0MDB215	C0MDB2315	R/W	–	√	√	Undefined
F06F3H		C0MDB315						
F06F4H	CAN0 message data byte 45 register 15	C0MDB415	C0MDB4515	R/W	–	√	√	Undefined
F06F5H		C0MDB515						
F06F6H	CAN0 message data byte 67 register 15	C0MDB615	C0MDB6715	R/W	–	√	√	Undefined
F06F7H		C0MDB715						
F06F8H	CAN0 message data length register 15	C0MDLC15		R/W	–	√	–	0xH
F06F9H	CAN0 message Configuration register 15	C0MCONF15		R/W	–	√	–	Undefined
F06FAH	CAN0 message ID register 15L	C0MIDL15		R/W	–	–	√	Undefined
F06FCH	CAN0 message ID register 15H	C0MIDH15		R/W	–	–	√	Undefined
F06FEH	CAN0 message control register 15	C0MCTRL15		R/W	–	–	√	Undefined

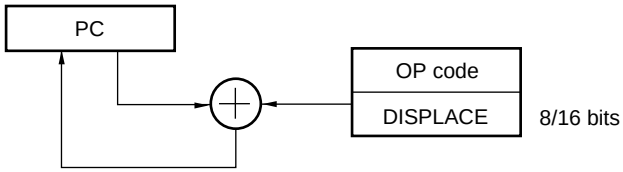
3.3 Instruction Address Addressing

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 3-26. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

Figure 3-27. Example of CALL !!addr20/BR !!addr20

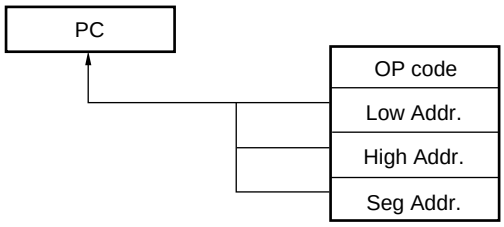
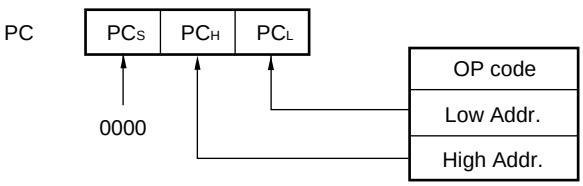


Figure 3-28. Example of CALL !addr16/BR !addr16



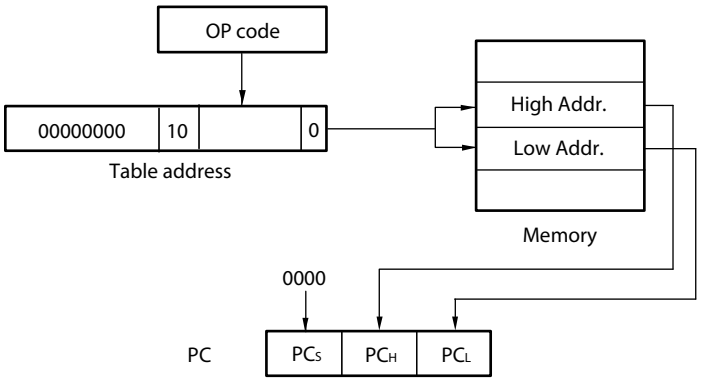
3.3.3 Table indirect addressing

[Function]

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

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

Figure 3-29. Outline of Table Indirect Addressing

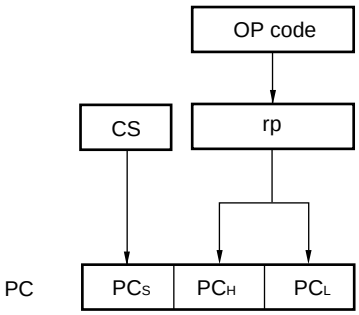


3.3.4 Register direct addressing

[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 3-30. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

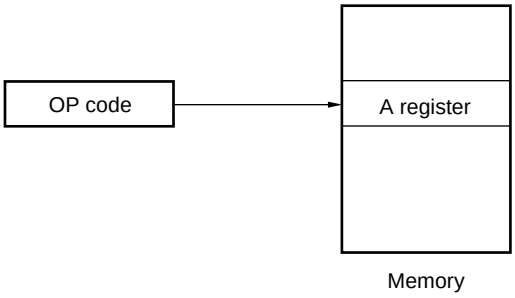
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[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 3-31. Outline of Implied Addressing



3.4.2 Register addressing

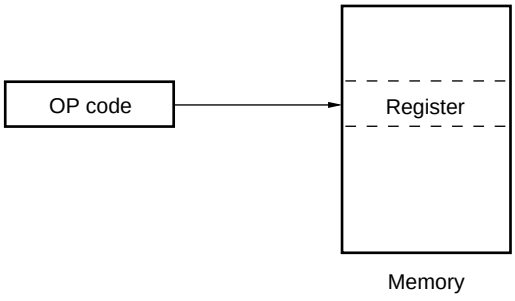
[Function]

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 3-32. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

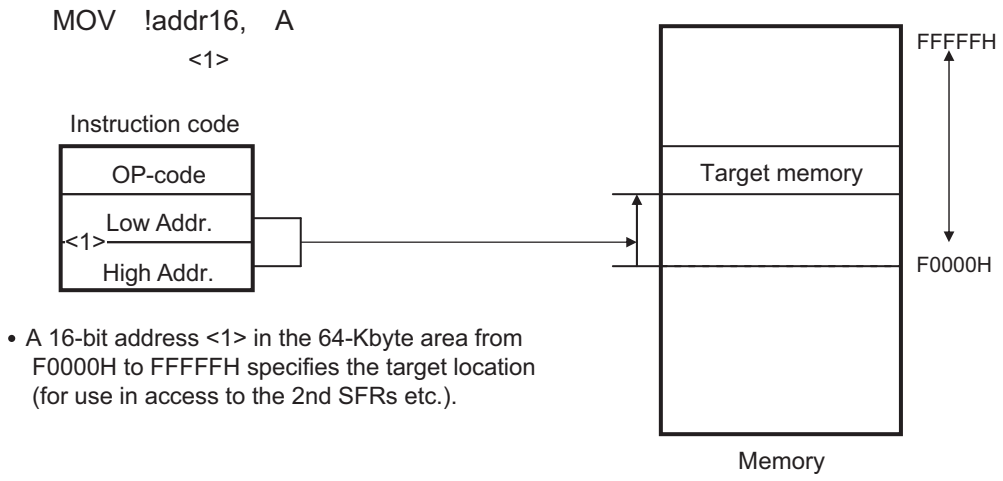
Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[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)

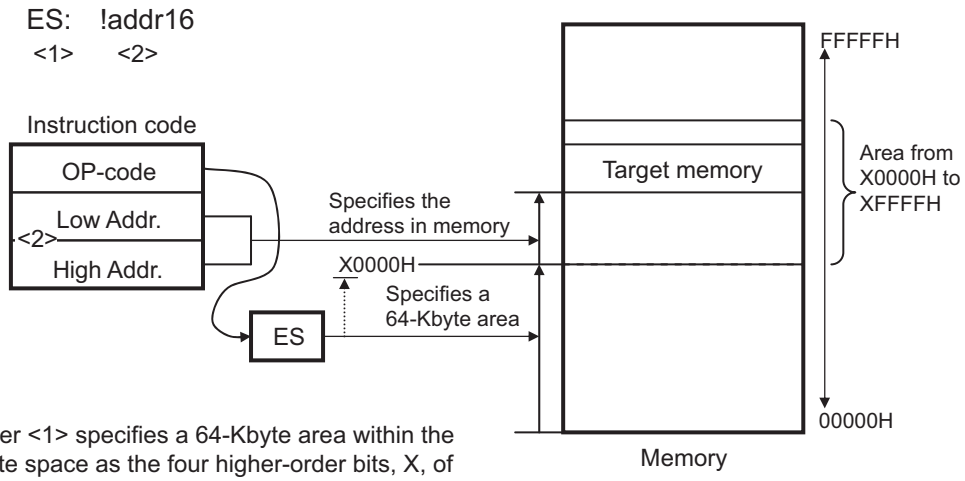
<R>

Figure 3-33. Example of ADDR16



<R>

Figure 3-34. Example of ES:ADDR16



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

3.4.4 Short direct addressing

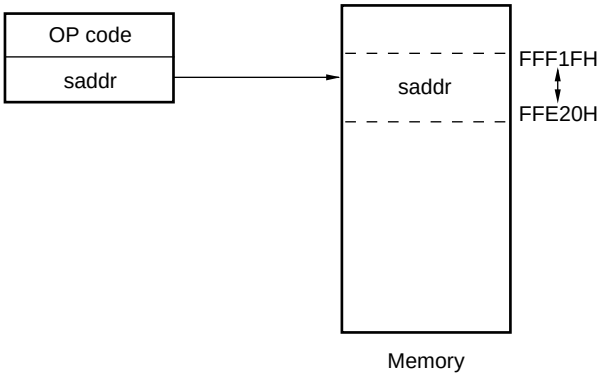
[Function]

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

[Operand format]

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

Figure 3-35. 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

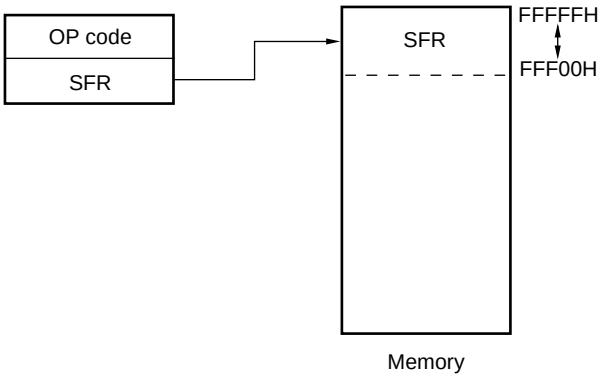
[Function]

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 3-36. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

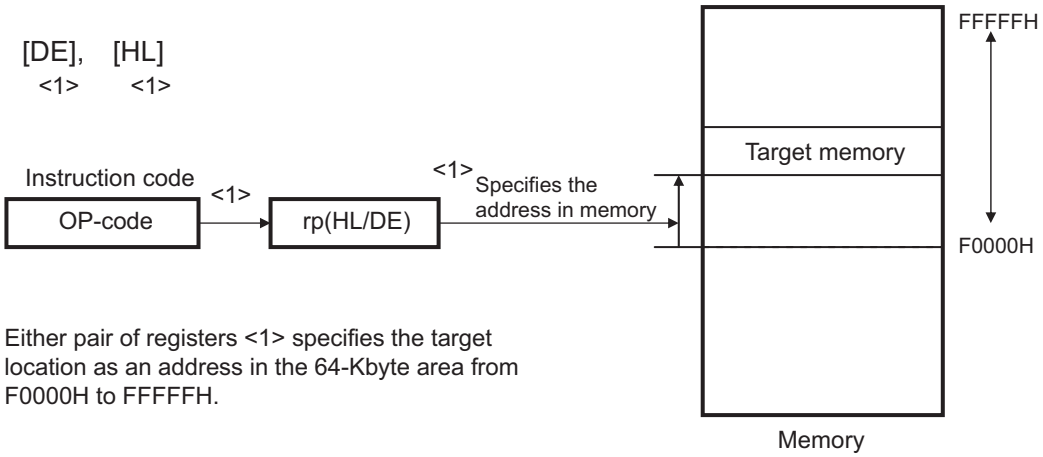
Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[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)

<R>

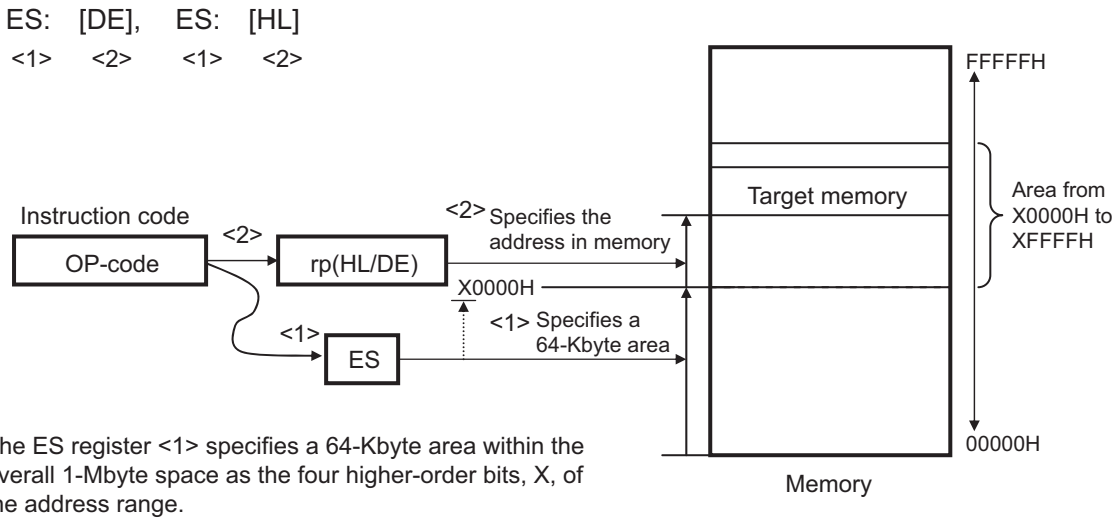
Figure 3-37. Example of [DE], [HL]



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

<R>

Figure 3-38. Example of ES:[DE], ES:[HL]



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

<R> 3.4.7 Based addressing

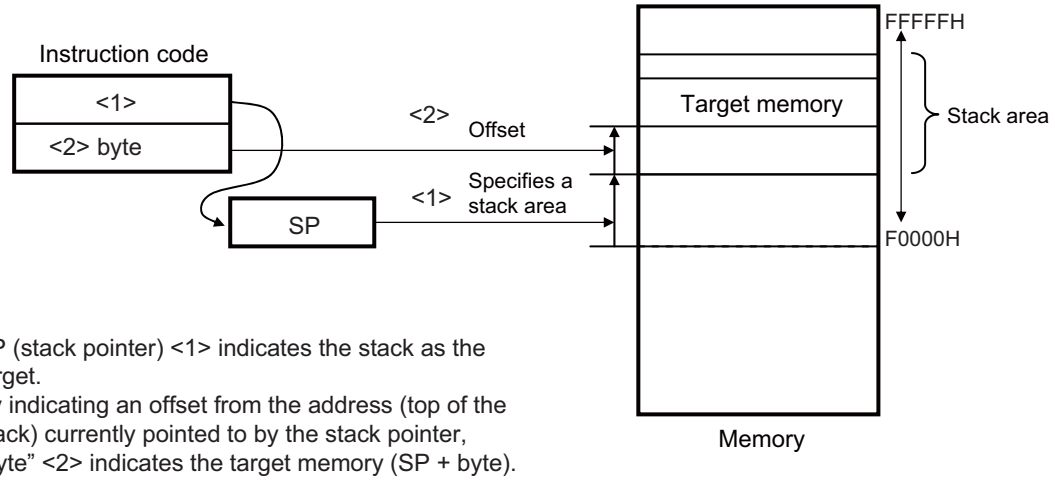
[Function]

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

[Operand format]

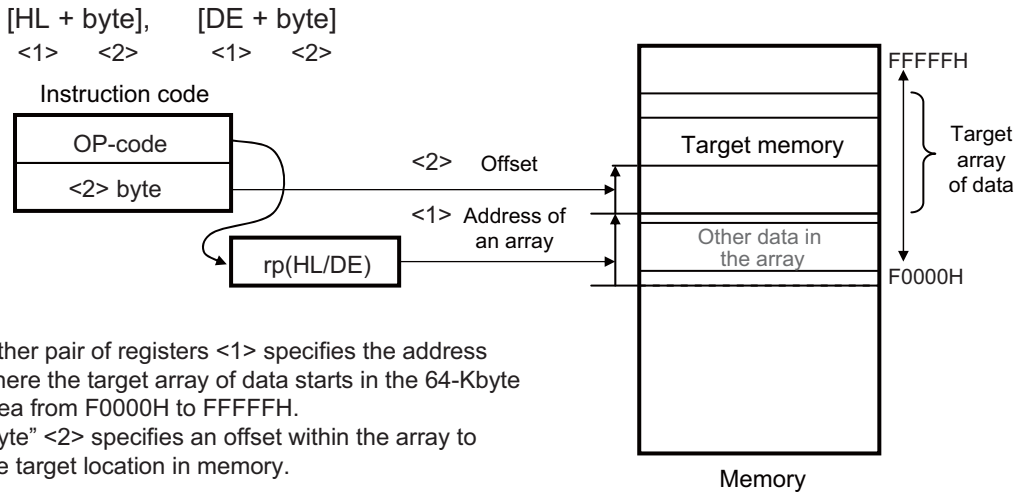
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit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

Figure 3-39. Example of [SP+byte]



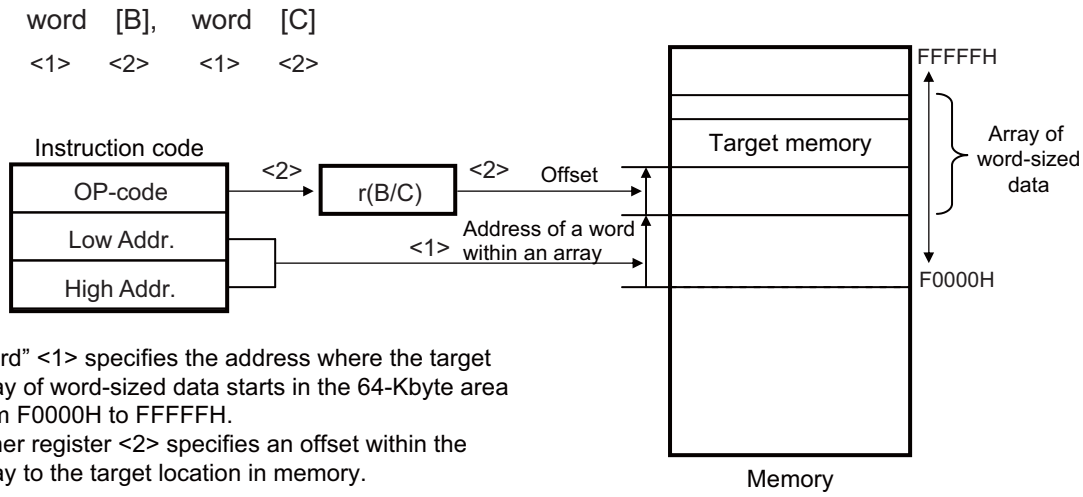
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

Figure 3-40. Example of [HL + byte], [DE + byte]



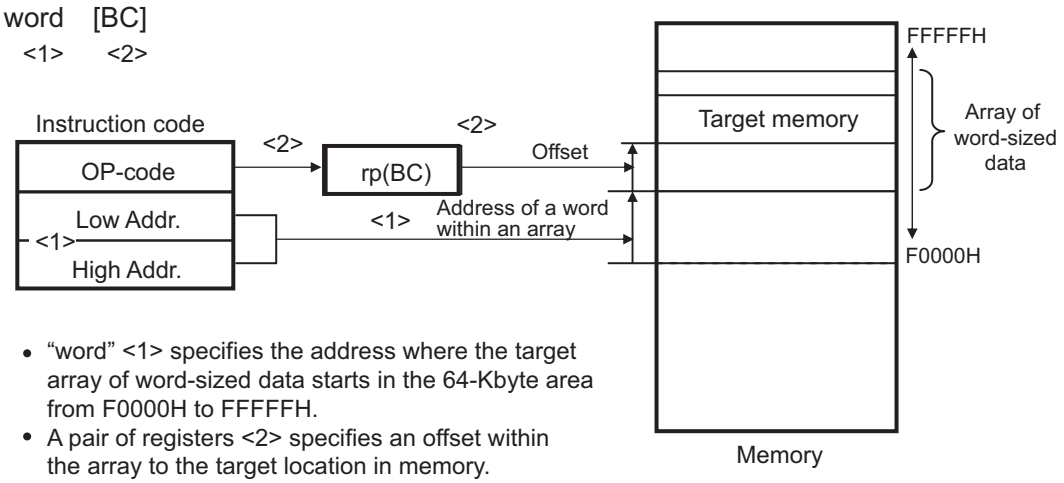
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3-41. Example of word[B], word[C]



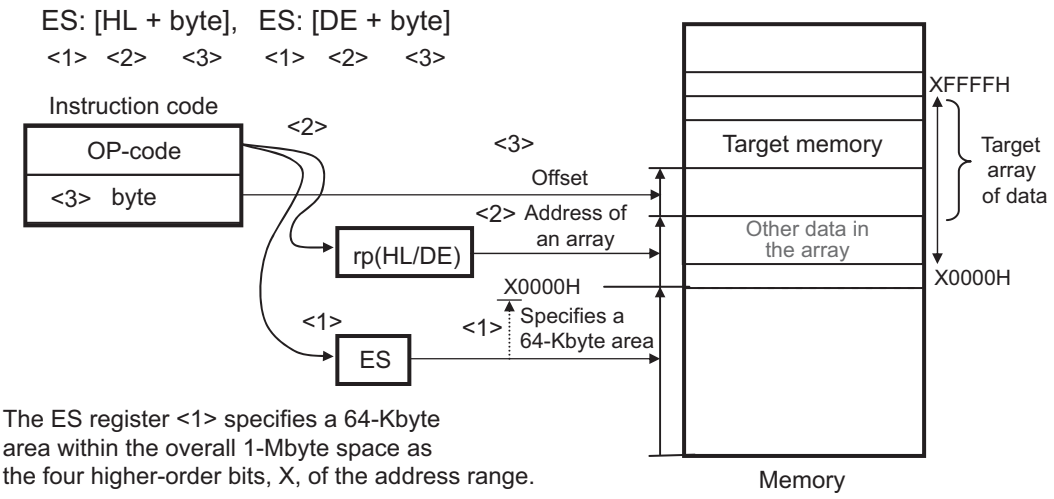
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-42. Example of word[BC]



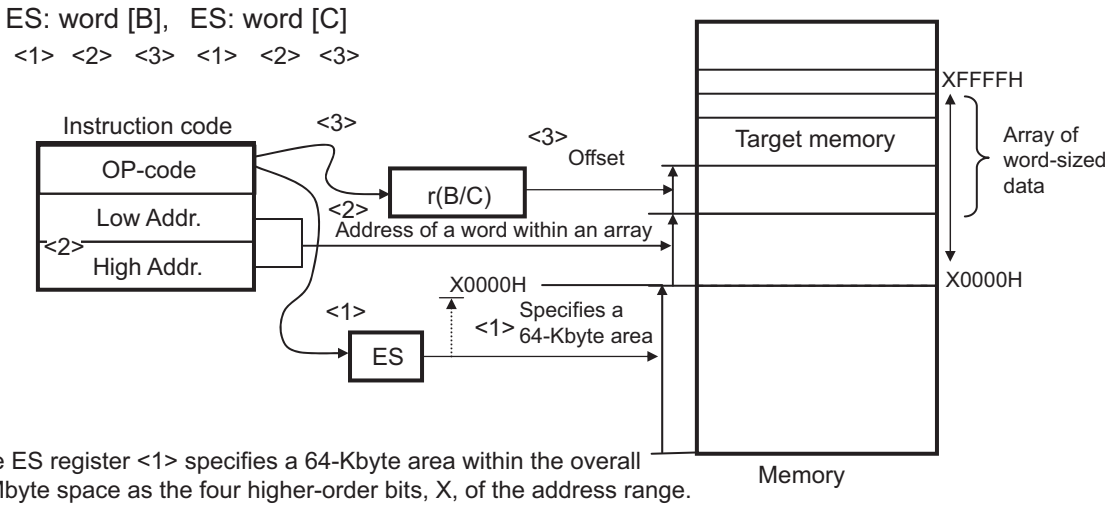
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3-43. Example of ES:[HL + byte], ES:[DE + byte]



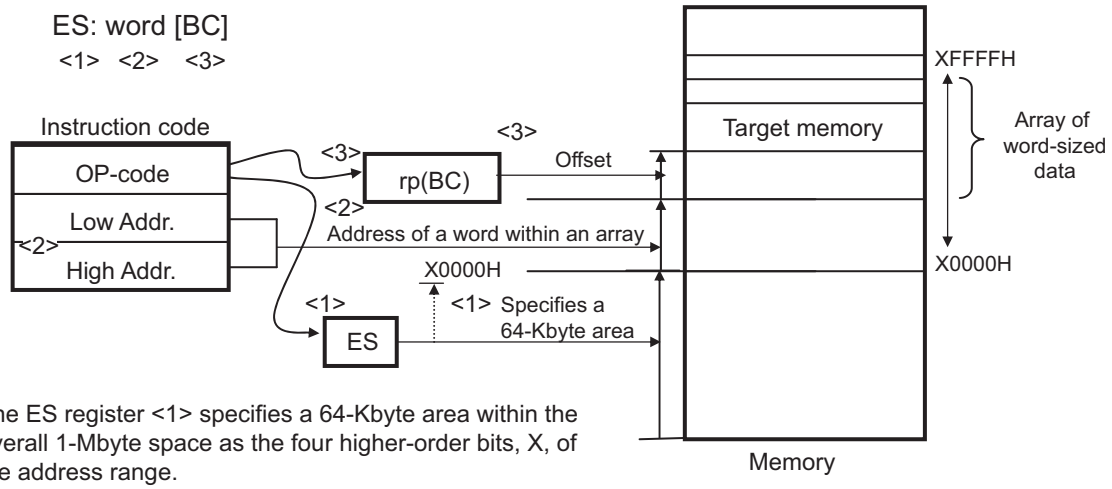
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies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

Figure 3-44. Example of ES:word[B], ES:word[C]



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

Figure 3-45. Example of ES:word[BC]



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

<R> 3.4.8 Based indexed addressing

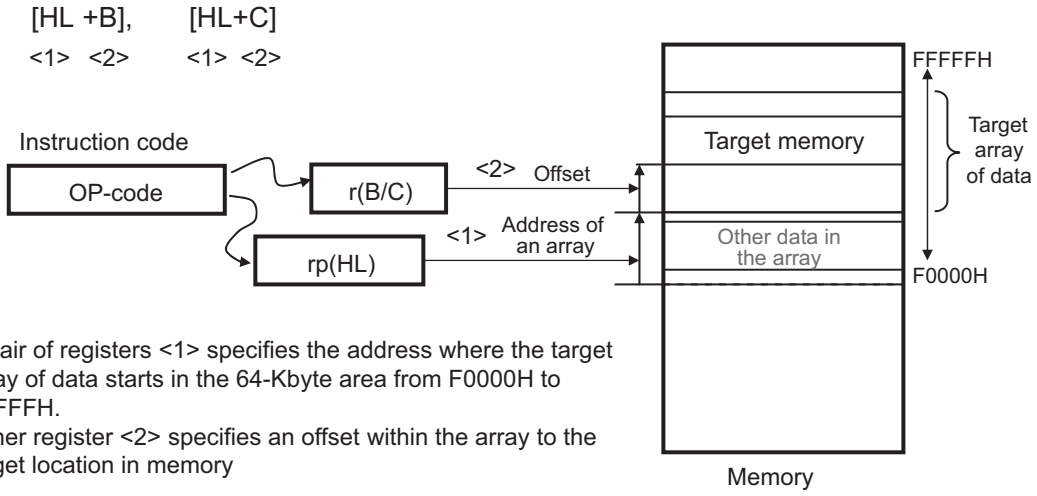
[Function]

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

[Operand format]

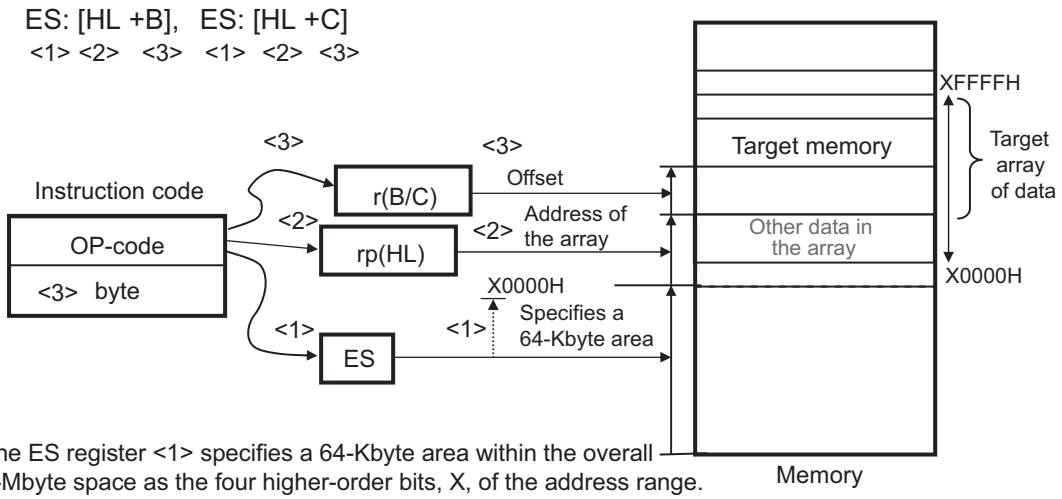
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3-46. Example of [HL+B], [HL+C]



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

Figure 3-47. Example of ES:[HL+B], ES:[HL+C]



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

<R> 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.

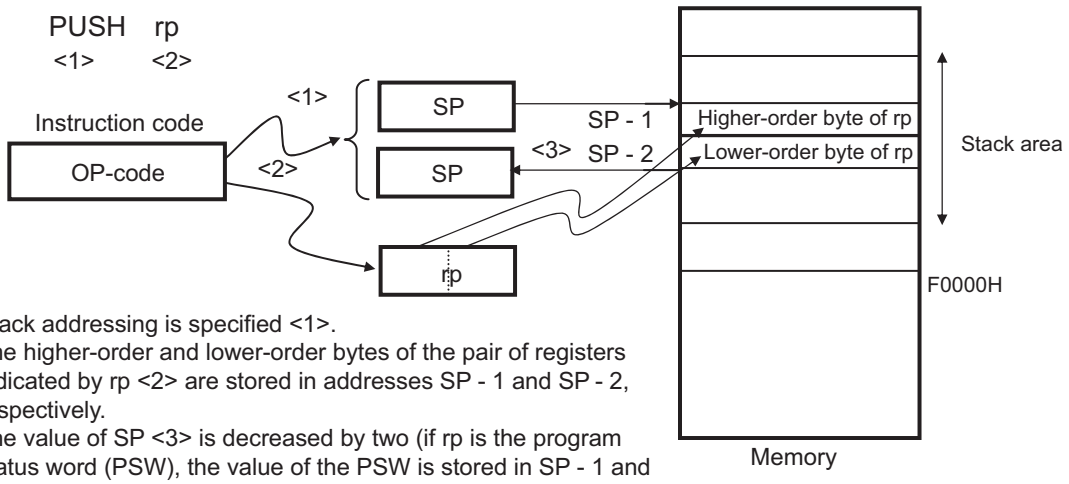
Stack addressing is applied only to the internal RAM area.

[Operand format]

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

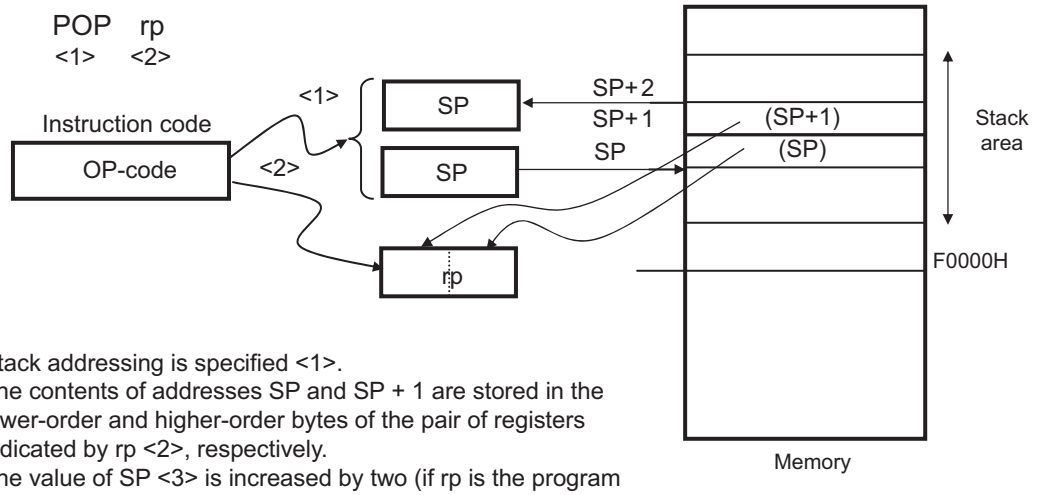
Each stack operation saves or restores data as shown in **Figures 3-48** to **3-53**.

Figure 3-48. Example of PUSH rp



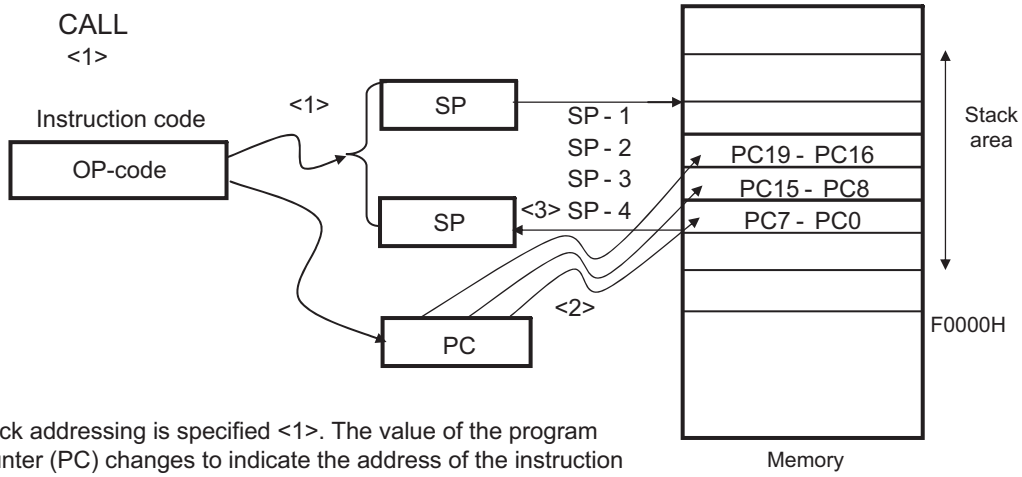
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP - 1 and SP - 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP -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

Figure 3-49. Example of POP



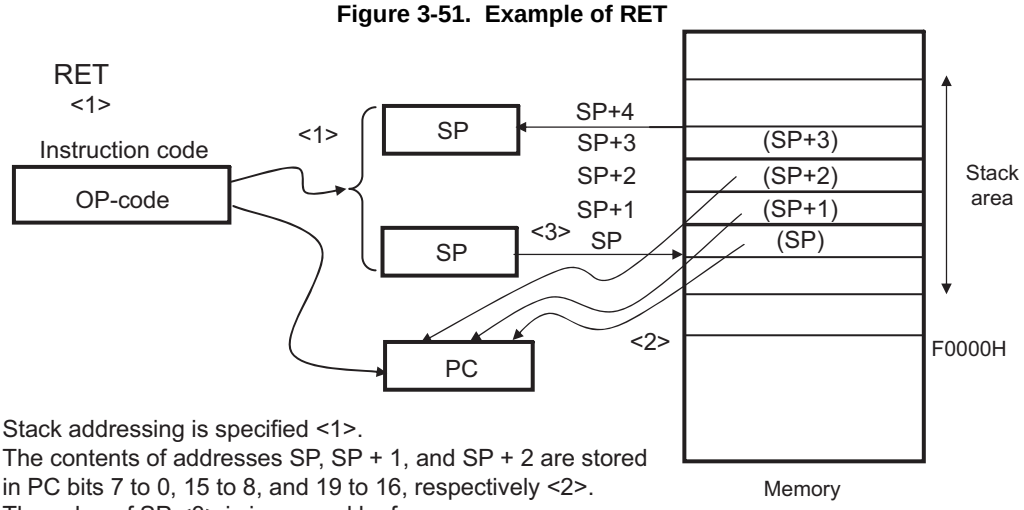
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-50. Example of CALL, CALLT



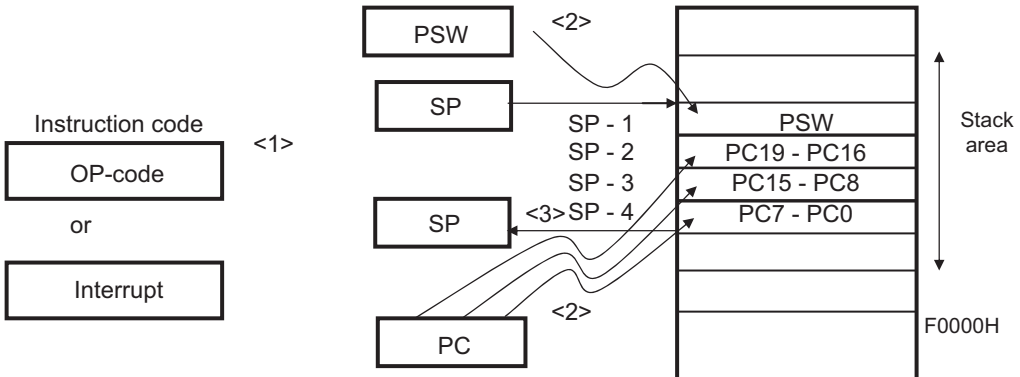
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- The values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the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

Figure 3-51. Example of RET



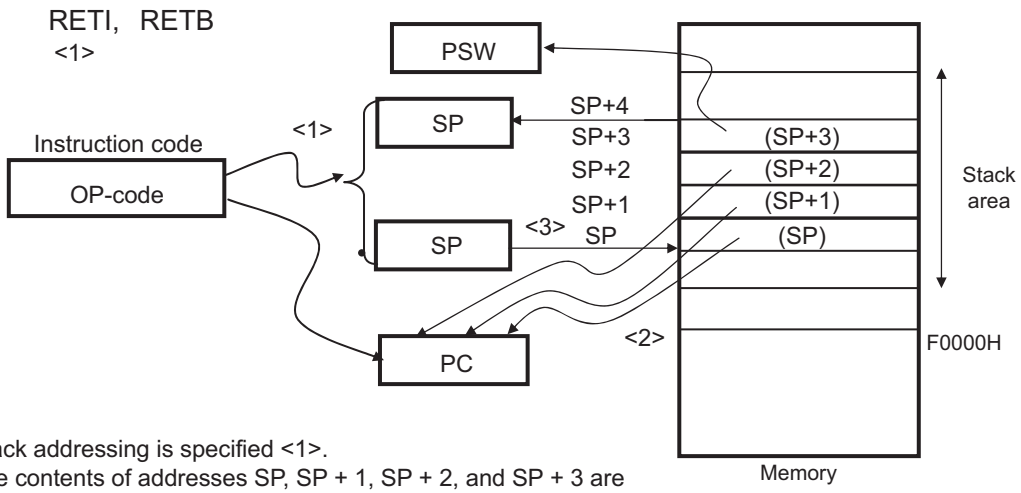
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-52. Example of Interrupt, BRK



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

Figure 3-53. Example of RETI, RETB



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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/D1A microcontrollers are provided with digital I/O ports, which enable variety of control operations.

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

<R> 4.2 Port Configuration

Ports include the following hardware.

Table 4-1. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM15) Port registers (P0 to P15) Pull-up resistor option registers (PU0, PU1, PU3 to PU14) Port input mode registers (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM11, PIM13) Port output mode registers (POM) LCD port function register (LCDPF0, LCDPF1, LCDPF3 to LCDPF5, LCDPF7 to LCDPF13) A/D port configuration register (ADPC) Stepper motor port control register (SMPC)
Port	• 48-pin products Total: 38 (CMOS I/O: 35 (LED direct drive: 9, N-ch open drain selectable: 4), CMOS input: 3) • 64-pin products Total: 54 (CMOS I/O: 49(LED direct drive: 13, N-ch open drain selectable: 4), CMOS input: 5) • 80-pin products Total: 68 (CMOS I/O: 63(LED direct drive: 16, N-ch open drain selectable: 4), CMOS input: 5) • 100-pin products Total: 84 (CMOS I/O: 78(LED direct drive: 16, N-ch open drain selectable:6), CMOS input: 5, CMOS output: 1) • 128-pin products Total: 112 (CMOS I/O: 106(LED direct drive: 16, N-ch open drain selectable:6), CMOS input: 5, CMOS output: 1)
Pull-up resistor	• 48-pin products Total: 30 • 64-pin products Total: 44 • 80-pin products Total: 55 • 100-pin products Total: 69 • 128-pin products Total: 95

4.2.1 Port 0

48-pin products: P00 to P01 function as a 2-bit I/O port.
64-pin products: P00 to P05 and P07 function as a 7-bit I/O port.
80-pin products: P00 to P07 function as an 8-bit I/O port.
100-pin products: P00 to P07 function as an 8-bit I/O port.
<R> 128-pin products: P00 to P07 function as an 8-bit I/O port.

Port 0 is an 8-bit or a 7-bit, or a 2-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P07 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P01 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer using port input mode register 0 (PIM0).

These pins also function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

To use P00 to P07 as the port function, refer Table 4-2.

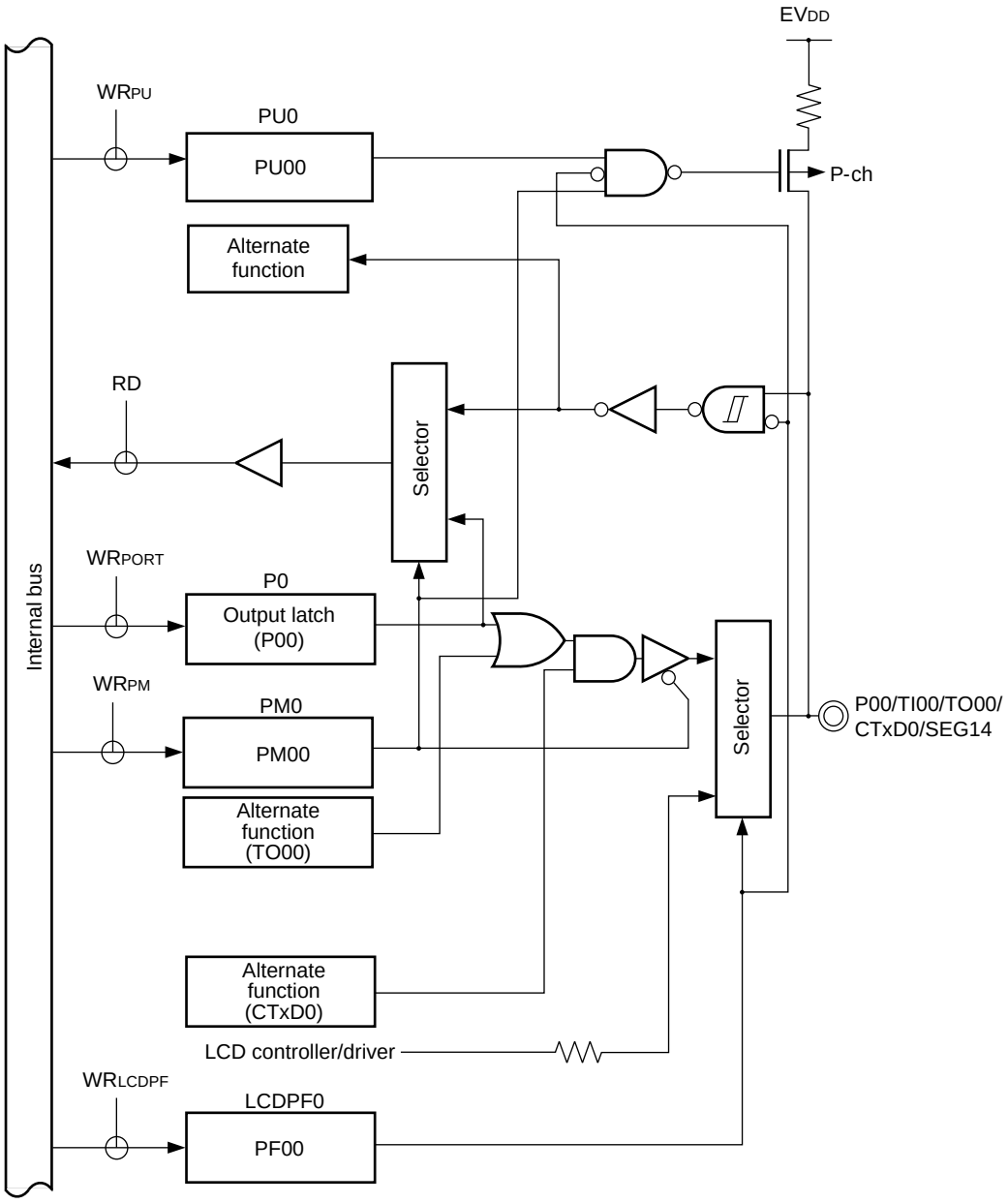
Table 4-2. Setting of P00 to P07 Pins to port function

P00 to P07 Pins		LCDPF0	Alternate function		PM0	PIM0	POM	Remarks
port	function	Register	Timer	Serial	Register	Register	Register	
P00	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P01	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
P02	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P03	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P04	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P05	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P06	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P07	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
-	-	LCD segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 0 to input mode.

Figures 4-1 to 4-6 show block diagrams of port 0.

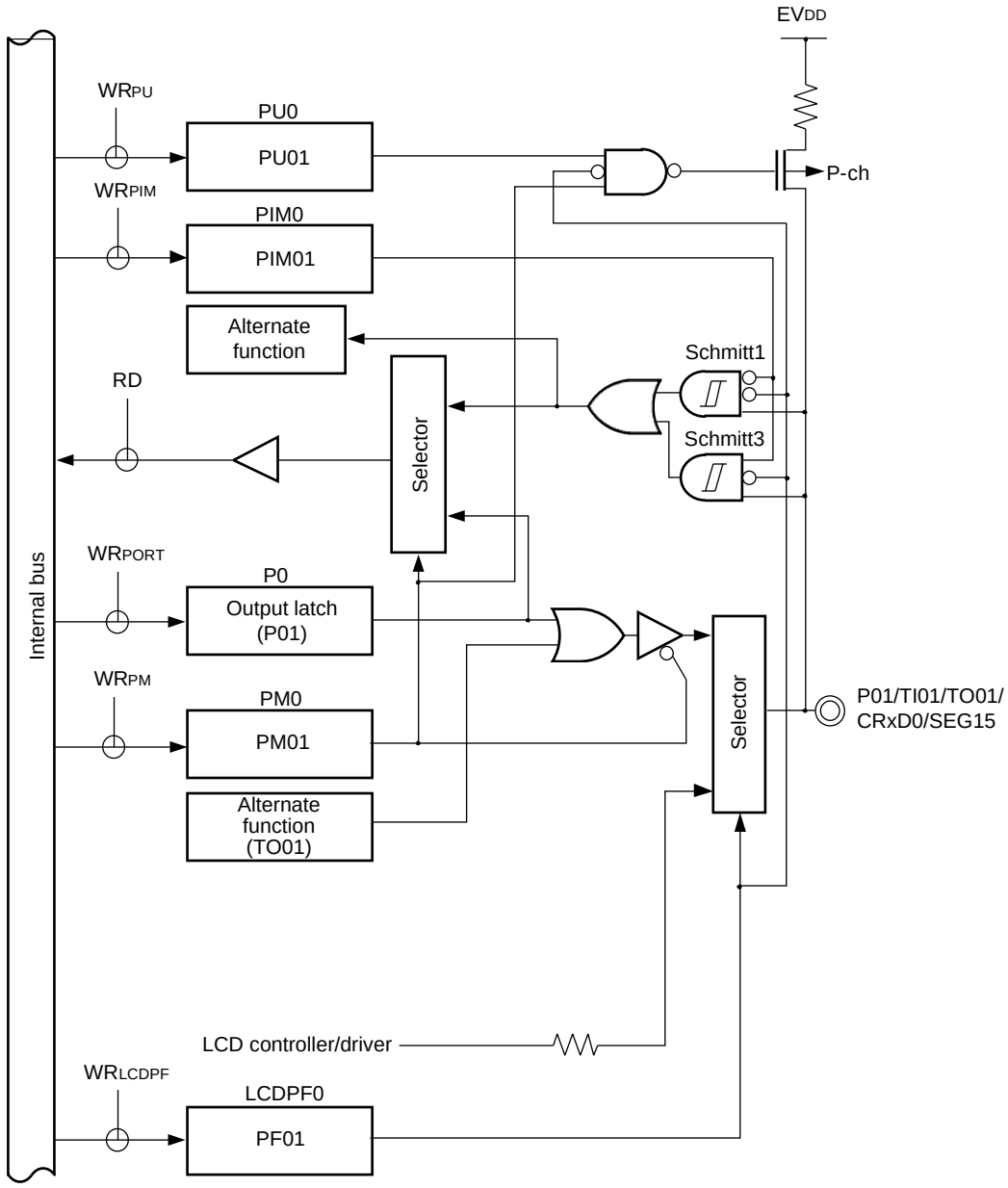
Figure 4-1. Block Diagram of P00



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
LCDPF0: LCD port function registers 0
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO00, set the port latch to 0.
When using the alternate function CTxD0, set the port latch to 1.
When using P00 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

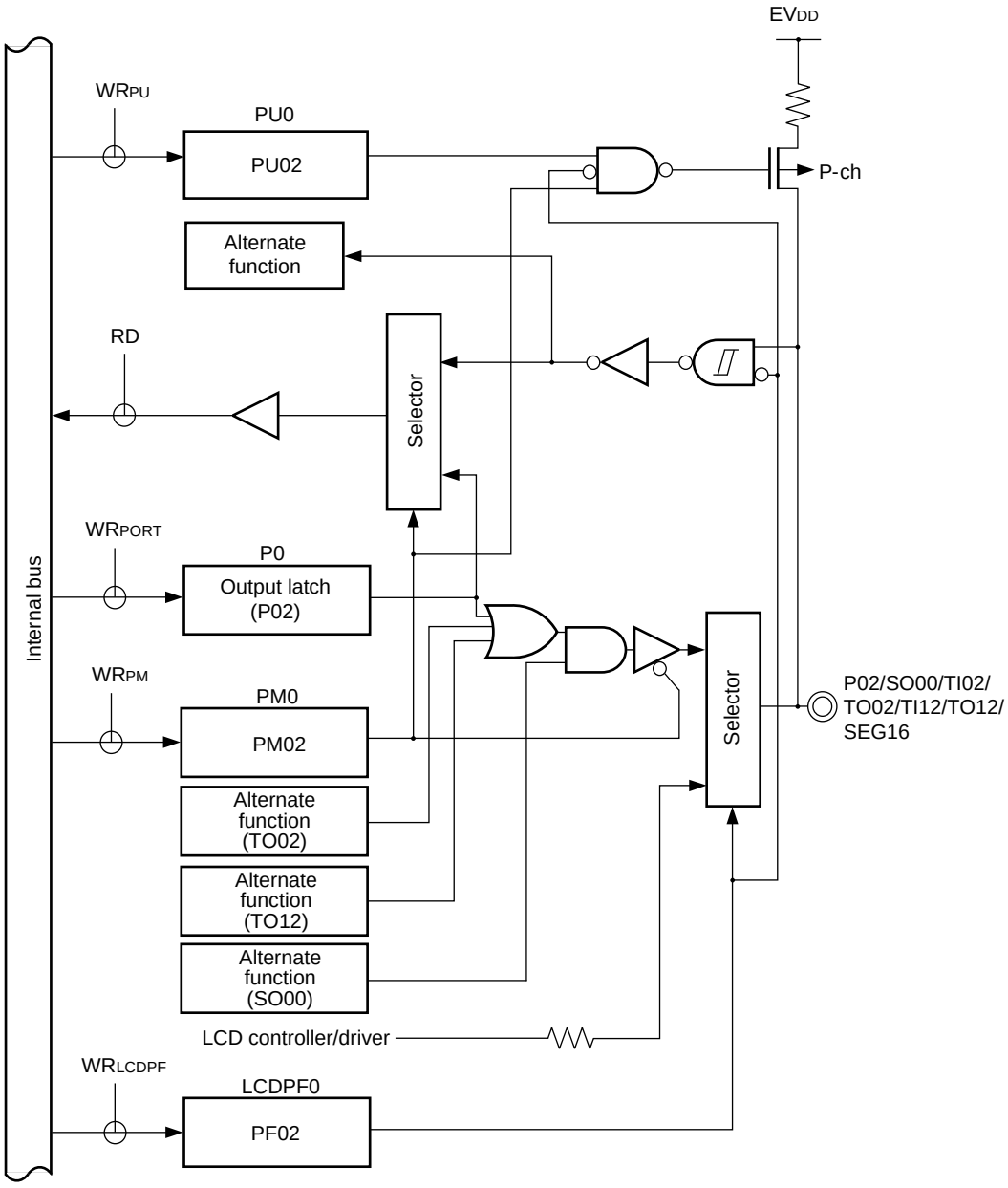
Figure 4-2. Block Diagram of P01



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PIM0: Port Input mode register 0
LCDPF0: LCD port function registers 0
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO01, set the port latch to 0.
When using P01 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

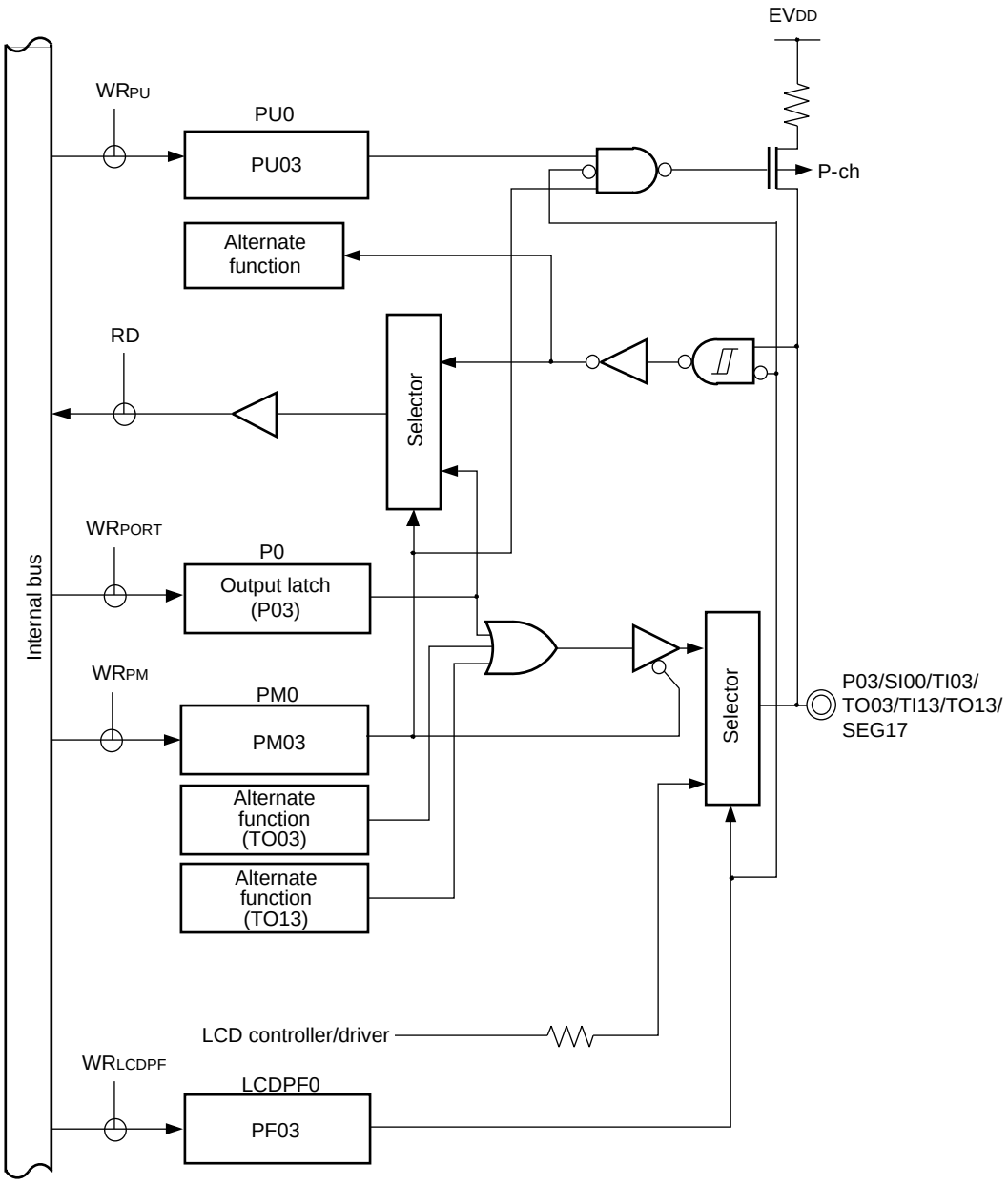
Figure 4-3. Block Diagram of P02



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
LCDPF0: LCD port function registers 0
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO02 or TO12, set the port latch to 0.
When using the alternate function SO00, set the port latch to 1.
When using P02 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

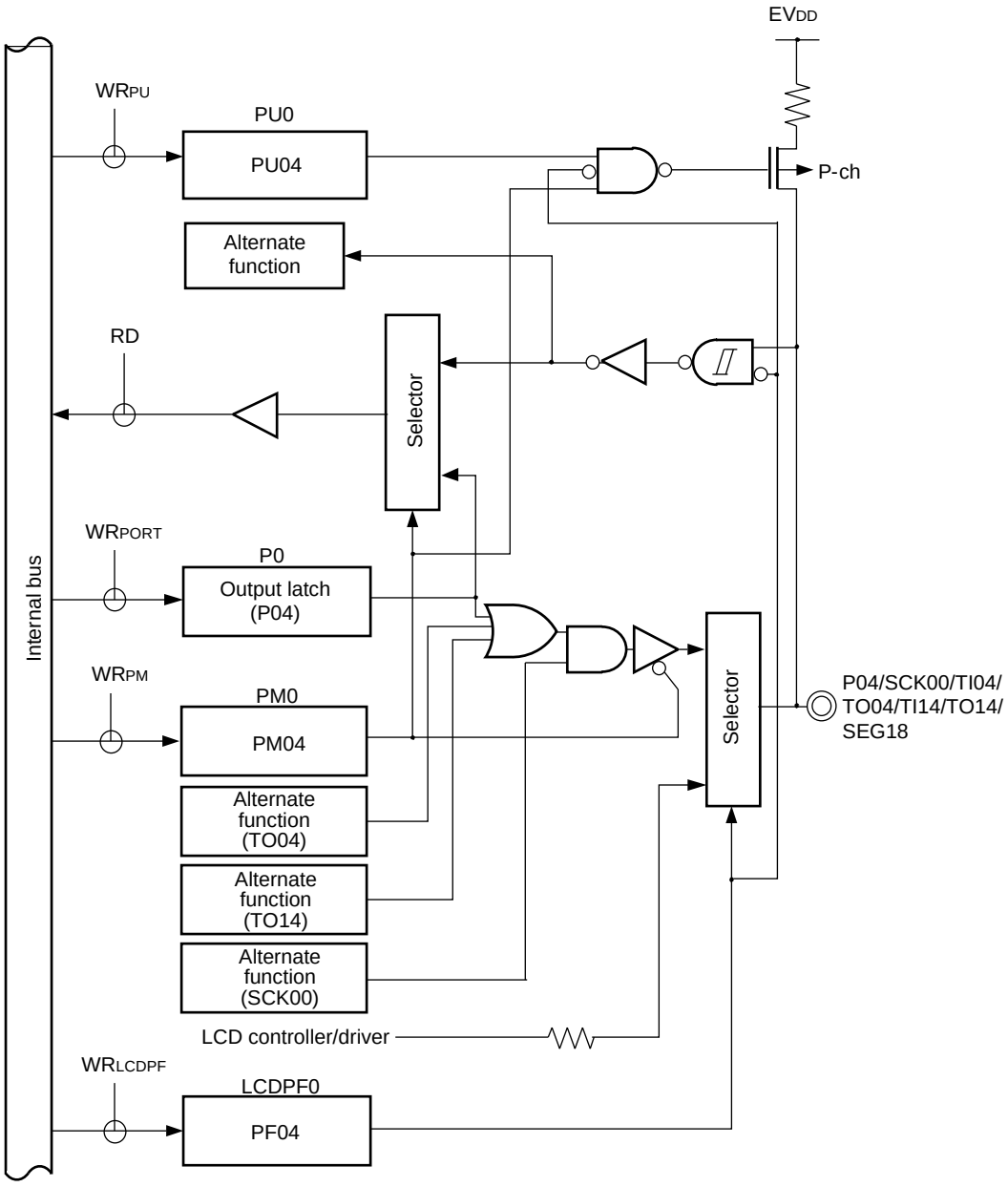
Figure 4-4. Block Diagram of P03



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
LCDPF0: LCD port function registers 0
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO03 or TO13, set the port latch to 0.
When using P03 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

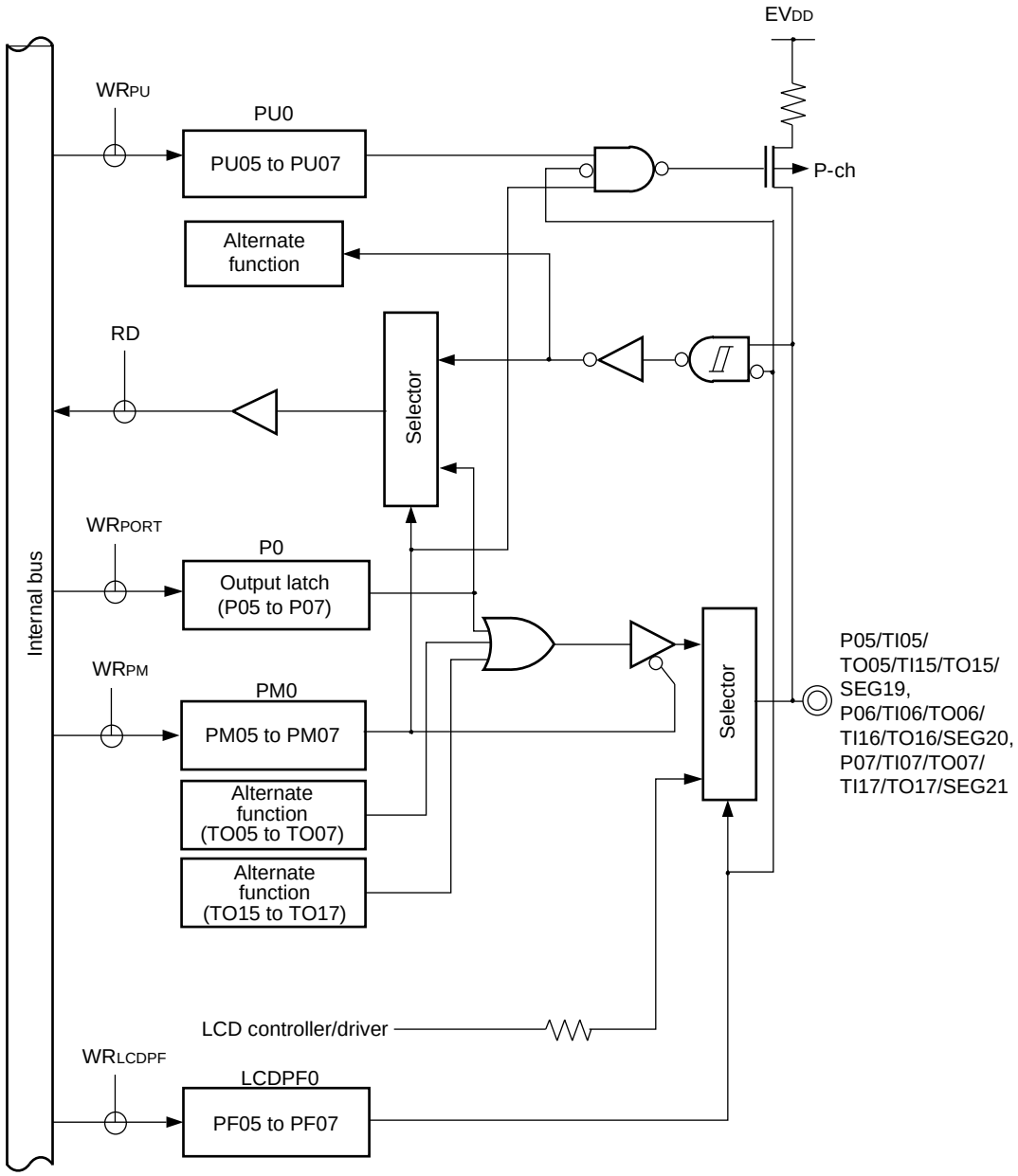
Figure 4-5. Block Diagram of P04



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
LCDPF0: LCD port function registers 0
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO04 or TO14, set the port latch to 0.
When using the alternate function SCK00, set the port latch to 1.
When using P04 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-6. Block Diagram of P05 to P07



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
LCDPF0: LCD port function registers 0
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO05 to TO07 or TO15 to TO17, set the port latch to 0.
When using P05 to P07 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

4.2.2 Port 1

48-pin products: P10 to P14 function as a 5-bit I/O port.
64-pin products: P10 to P15 and P17 function as a 7-bit I/O port.
80-pin products: P10 to P17 function as an 8-bit I/O port.
100-pin products: P10 to P17 function as an 8-bit I/O port.
<R> 128-pin products: P10 to P17 function as an 8-bit I/O port.

Port 1 is an 8-bit or a 7-bit, or a 5-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10, P11, and P17 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer using port input mode register 1 (PIM1).

These pins also function as serial interface data I/O, timer I/O, clock I/O, external interrupt request input and segment signal outputs for the LCD controller/driver.

To use P10 to P17 as the port function, refer Table 4-3.

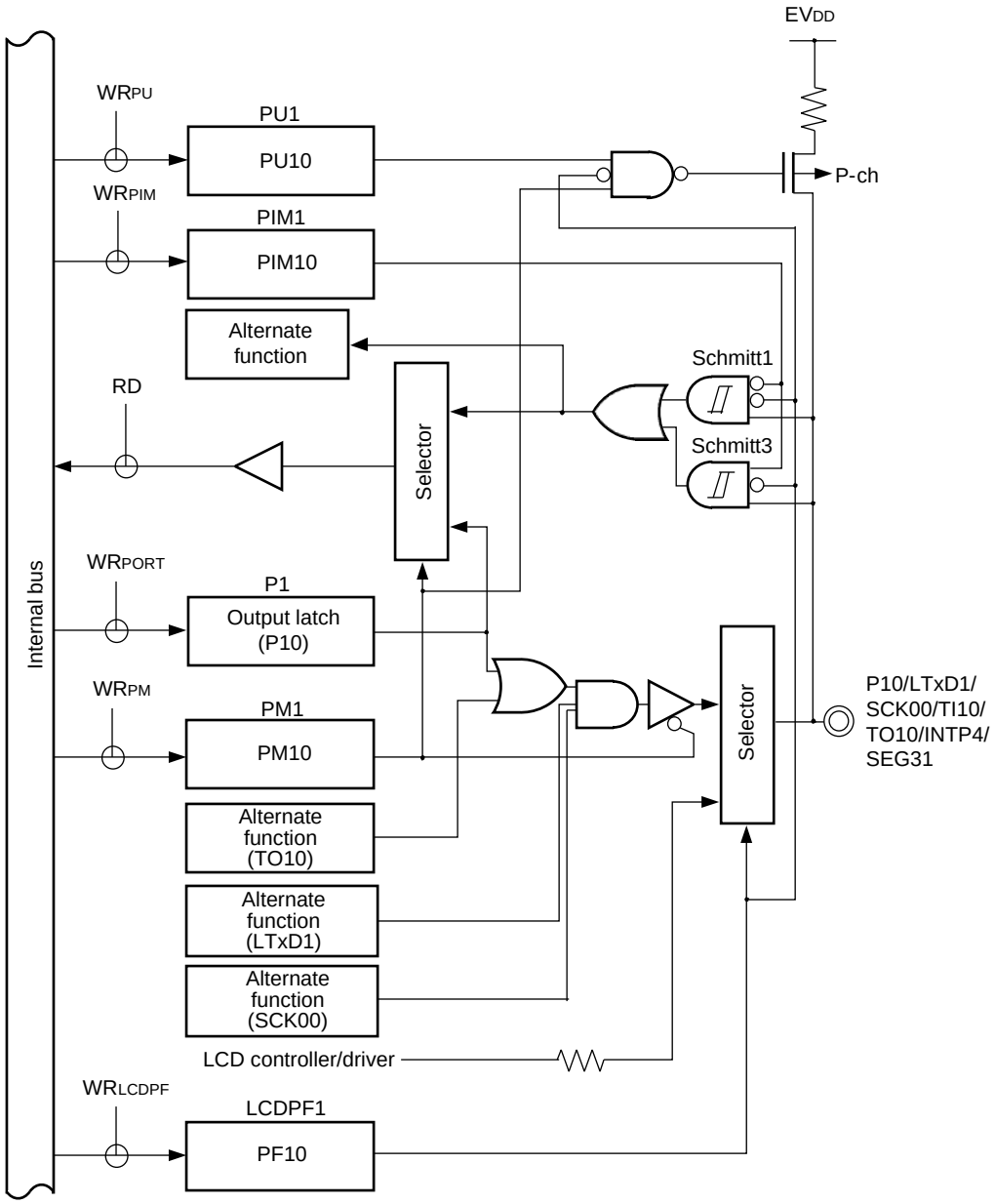
Table 4-3. Setting of P10 to P17 Pins to port function

P10 to P17 Pins		LCDPF1 Register	Alternate function		PM1 Register	PIM1 Register	POM Register	Remarks
port	function		Timer	Serial				
P10	Input port	Digital I/O selection	-	-	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0	1	Output mode			
P11	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
P12	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P13	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P14	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P15	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P16	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P17	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
-	-	LCD segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 1 to input mode.

Figures 4-7 to 4-11 show block diagrams of port 1.

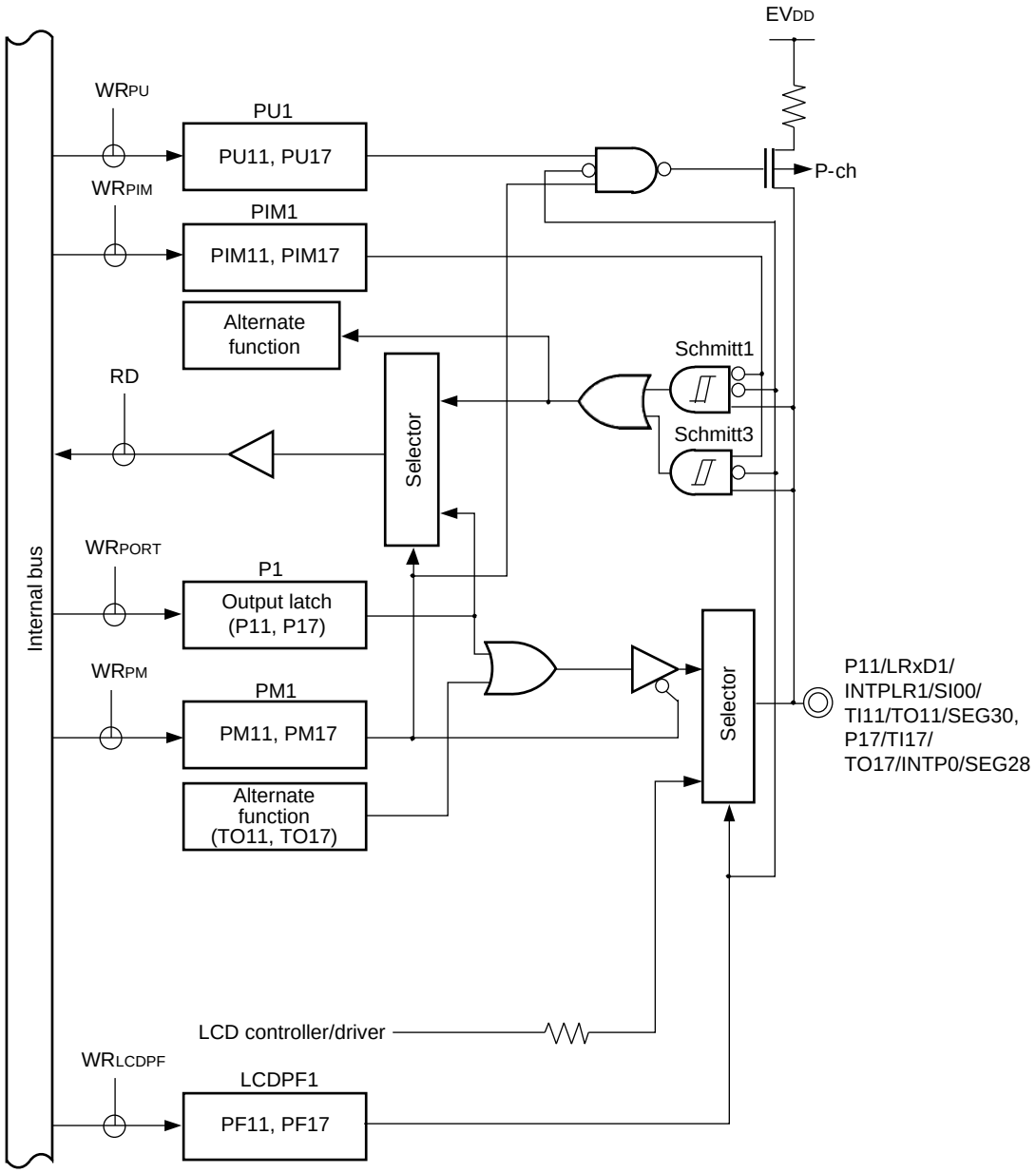
Figure 4-7. Block Diagram of P10



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port Input mode register 1
LCDPF1: LCD port function registers 1
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO10, set the port latch to 0.
When using the alternate function LTxD1 or SCK00, set the port latch to 1.
When using P10 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-8. Block Diagram of P11, P17

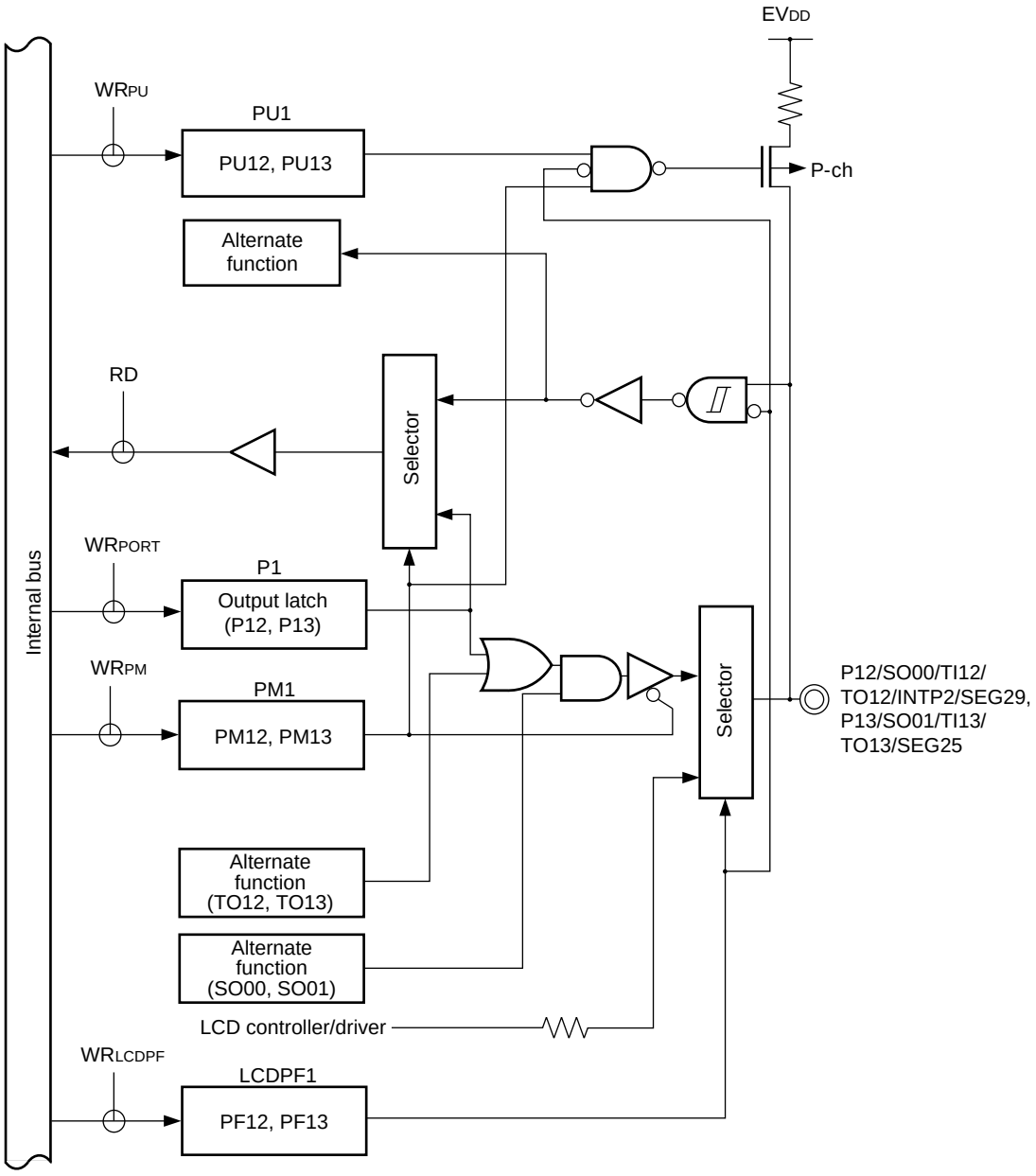


P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port Input mode register 1
LCDPF1: LCD port function registers 1
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO11 or TO17, set the port latch to 0.

When using P11 or P17 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

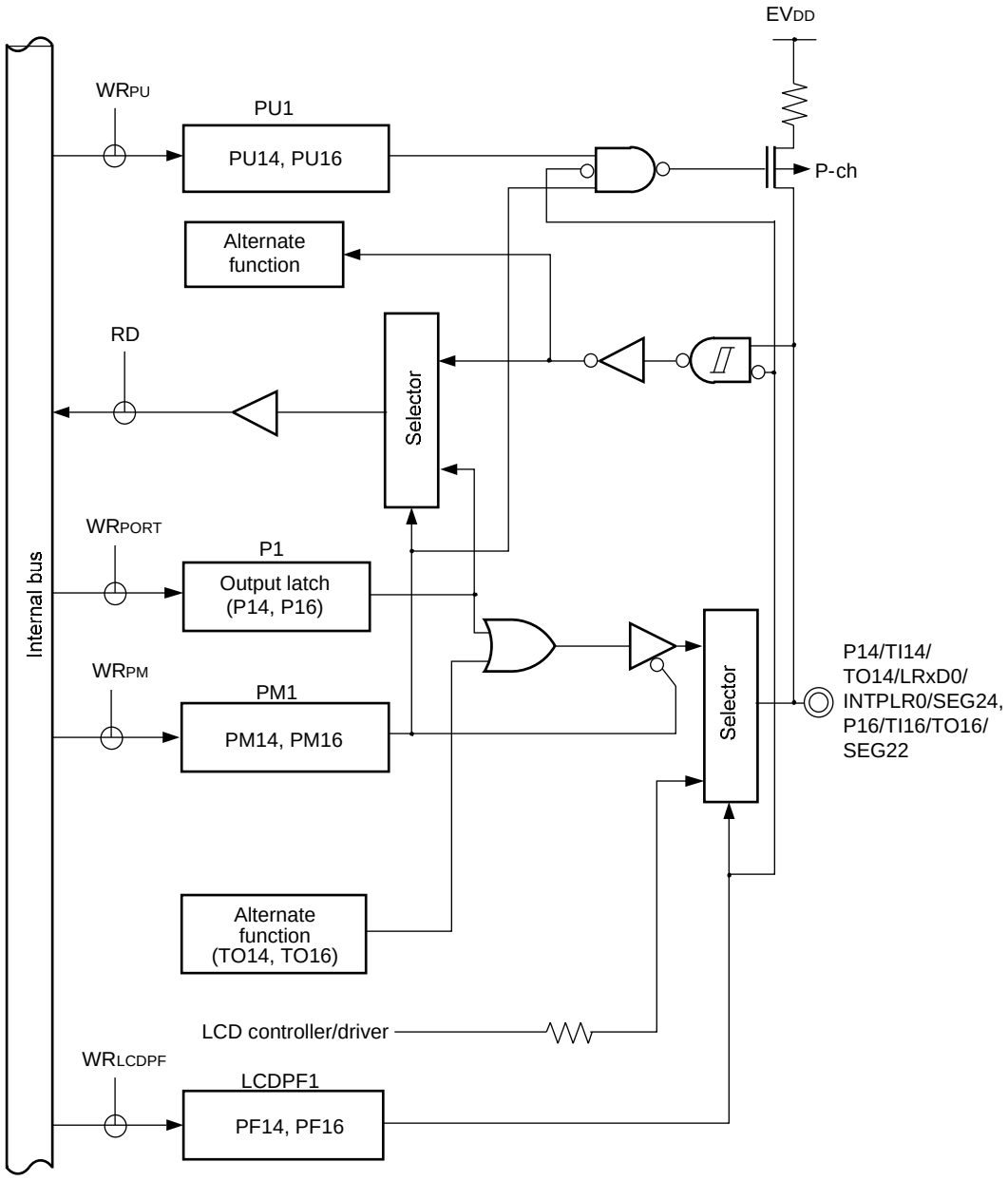
Figure 4-9. Block Diagram of P12, P13



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
LCDPF1: LCD port function registers 1
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO12 or TO13, set the port latch to 0.
When using the alternate function SO00 or SO01, set the port latch to 1.
When using P12 or P13 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

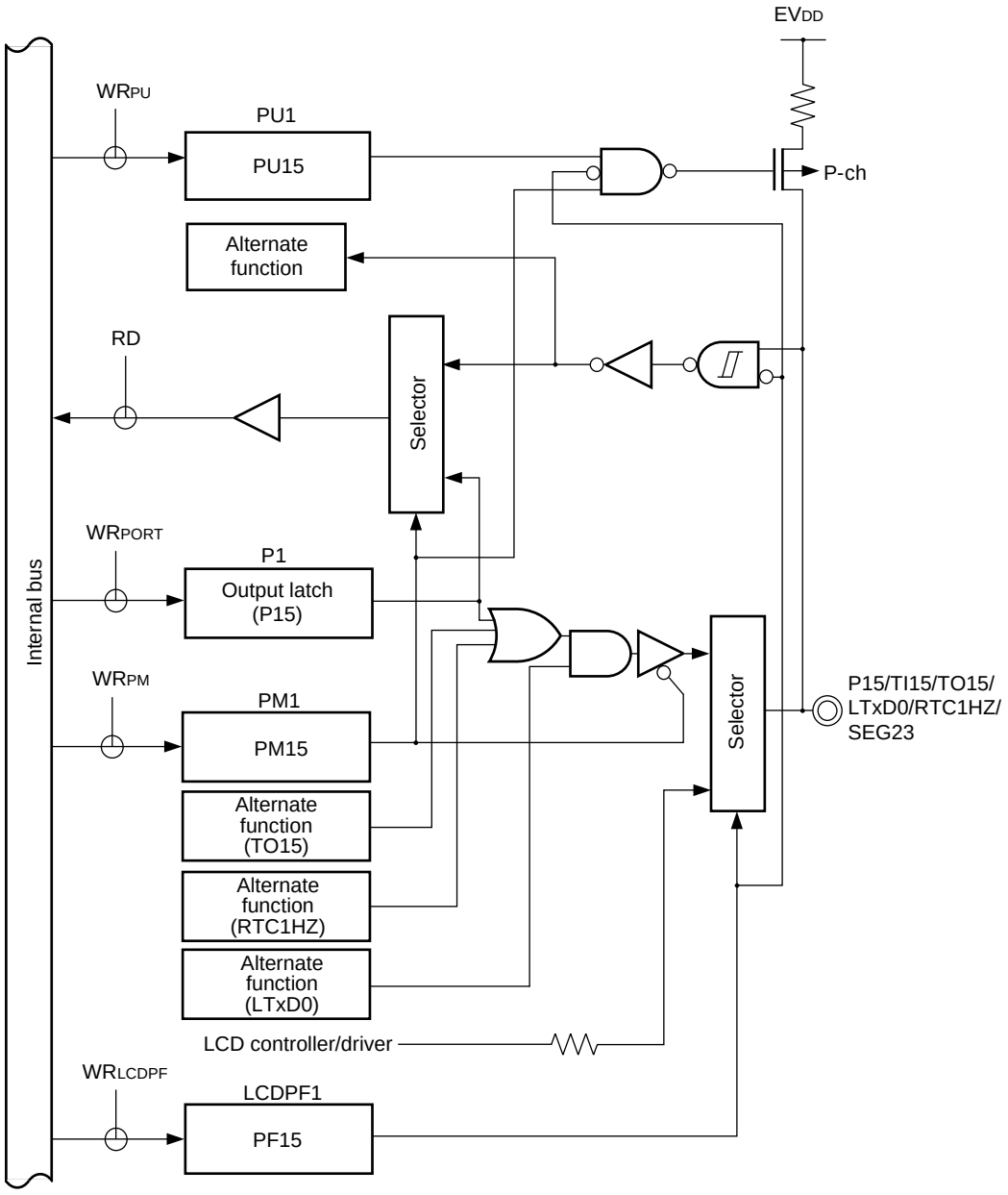
Figure 4-10. Block Diagram of P14, P16



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
LCDPF1: LCD port function registers 1
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO14 or TO16, set the port latch to 0.
When using P14 or P16 as a general-purpose port, specify the port settings so that the alternate function output is fixed 0.

Figure 4-11. Block Diagram of P15



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
LCDPF1: LCD port function registers 1
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO15, set the port latch to 0.
When using the alternate function LTxD0, set the port latch to 1.
When using P15 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

4.2.3 Port 2

48-pin products: P20 to P23 and P27 function as a 5-bit I/O.
64-pin products: P20 to P23 and P27 function as a 5-bit I/O.
80-pin products: P20 to P27 function as an 8-bit I/O port.
100-pin products: P20 to P27 function as an 8-bit I/O port
<R> 128-pin products: P20 to P27 function as an 8-bit I/O port

Port 2 is an 8-bit or a 5-bit I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

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

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

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

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

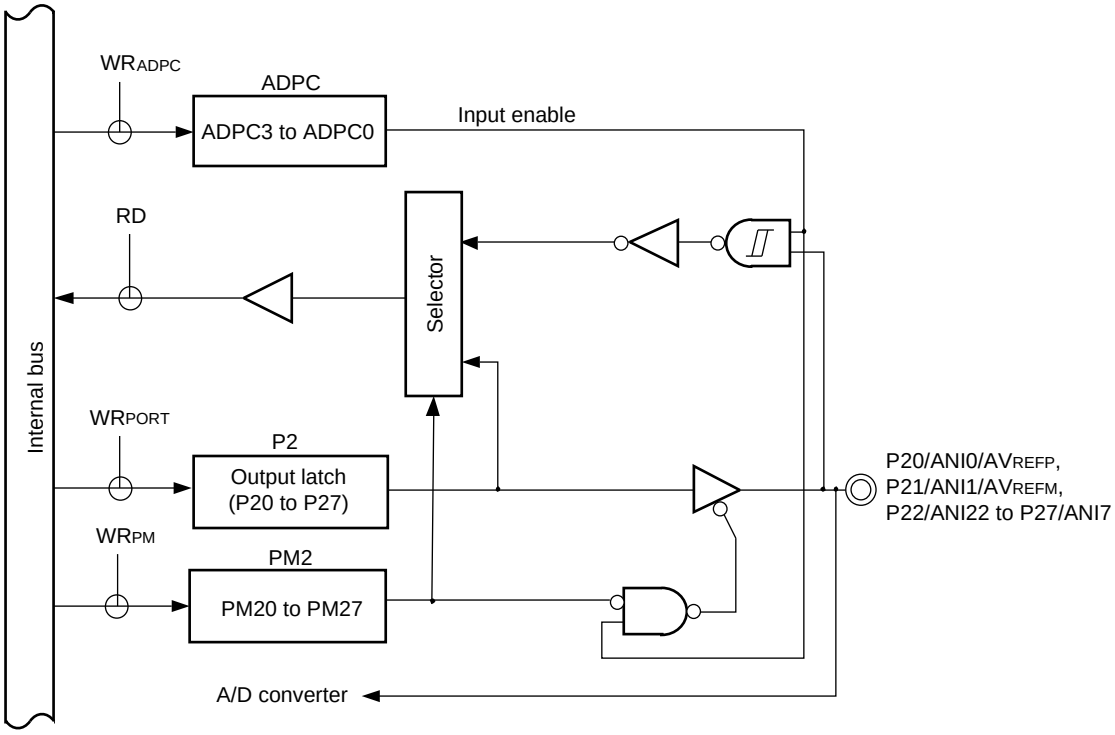
Table 4-4. Setting Functions of P20 to P27 Pins

P20 to P27 Pins		ADPC	PM2	Remarks
port	function	Register	Register	
P20	Input port	0001	Input mode	
	Output port		Output mode	
P21	Input port	0001 or 0010	Input mode	
	Output port		Output mode	
P22	Input port	0001 to 0011	Input mode	
	Output port		Output mode	
P23	Input port	0001 to 0100	Input mode	
	Output port		Output mode	
P24	Input port	0001 to 0101	Input mode	
	Output port		Output mode	
P25	Input port	0001 to 0110	Input mode	
	Output port		Output mode	
P26	Input port	0001 to 0111	Input mode	
	Output port		Output mode	
P27	Input port	0001 to 1000	Input mode	
	Output port		Output mode	

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

Figures 4-12 shows block diagram of port 2.

Figure 4-12. Block Diagram of P20 to P27



P2: Port register 2
PM2: Port mode register 2
ADPC: A/D port configuration register
RD: Read signal
WRxx: Write signal

4.2.4 Port 3

Port 3 is an 8-bit 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 P37 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

Input to the P31 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer using port input mode register 3 (PIM3).

Output from the P30 and P31 pins can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance) in 1-bit units, using port output mode register (POM).

These pins also function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

To use P30 to P37 as the port function, refer Table 4-5.

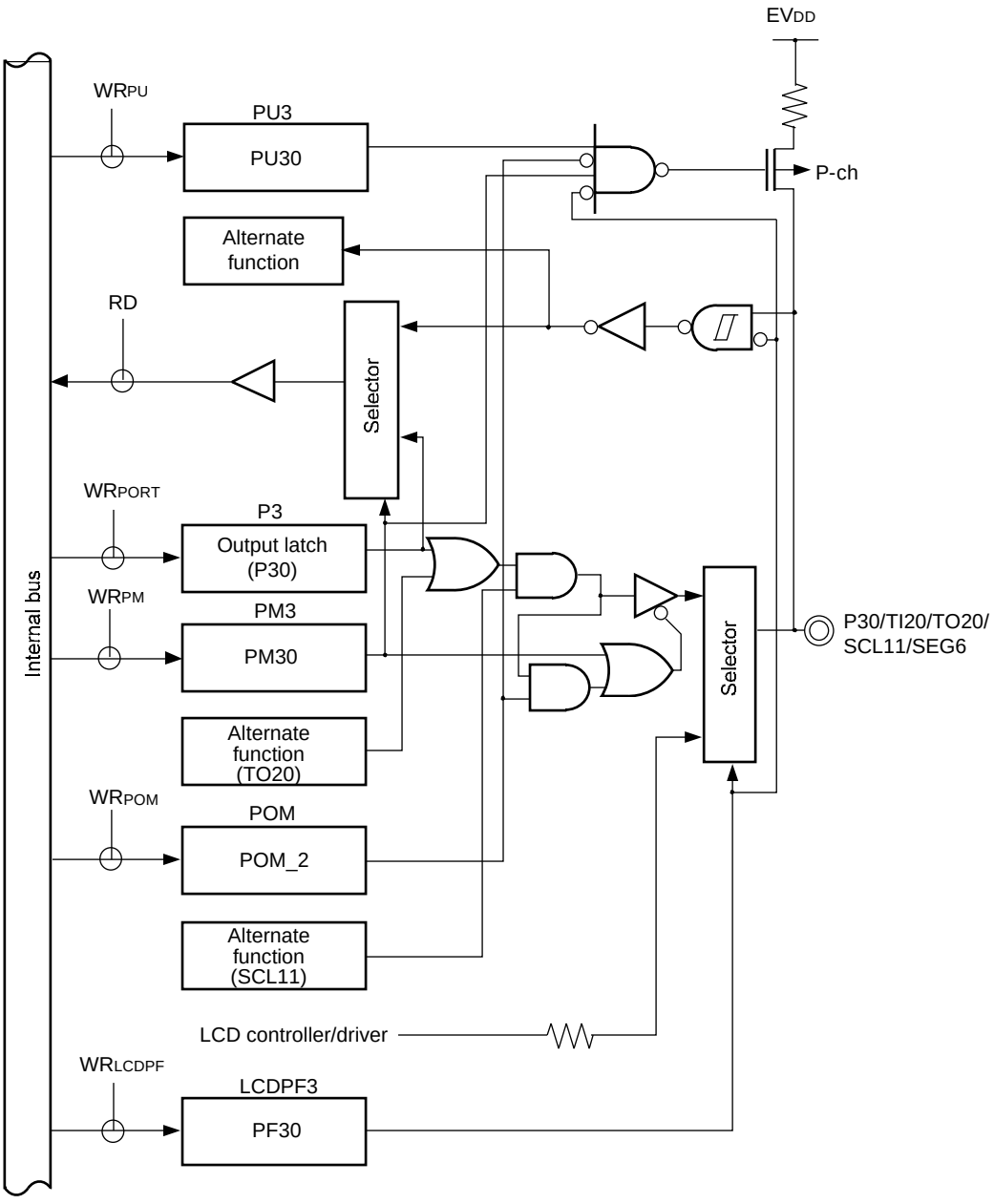
Table 4-5. Setting of P30 to P37 Pins to port function

P30 to P37 Pins		LCDPF3	Alternate function		PM3	PIM3	POM	Remarks
port	function	Register	Timer	Serial	Register	Register	Register	
P30	Input port	Digital I/O selection	-	-	Input mode	N/A	-	
	Output port		0	1	Output mode		0	CMOS output
							1	N-ch OD output
P31	Input port	Digital I/O selection	-	-	Input mode	0		Schmitt1 input
						1		Schmitt3 input
	Output port		0	1	Output mode	-	0	CMOS output
							1	N-ch OD output
P32	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P33	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P34	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P35	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P36	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P37	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 3 to input mode.

Figures 4-13 to 4-16 show block diagrams of port3.

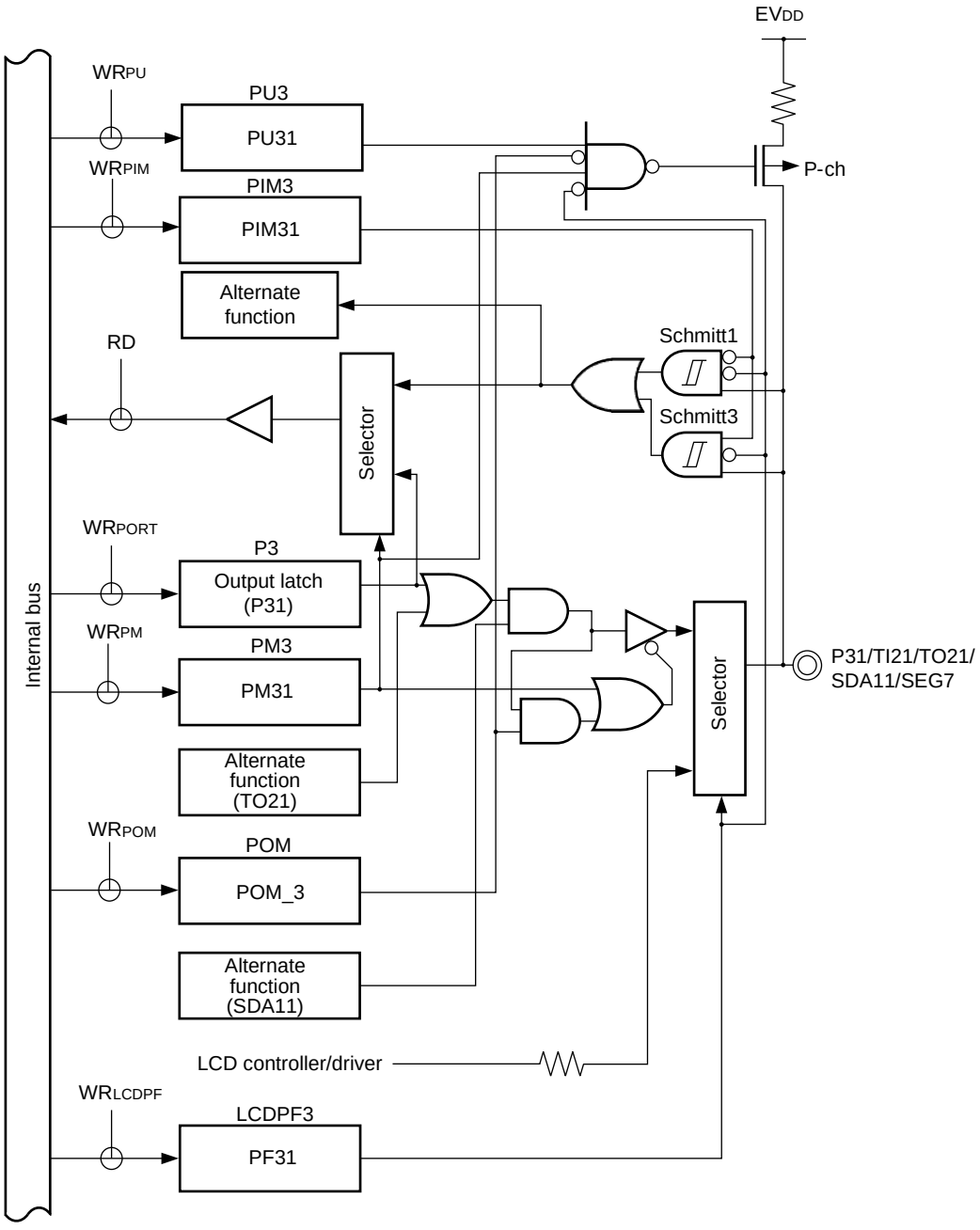
Figure 4-13. Block Diagram of P30



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
LCDPF3: LCD port function registers 3
POM: Port Output mode register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO20, set the port latch to 0.
When using the alternate function SCL11, set the port latch to 1.
When using P30 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

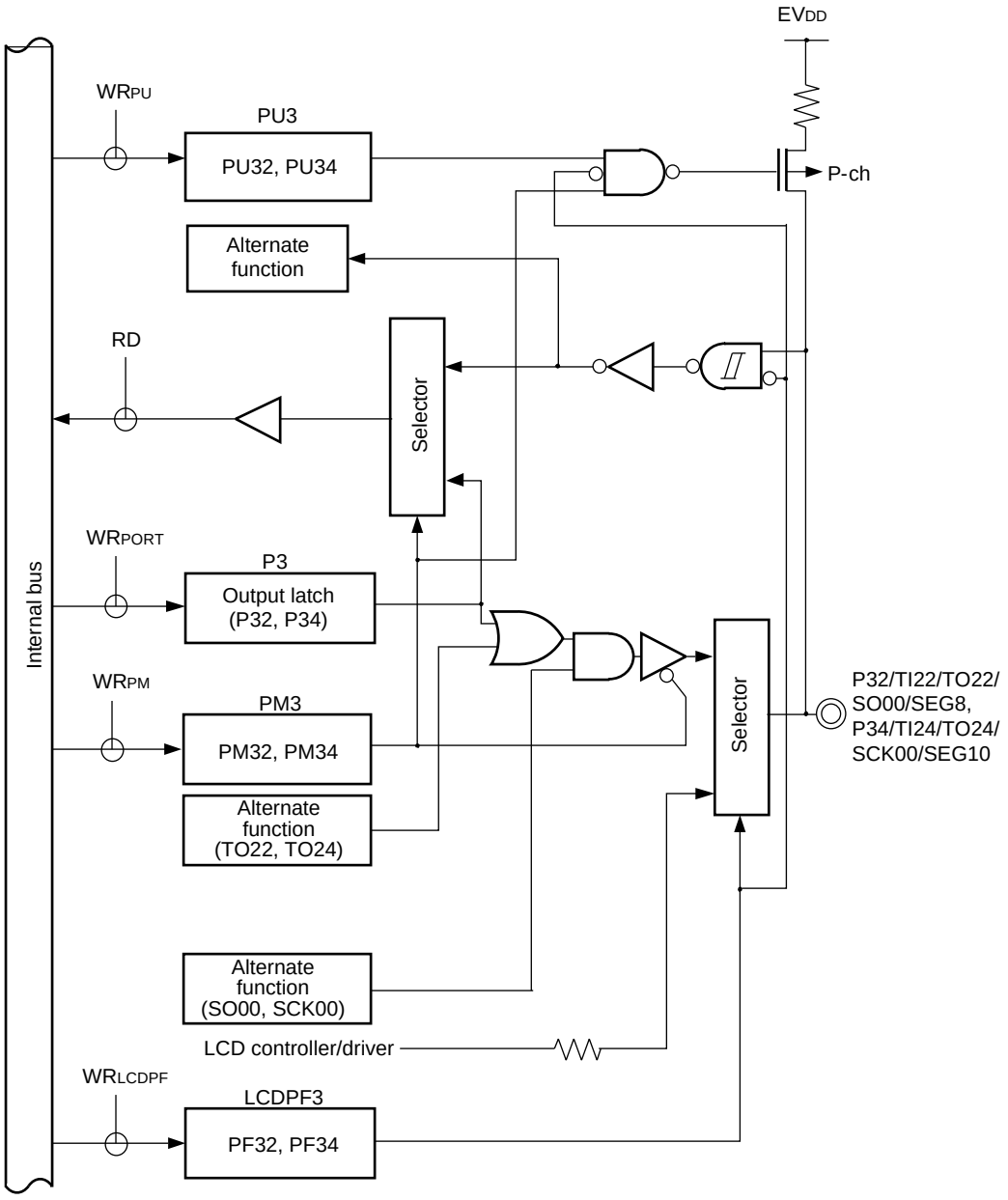
Figure 4-14. Block Diagram of P31



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
PIM3: Port Input mode register 3
LCDPF3: LCD port function registers 3
POM: Port Output mode register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO21, set the port latch to 0.
When using the alternate function SDA11, set the port latch to 1.
When using P31 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

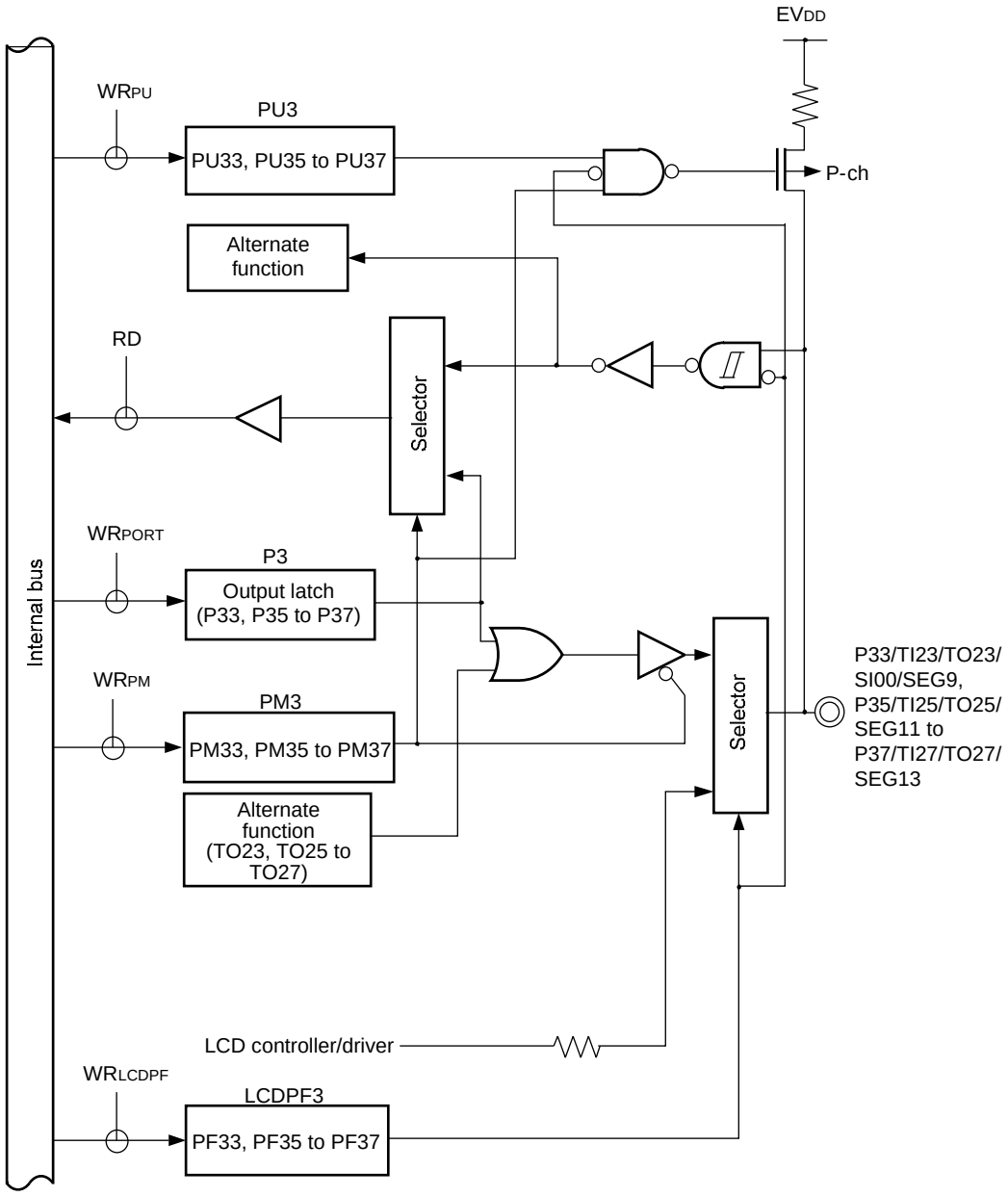
Figure 4-15. Block Diagram of P32, P34



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
LCDPF3: LCD port function registers 3
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO22 or TO24, set the port latch to 0.
When using the alternate function SO00 or SCK00, set the port latch to 1.
When using P32 or P34 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-16. Block Diagram of P33, P35 to P37



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
LCDPF3: LCD port function registers 3
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO23 or TO25 to TO27, set the port latch to 0.
When using P33 or P35 to P37 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

<R> **4.2.5 Port 4**

48-pin products: P40 function as a 1-bit I/O.
64-pin products: P40 function as a 1-bit I/O.
80-pin products: P40 function as a 1-bit I/O.
100-pin products: P40 function as a 1-bit I/O.
128-pin products: P40 to P47 function as an 8-bit I/O port

Port 4 is a 1-bit or a 8-bit 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 pin is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

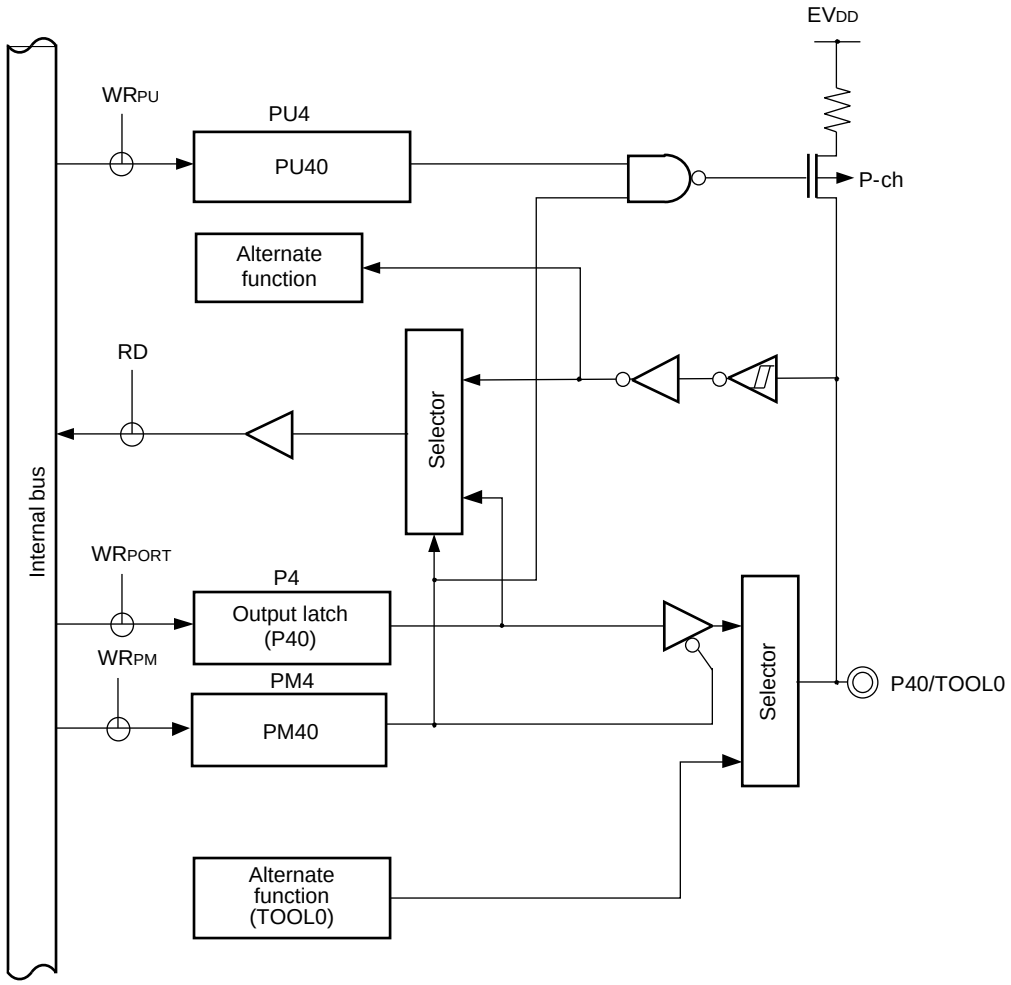
To use P40 to P47 as the port function, refer Table 4-6.

Table 4-6. Setting of P40 to P47 Pins to port function

P40 to P47 Pins		LCDPF4	Alternate function		PM4	PIM4	POM	Remarks
port	function	Register	Timer	Serial	Register	Register	Register	
P40	Input port	Digital I/O selection	N/A	N/A	Input mode	N/A	N/A	
	Output port				Output mode			
P41	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P42	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P43	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P44	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P45	Input port	Digital I/O selection	N/A	N/A	Input mode	N/A	N/A	
	Output port				Output mode			
P46	Input port	Digital I/O selection	N/A	N/A	Input mode	N/A	N/A	
	Output port				Output mode			
P47	Input port	Digital I/O selection	N/A	N/A	Input mode	N/A	N/A	
	Output port				Output mode			
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 0 to input mode.
Figures 4-17 to 4-20 show block diagrams of port 4.

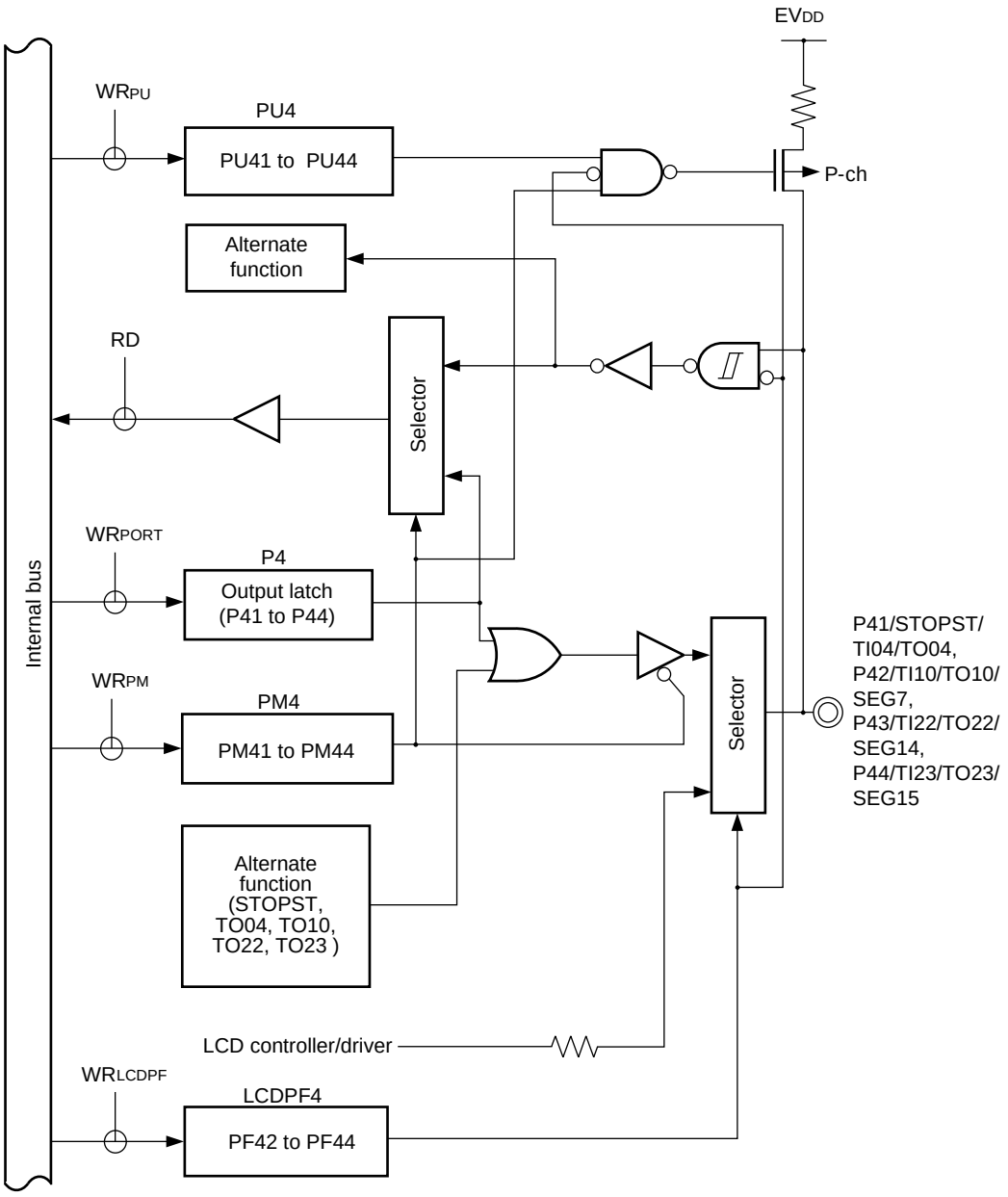
Figure 4-17. Block Diagram of P40



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

<R>

Figure 4-18. Block Diagram of P41-P44

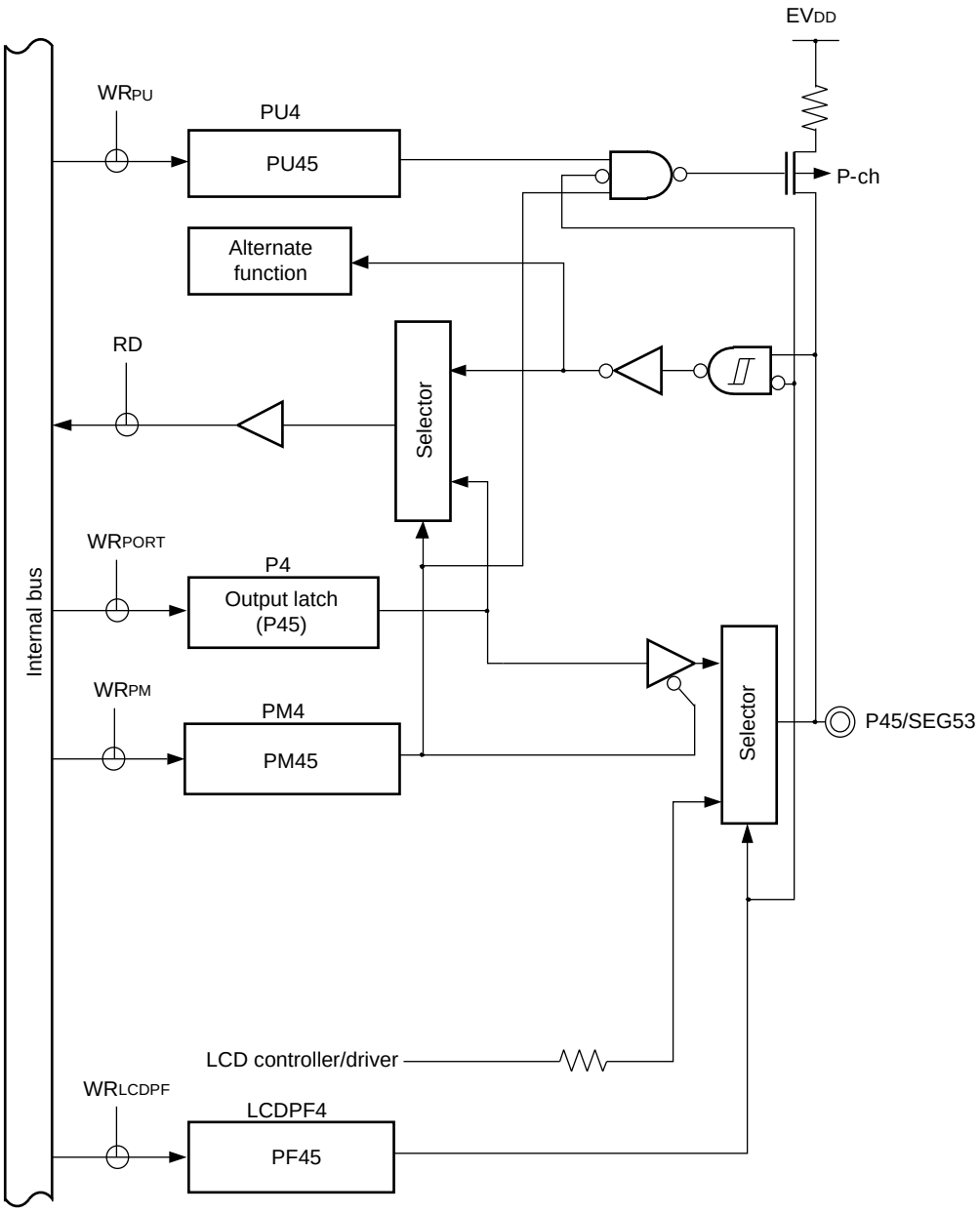


P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
LCDPF4: LCD port function registers 4
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO04, TO10, TO22, TO23, set the port latch to 0.
When using P41 to P44 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed to 0.

<R>

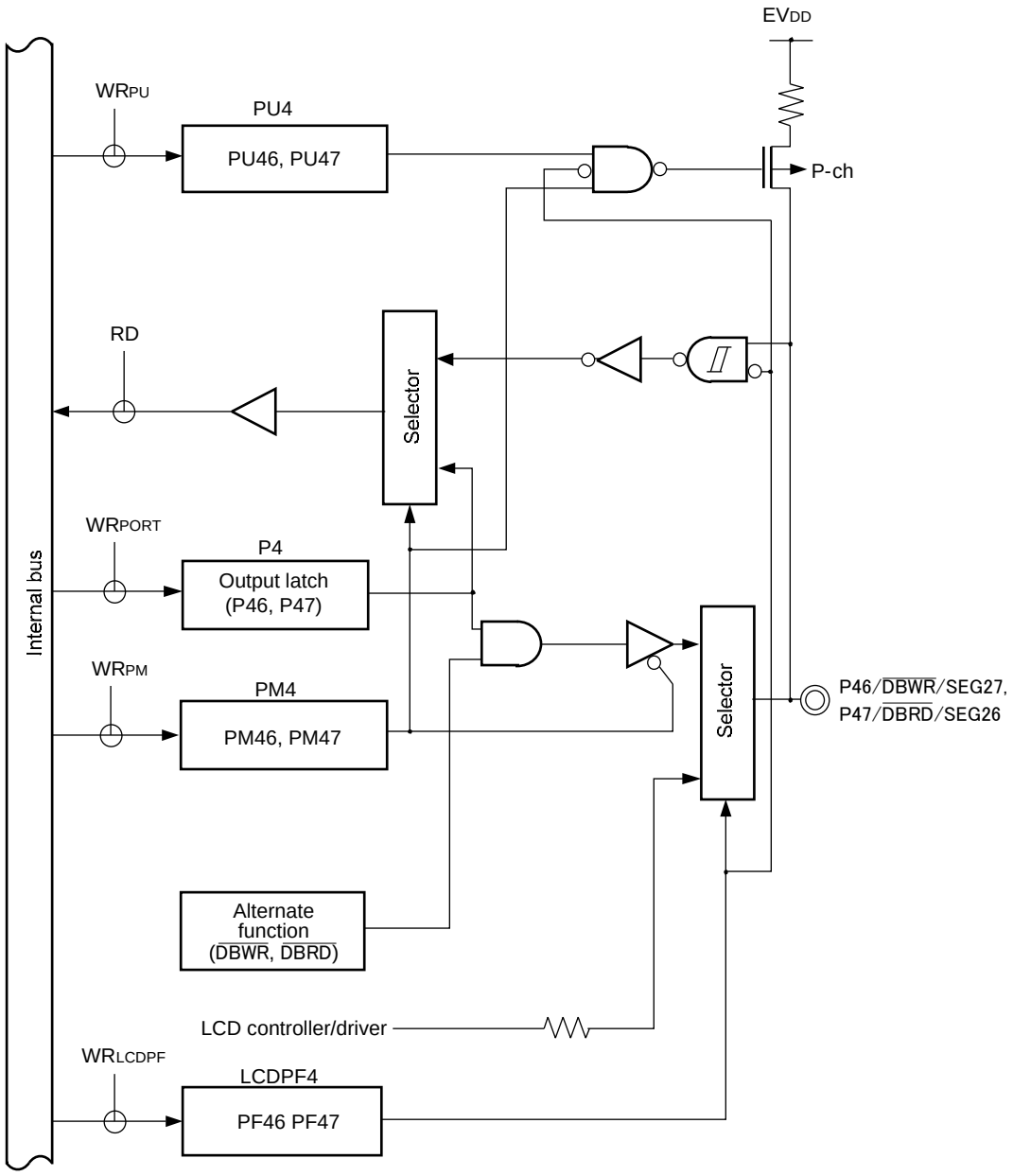
Figure 4-19. Block Diagram of P45



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
LCDPF4: LCD port function registers 4
RD: Read signal
WRxx: Write signal

<R>

Figure 4-20. Block Diagram of P46, P47



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
LCDPF4: LCD port function registers 4
RD: Read signal
WRxx: Write signal

Caution When using the alternate function $\overline{\text{DBWR}}$ or $\overline{\text{DBRD}}$, set the port latch to 1.
When using P46 or P47 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed to 1.

4.2.6 Port 5

48-pin products: P54 to P57 function as a 4-bit I/O port.
64-pin products: P54 to P57 function as a 4-bit I/O port.
80-pin products: P54 to P57 function as a 4-bit I/O port.
100-pin products: P50 to P57 function as an 8-bit I/O port.
<R> 128-pin products: P50 to P57 function as an 8-bit I/O port.

Port 5 is an 8-bit or a 4-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Input to the P50 to P52 and P55 to P57 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer using port input mode register 5 (PIM5).

Output from the P50 pin can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance), using port output mode register (POM).

These pins also function as timer I/O, serial interface data I/O, and segment signal outputs for the LCD controller/driver.

To use P50 to P57 as the port function, refer Table 4-7.

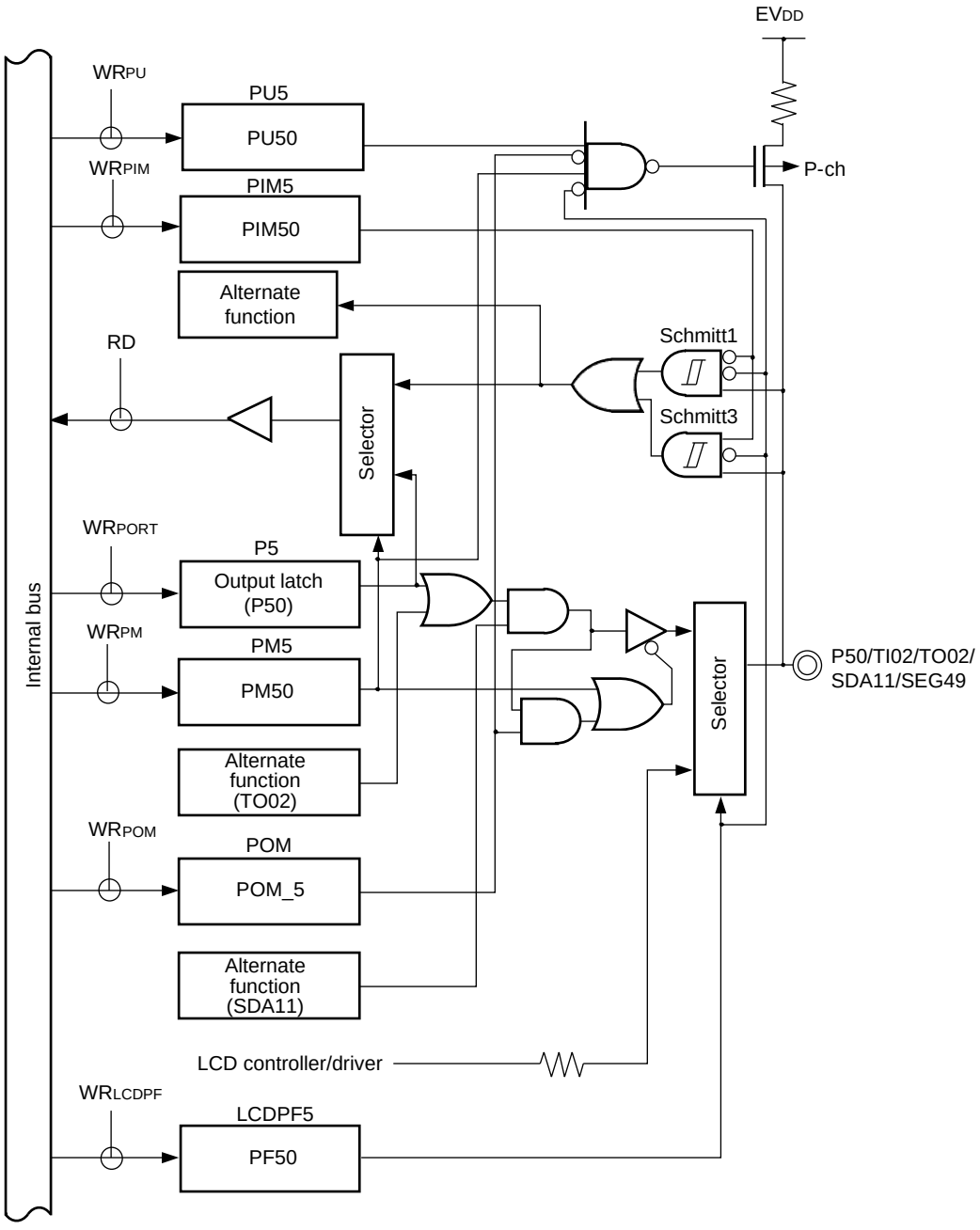
Table 4-7. Setting of P50 to P57 Pins to port function

P50 to P57 Pins		LCDPF5	Alternate function		PM5	PIM5	POM	Remarks
port	function	Register	Timer	Serial	Register	Register	Register	
P50	Input port	Digital I/O selection	-	-	Input mode	0	-	Schmitt1 input
						1		Schmitt3 input
	Output port		0	1	Output mode	-	0	CMOS output
							1	N-ch OD output
P51	Input port	Digital I/O selection	-	-	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0	1	Output mode	-		
P52	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
P53	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P54	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P55	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
P56	Input port	Digital I/O selection	-	-	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0	1	Output mode	-		
P57	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 5 to input mode.

Figures 4-21 to 4-24 show block diagrams of port5.

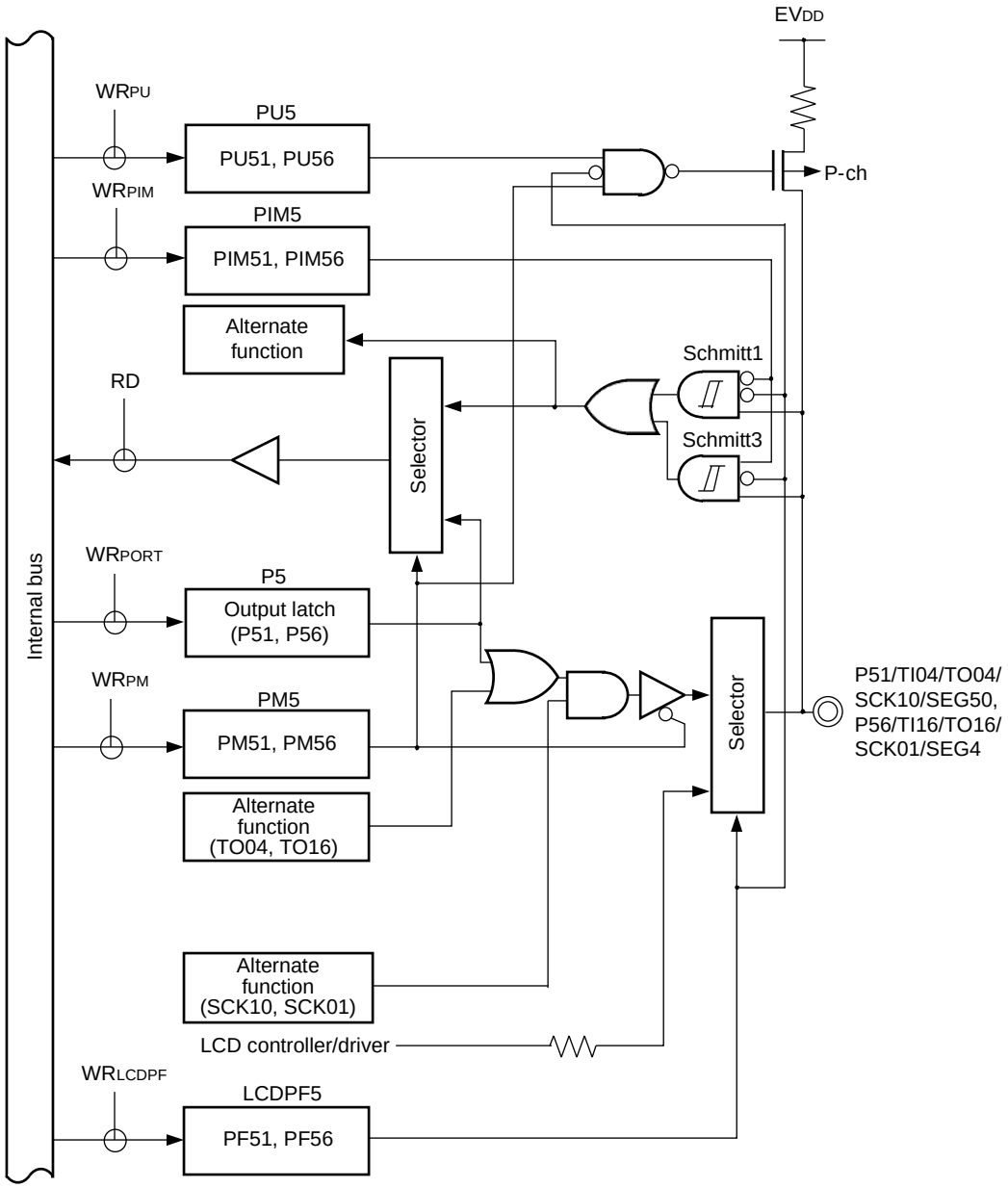
Figure 4-21. Block Diagram of P50



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIM5: Port Input mode register 5
LCDPF5: LCD port function registers 5
POM: Port Output mode register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO02, set the port latch to 0.
When using the alternate function SDA11, set the port latch to 1.
When using P50 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

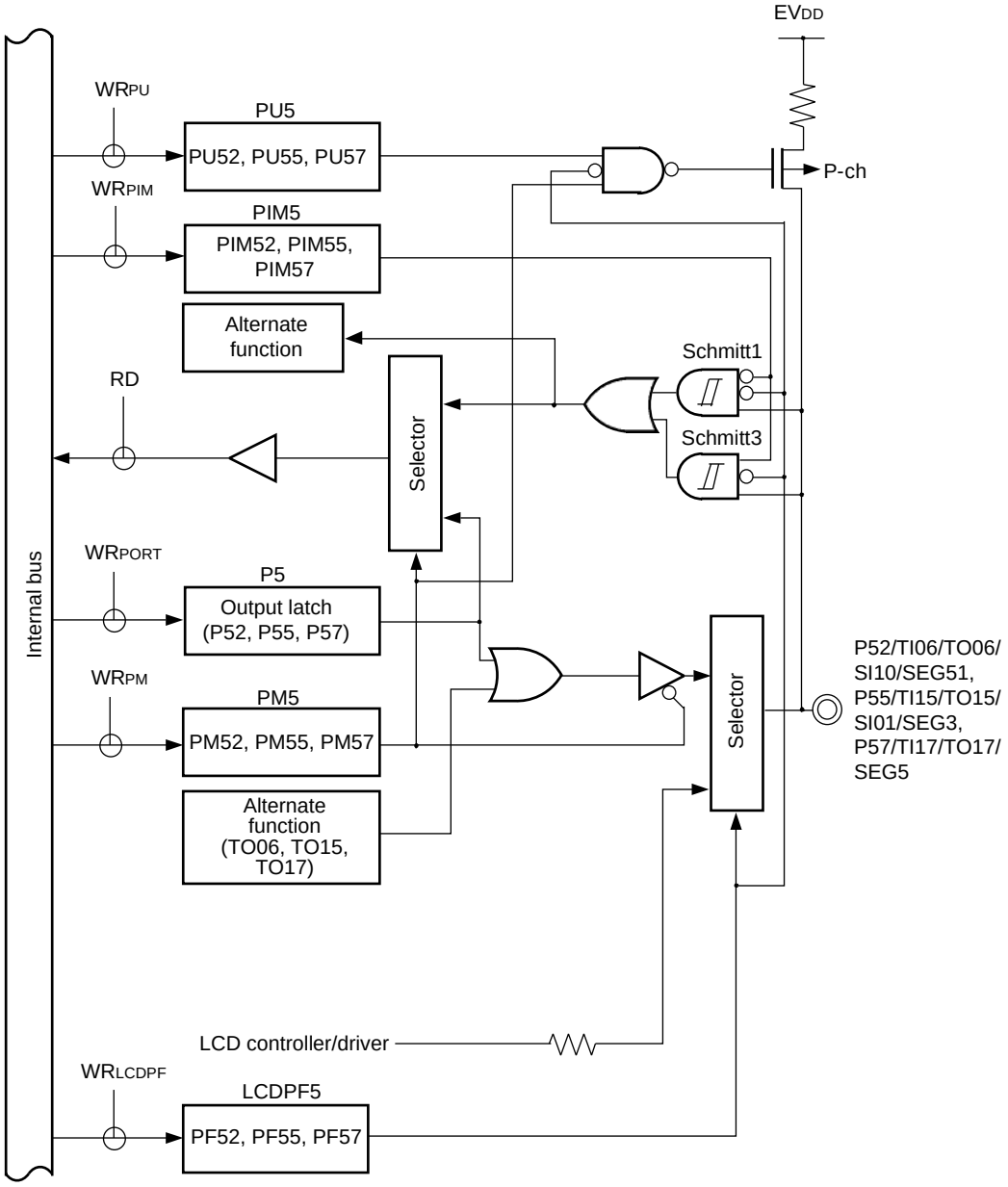
Figure 4-22. Block Diagram of P51, P56



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIM5: Port Input mode register 5
LCDPF5: LCD port function registers 5
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO04 or TO16 , set the port latch to 0.
When using the alternate function SCK10 or SCK01, set the port latch to 1.
When using P51 or P56 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-23. Block Diagram of P52, P55, P57



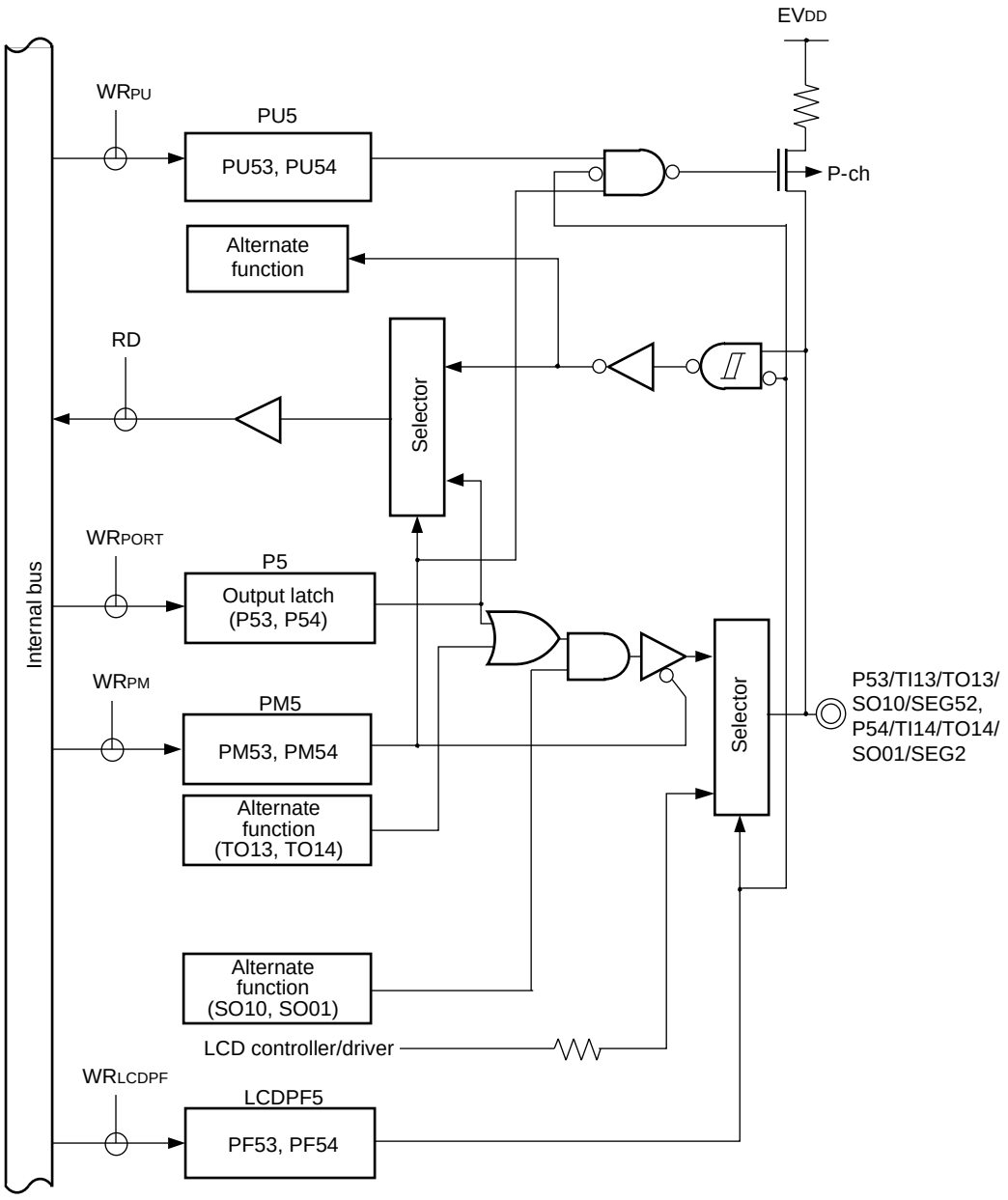
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIM5: Port Input mode register 5
LCDPF5: LCD port function registers 5
RD: Read signal
WRxx: Write signal

Caution

When using the alternate function TO06, TO15, or TO17, set the port latch to 0.

When using P52, P55, or P57 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

Figure 4-24. Block Diagram of P53, P54



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
LCDPF5: LCD port function registers 5
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO13 or TO14 , set the port latch to 0.
When using the alternate function SO10 or SO01, set the port latch to 1.
When using P53 or P54 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

4.2.7 Port 6

48-pin products: P60 and P61 function as a 2-bit I/O port.
64-pin products: P60 and P61 function as a 2-bit I/O port.
80-pin products: P60, 61, 65, and P66 function as a 4-bit I/O port.
100-pin products: P60 to P66 function as a 7-bit I/O port.
<R> 128-pin products: P60 to P66 function as a 7-bit I/O port.

Port 6 is a 7-bit, a 4-bit, or 2-bit 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 P66 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).

Input to the P61 and P63 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer using port input mode register 5 (PIM5).

Output from the P60 and P61 pins can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance), using port output mode register (POM).

These pins also function as serial interface data I/O, timer I/O, real-time clock correction clock output, clock / buzzer output, and external interrupt request input.

To use P60 to P66 as the port function, refer Table 4-8.

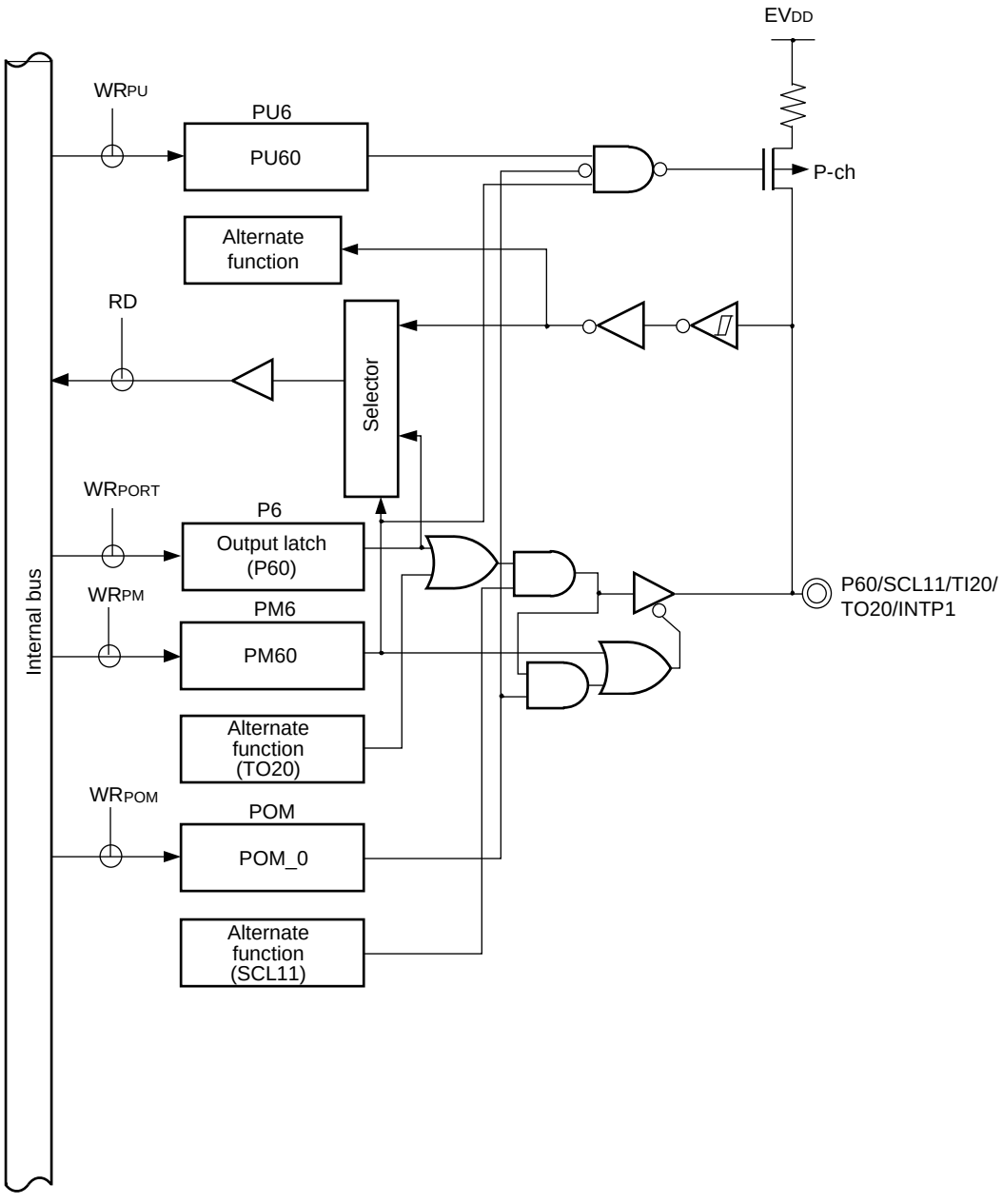
Table 4-8. Setting of P60 to P66 Pins to port function

P60 to P66 Pins		Alternate function		PM6	PIM6	POM	Remarks
port	function	Timer	Serial	Register	Register	Register	
P60	Input port	-	-	Input mode	N/A	-	
	Output port	0	1	Output mode		0	CMOS output
						1	N-ch OD output
P61	Input port	-	-	Input mode	0	-	Schmitt1 input
					1		Schmitt3 input
	Output port	0	1	Output mode	-	0	CMOS output
						1	N-ch OD output
P62	Input port	-	-	Input mode	N/A	N/A	
	Output port	0	1	Output mode			
P63	Input port	-	N/A	Input mode	0	N/A	Schmitt1 input
					1		Schmitt3 input
	Output port	0		Output mode	-		
P64	Input port	-	N/A	Input mode	N/A	N/A	
	Output port	0		Output mode			
P65	Input port	-	N/A	Input mode	N/A	N/A	
	Output port	0		Output mode			
P66	Input port	-	N/A	Input mode	N/A	N/A	
	Output port	0		Output mode			

Reset signal generation sets port 6 to input mode.

Figures 4-25 to 4-30 show block diagrams of port6.

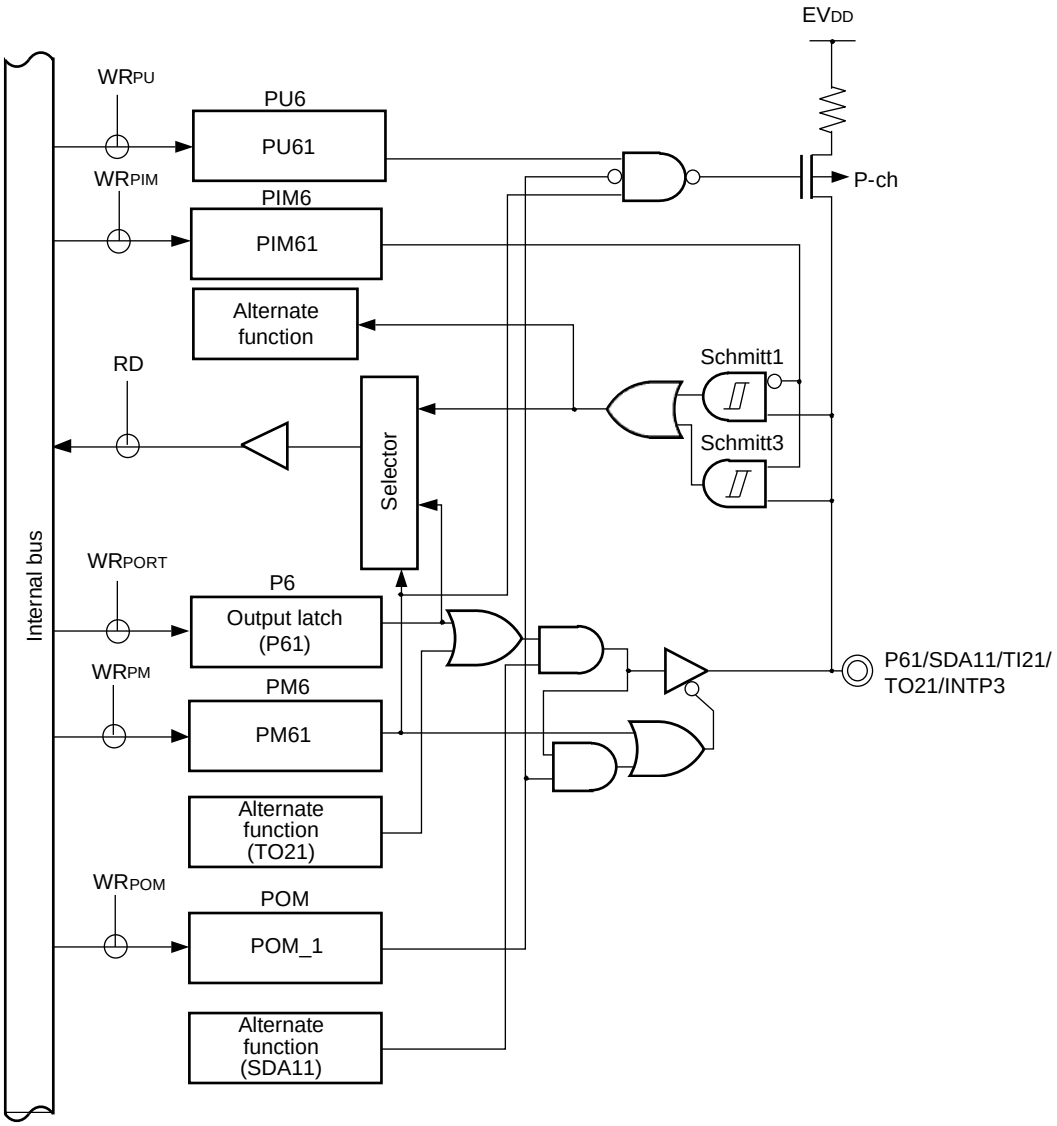
Figure 4-25. Block Diagram of P60



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
POM: Port Output mode register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO20, set the port latch to 0.
When using the alternate function SCL11, set the port latch to 1.
When using P60 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

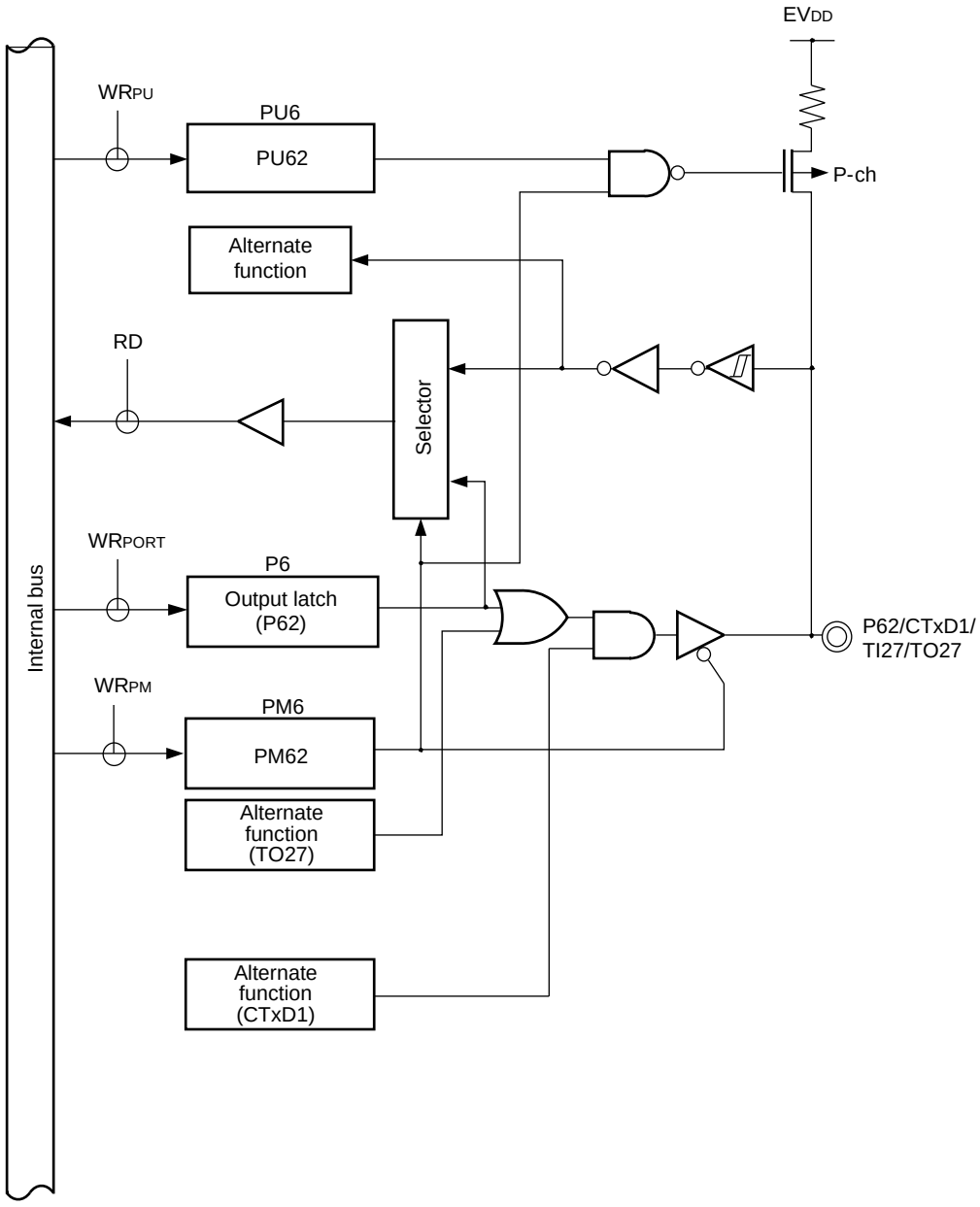
Figure 4-26. Block Diagram of P61



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PIM6: Port Input mode register 6
POM: Port Output mode register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function T TO21, set the port latch to 0.
When using the alternate function SDA11, set the port latch to 1.
When using P61 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

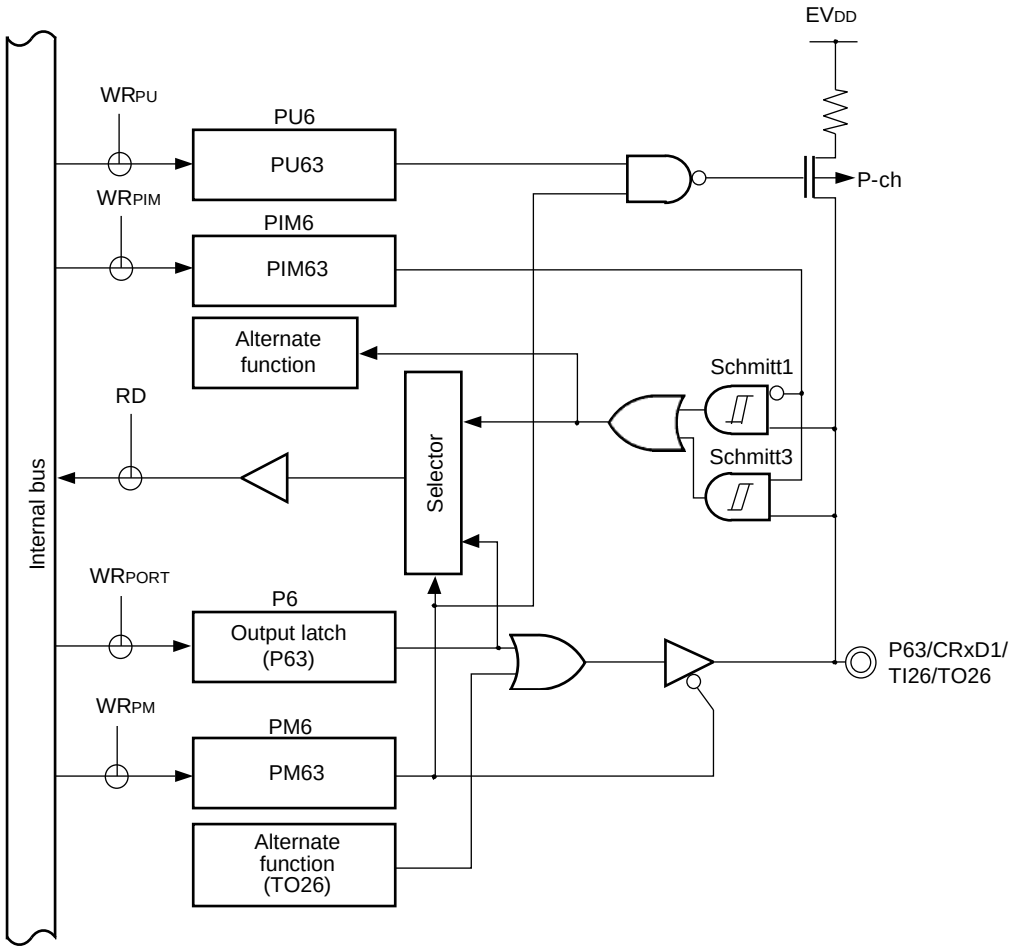
Figure 4-27. Block Diagram of P62



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO27, set the port latch to 0.
When using the alternate function CTxD1, set the port latch to 1.
When using P62 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

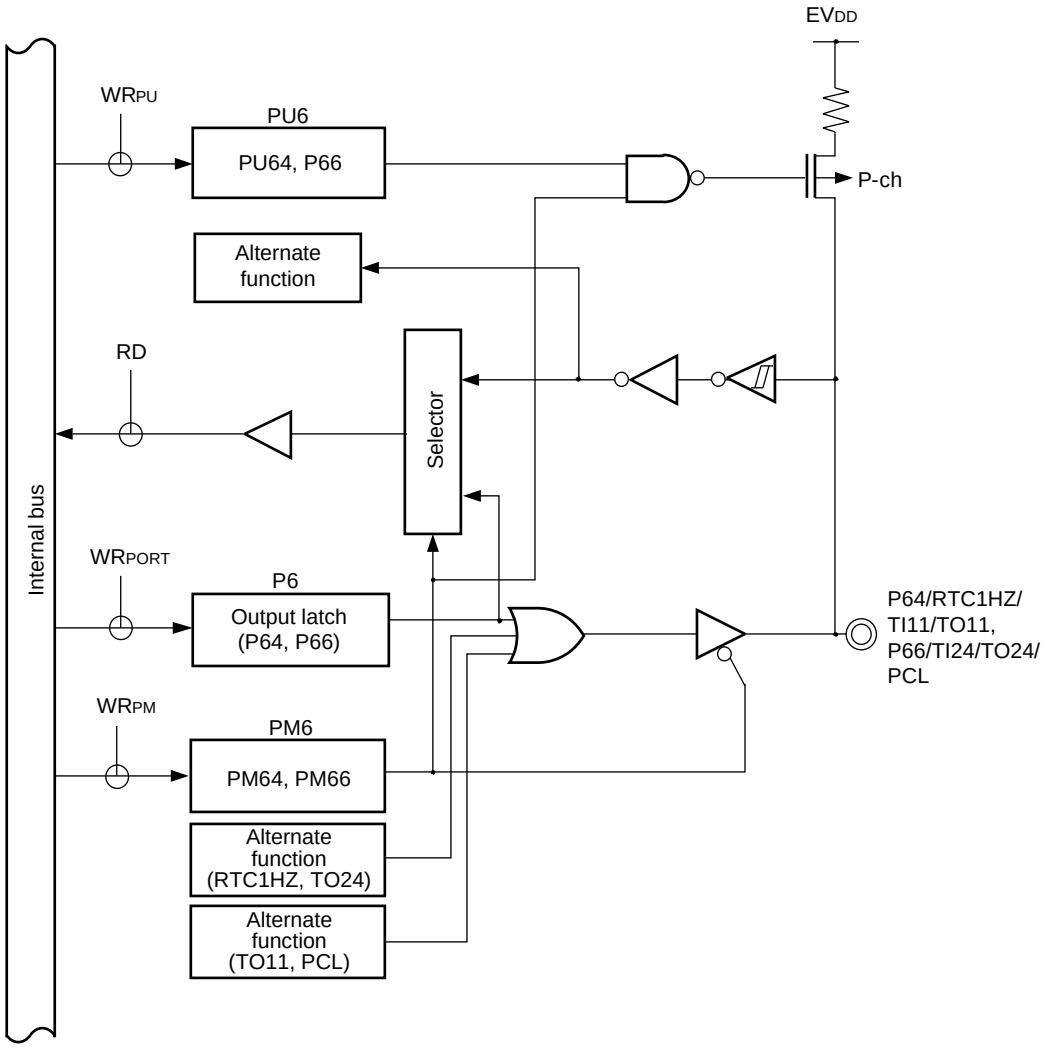
Figure 4-28. Block Diagram of P63



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
WRxx: Write signal

Caution When using the alternate function TO26, set the port latch to 0.
When using P63 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

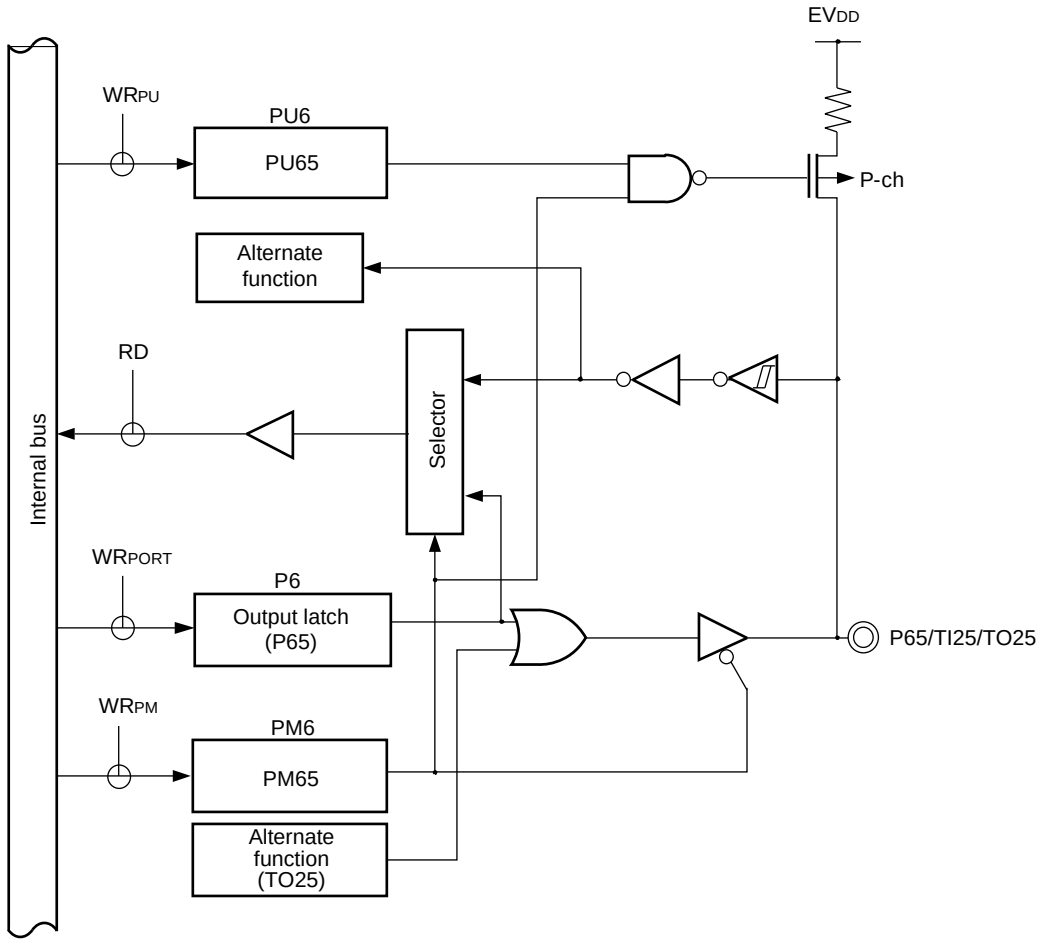
Figure 4-29. Block Diagram of P64, P66



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WRxx: Write signal

Caution When using the alternate function RTC1HZ, TO24, TO11, or PCL, set the port latch to 0.
When using P64 or P66 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

Figure 4-30. Block Diagram of P65



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO25, set the port latch to 0.
When using P65 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

4.2.8 Port 7

Port 7 is a 6-bit 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 P75 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 7 (PU7).

Input to the P70 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer using port input mode register 7 (PIM7).

These pins also function as A/D conversion start trigger input, output pins for the sound generator, serial interface data I/O, timer I/O, clock / buzzer output, flash memory programming I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

To use P70 to P75 as the port function, refer Table 4-9.

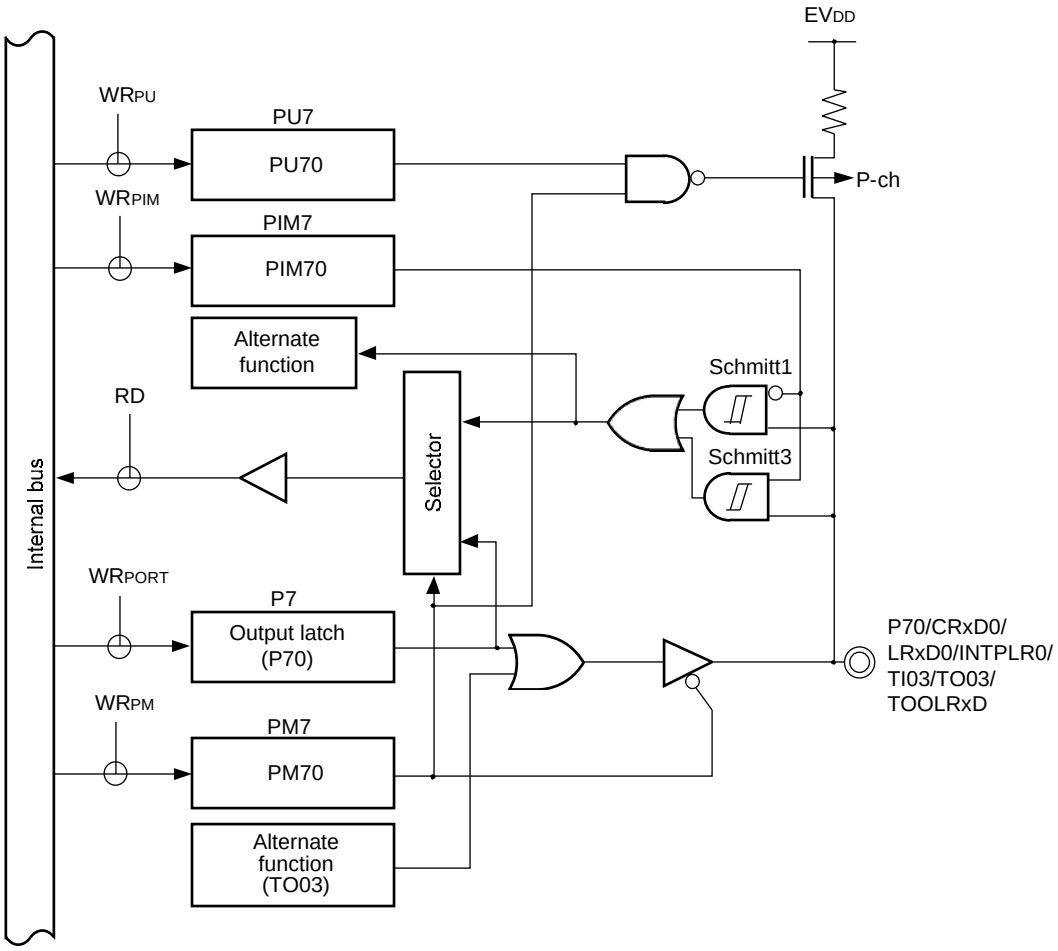
Table 4-9. Setting of P70 to P75 Pins to port function

P70 to P75 Pins		LCDPF7 Register	Alternate function		PM7 Register	PIM7 Register	POM Register	Remarks
port	function		Timer/ SGO	Serial				
P70	Input port	N/A	-	N/A	Input mode	0	N/A	Schmitt1 input
						1		Schmitt3 input
	Output port		0		Output mode	-		
P71	Input port	N/A	N/A	-	Input mode	N/A	N/A	
	Output port			1	Output mode			
P72	Input port	Digital I/O selection	-	N/A	Input mode		N/A	
	Output port		0		Output mode	-		
P73	Input port	Digital I/O selection	-	N/A	Input mode	-	N/A	
	Output port		0		Output mode			
P74	Input port	Digital I/O selection	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P75	Input port	Digital I/O selection	-	N/A	Input mode	-	N/A	
	Output port		0		Output mode			
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 7 to input mode.

Figures 4-31 to 4-35 show block diagrams of port 7.

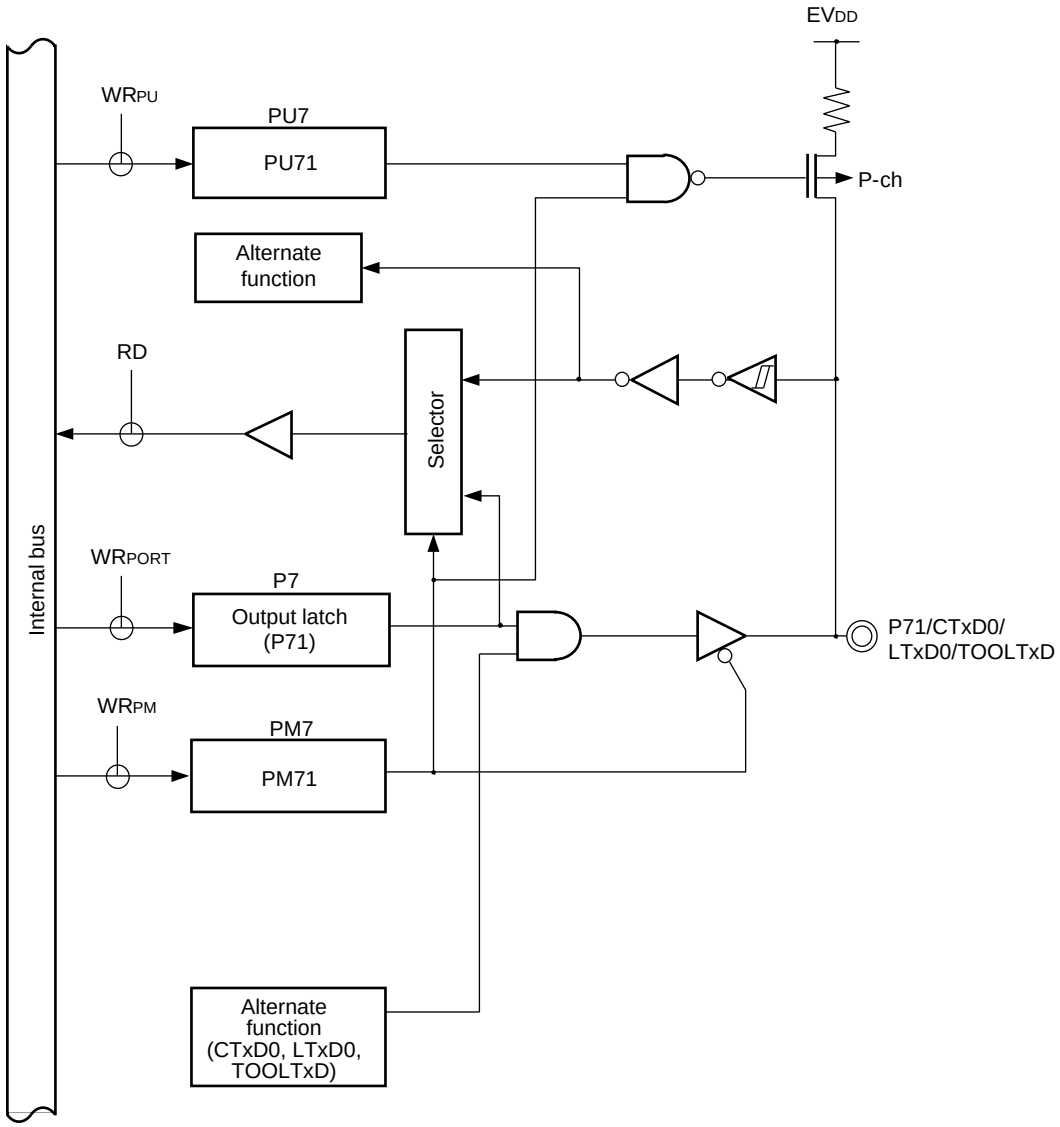
Figure 4-31. Block Diagram of P70



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
WRxx: Write signal

Caution When using the alternate function TO03, set the port latch to 0.
When using P70 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

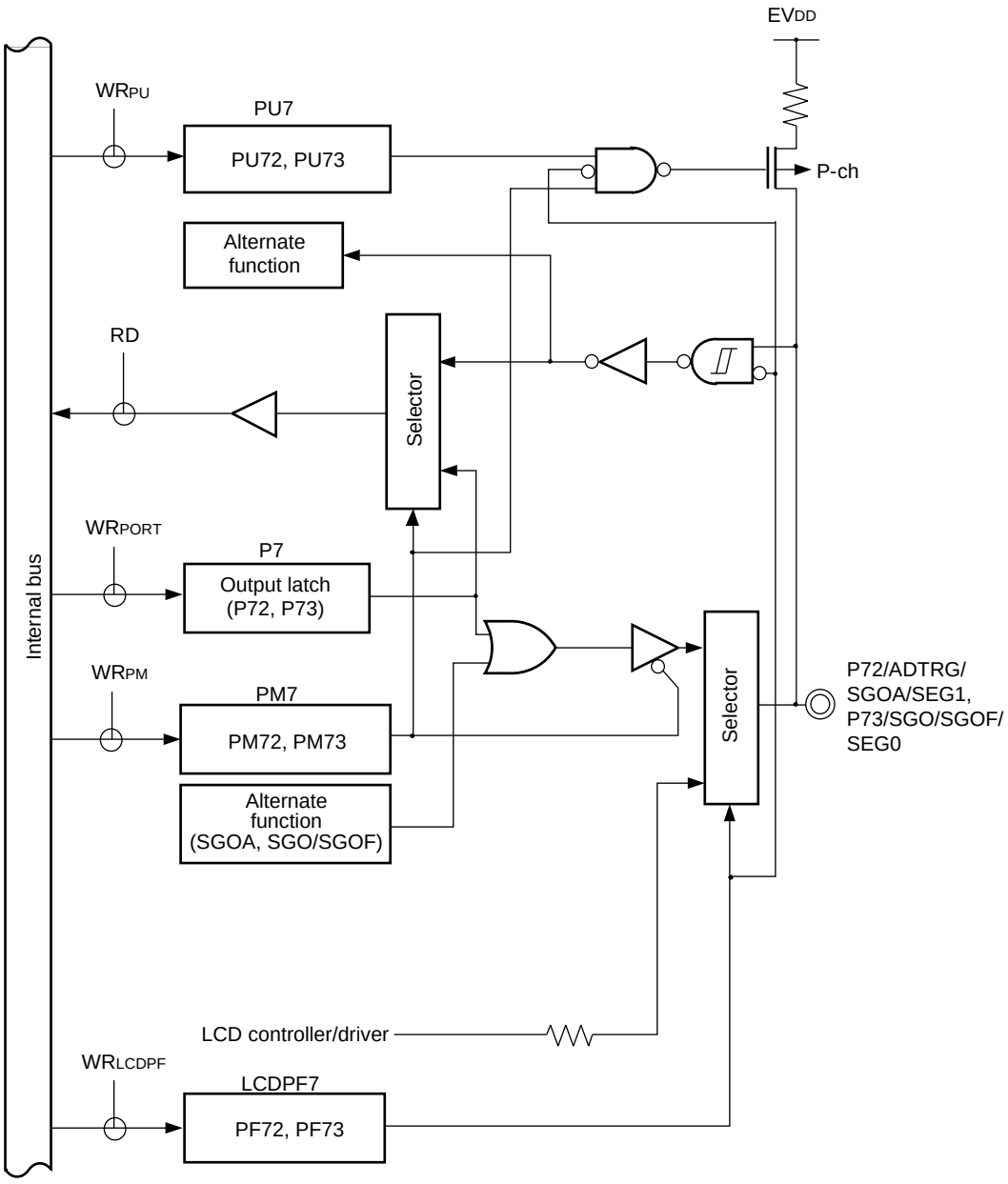
Figure 4-32. Block Diagram of P71



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

Caution When using the alternate function CTxD0, LTxD0, TOOLTxD, set the port latch to1.
When using P71 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed to 1.

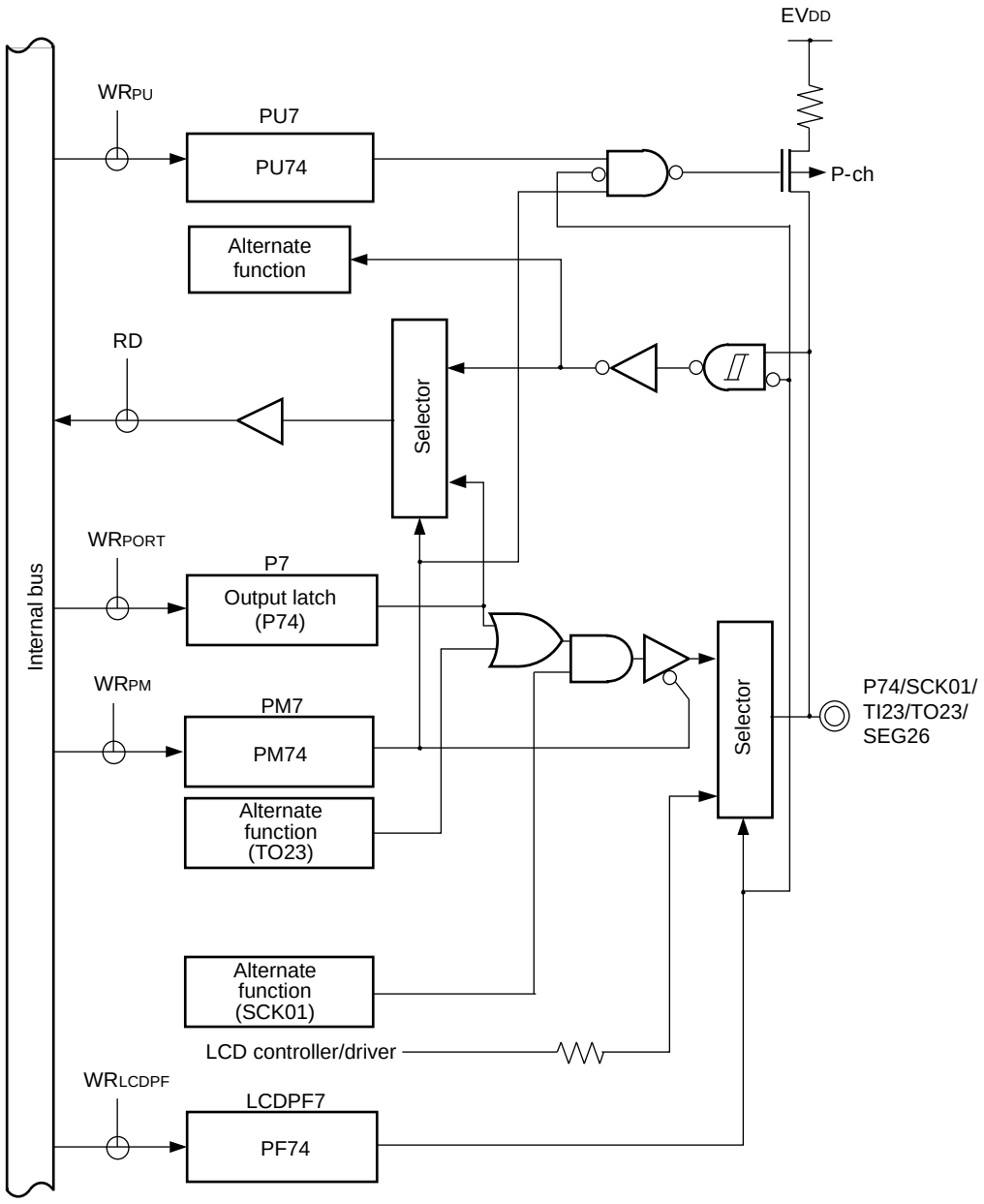
Figure 4-33. Block Diagram of P72, P73



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
LCDPF7: LCD port function registers 7
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SGOA, SGO/SGOF, or TO22, set the port latch to 0.
When using P72 or P73 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

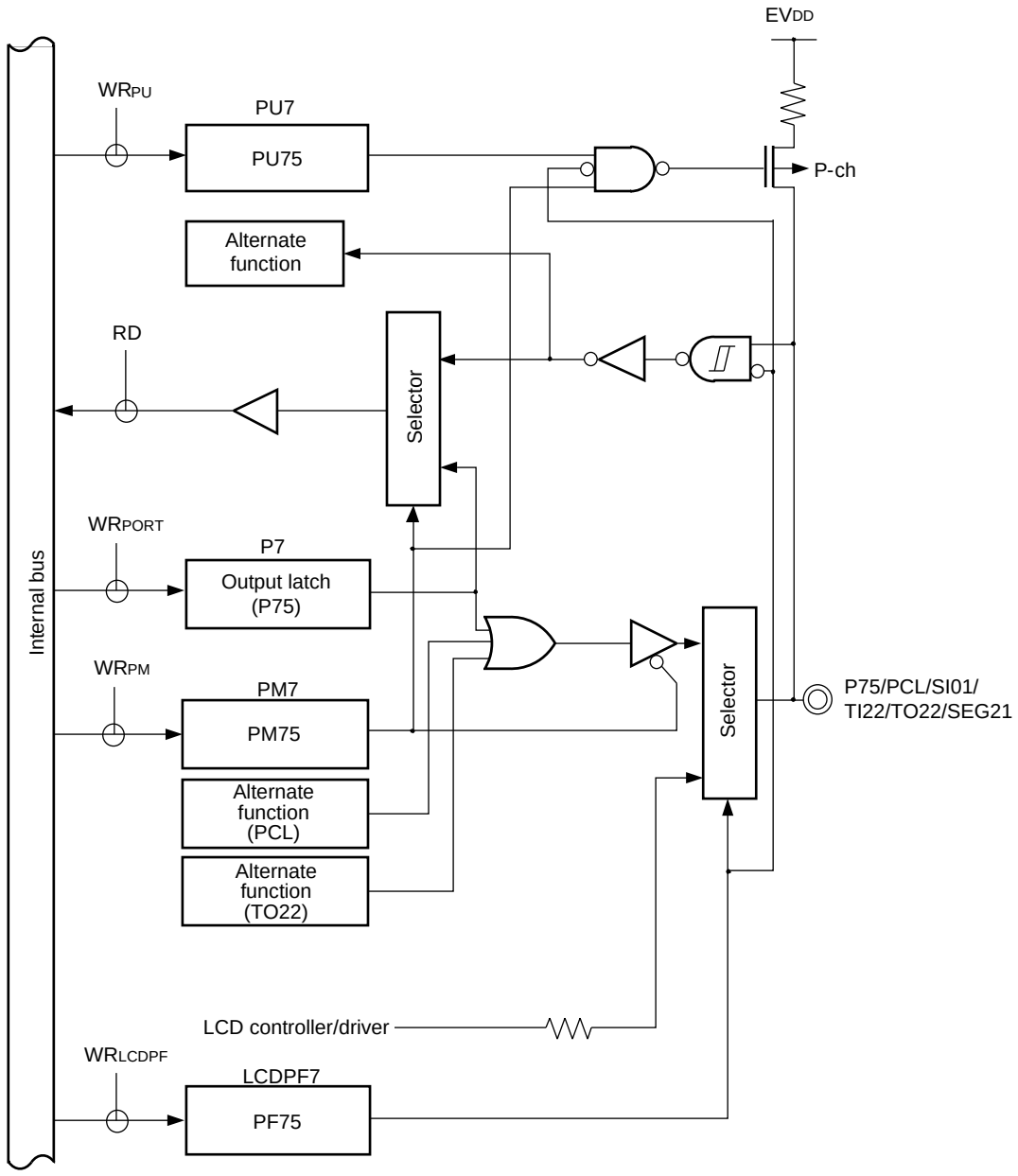
Figure 4-34. Block Diagram of P74



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
LCDPF7: LCD port function registers 7
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO23, set the port latch to 0.
When using the alternate function SCK01, set the port latch to 1.
When using P74 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-35. Block Diagram of P75



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
LCDPF7: LCD port function registers 7
RD: Read signal
WRxx: Write signal

Caution When using the alternate function PCL or TO22, set the port latch to 0.
When using P75 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

4.2.9 Port 8

Port 8 is an 8-bit I/O port with an output latch and LED direct drive capability. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P87 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 8(PU8)

These pins also function as timer I/O, stepper motor controller/driver outputs/inputs, and segment signal outputs for the LCD controller/driver.

To use P80 to P87 as the port function, refer Table 4-10.

Table 4-10. Setting of P80 to P87 Pins to port function

P80 to P87 Pins		LCDPF8 Register	Alternate function				PM8 Register	Remarks
port	function		Timer	Serial	SMPC Register	ZPDS0 Register		
P80	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note1		Output mode	
P81	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note2		Output mode	
P82	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note3		Output mode	
P83	Input port	Digital I/O selection	-	N/A	-	ZPD1PC=0	Input mode	
	Output port		0		Port mode Note4		Output mode	
P84	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note1		Output mode	
P85	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note2		Output mode	
P86	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note3		Output mode	
P87	Input port	Digital I/O selection	-	N/A	-	ZPD2PC=0	Input mode	
	Output port		0		Port mode Note4		Output mode	
-	-	LCD Segment output selection	-		-	-	-	LCD segment output

Note1. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1110 or 1111

2. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1100 or 1101

3. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1101 or 1110

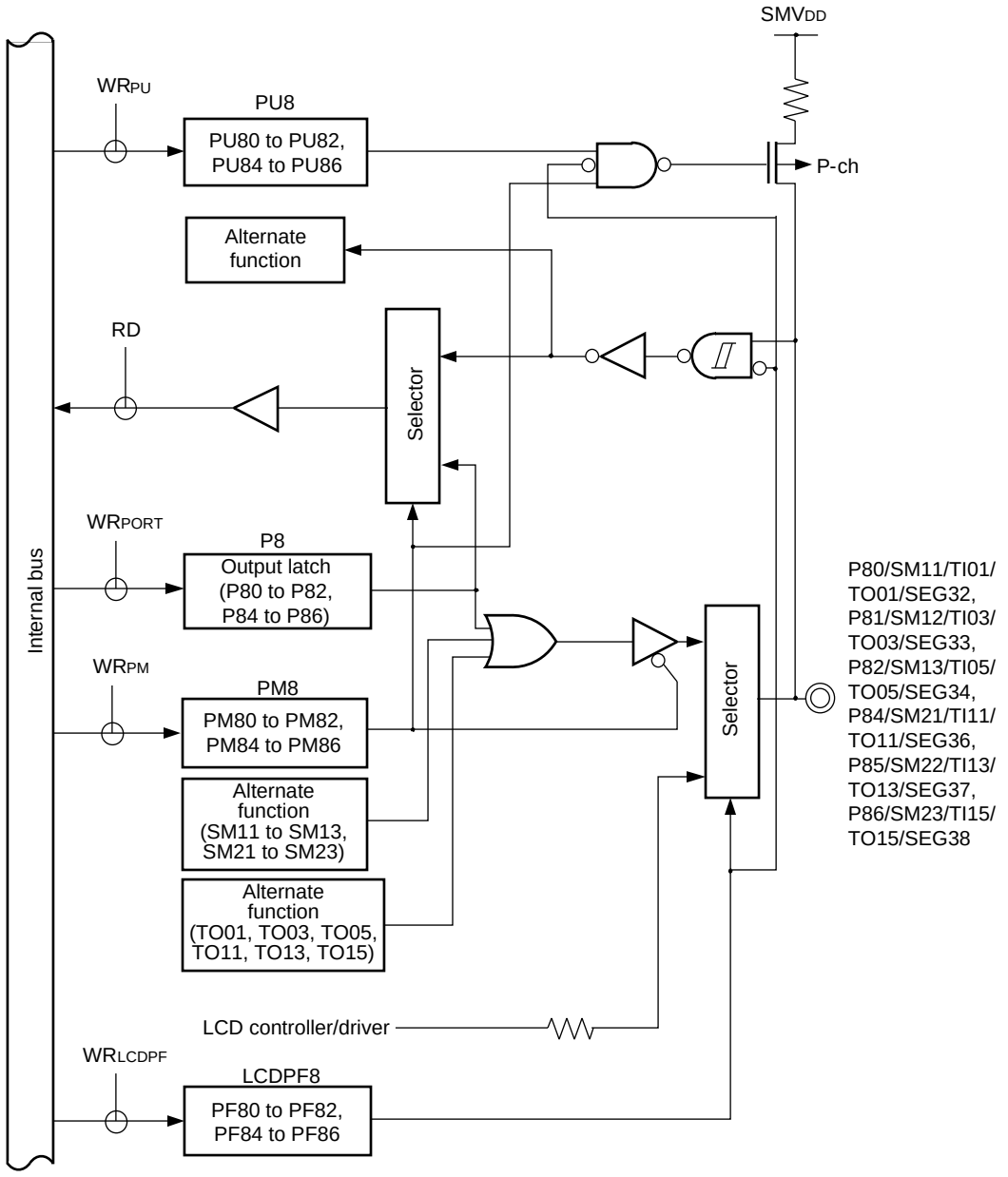
4. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1100 or 1111

(k = 0, 1)

Reset signal generation sets port 8 to input mode.

Figure 4-36 and 4-37 show block diagrams of port 8.

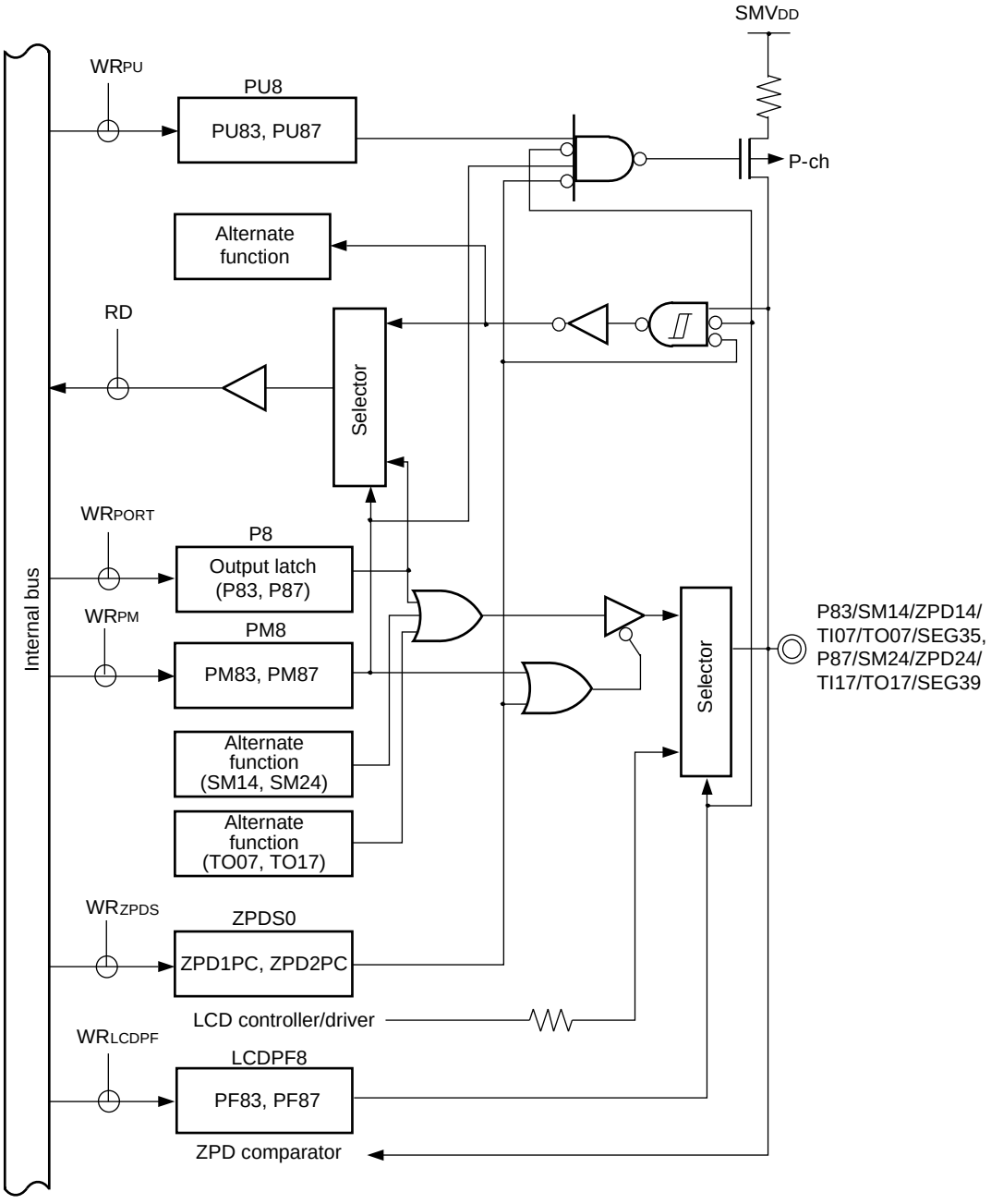
Figure 4-36. Block Diagram of P80 to P82 and P84 to P86



P8: Port register 8
PU8: Pull-up resistor option register 8
PM8: Port mode register 8
LCDPF8: LCD port function registers 8
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SM11 to SM13, SM21 to SM23, TO01, TO03, TO05, TO11, TO13, or TO15, set the port latch to 0.
When using P80 to P82 or P84 to P86 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

Figure 4-37. Block Diagram of P83, P87



P8: Port register 8
PU8: Pull-up resistor option register 8
PM8: Port mode register 8
LCDPF8: LCD port function registers 8
ZPDS0: ZPD detection voltage setting register0/ZPD analog input control register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SM14, SM18, TO07, or TO17 set the port latch to 0.
When using P83 or P87 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

4.2.10 Port 9

48-pin products: P90 to P94 function as a 5-bit I/O port.
64-pin products: P90 to P94 function as a 5-bit I/O port.
80-pin products: P90 to P97 function as an 8-bit I/O port.
100-pin products: P90 to P97 function as an 8-bit I/O port.
<R> 128-pin products: P90 to P97 function as an 8-bit I/O port.

Port 9 is an 8-bit or a 5-bit I/O port with an output latch and LED direct drive capability. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9). When the P90 to P97 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 9(PU9)

These pins also function as timer I/O, stepper motor controller/driver outputs/inputs, and segment signal outputs for the LCD controller/driver.

To use P90 to P97 as the port function, refer Table 4-11.

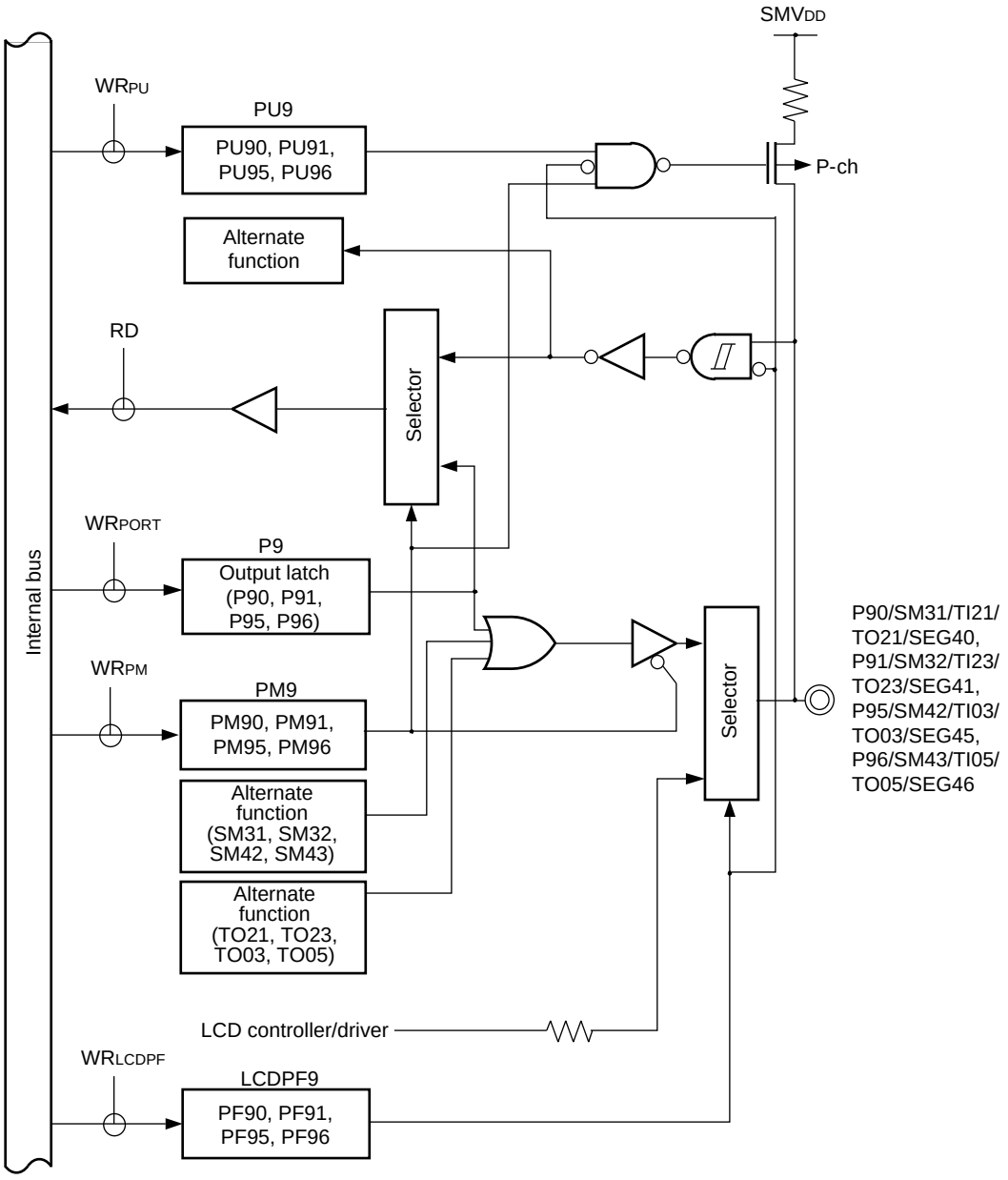
Table 4-11. Setting of P90 to P97 Pins to port function

P90 to P97 Pins		LCDPF9 Register	Alternate function				PM9 Register	Remarks
port	function		Timer /SGO	Serial	SMPC Register	ZPDS1 Register		
P90	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note1		Output mode	
P91	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note2		Output mode	
P92	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note3		Output mode	
P93	Input port	Digital I/O selection	-	N/A	-	ZPD3PC = 0	Input mode	
	Output port		0		Port mode Note4		Output mode	
P94	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note1		Output mode	
P95	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note2		Output mode	
P96	Input port	Digital I/O selection	-	N/A	-	N/A	Input mode	
	Output port		0		Port mode Note3		Output mode	
P97	Input port	Digital I/O selection	-	N/A	-	ZPD4PC = 0	Input mode	
	Output port		0		Port mode Note4		Output mode	
-	-	LCD segment output selection	-		-	-	-	LCD segment output

Note1. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1110 or 1111
2. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1100 or 1101
3. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1101 or 1110
4. ENk = 0 or (ENk, MODk, DIRk1, DIRk0)=1100 or 1111
(k = 0, 1)

Reset signal generation sets port 9 to input mode.
Figures 4-38 to 4-41show block diagrams of port 9.

Figure 4-38. Block Diagram of P90, P91, P95, and P96

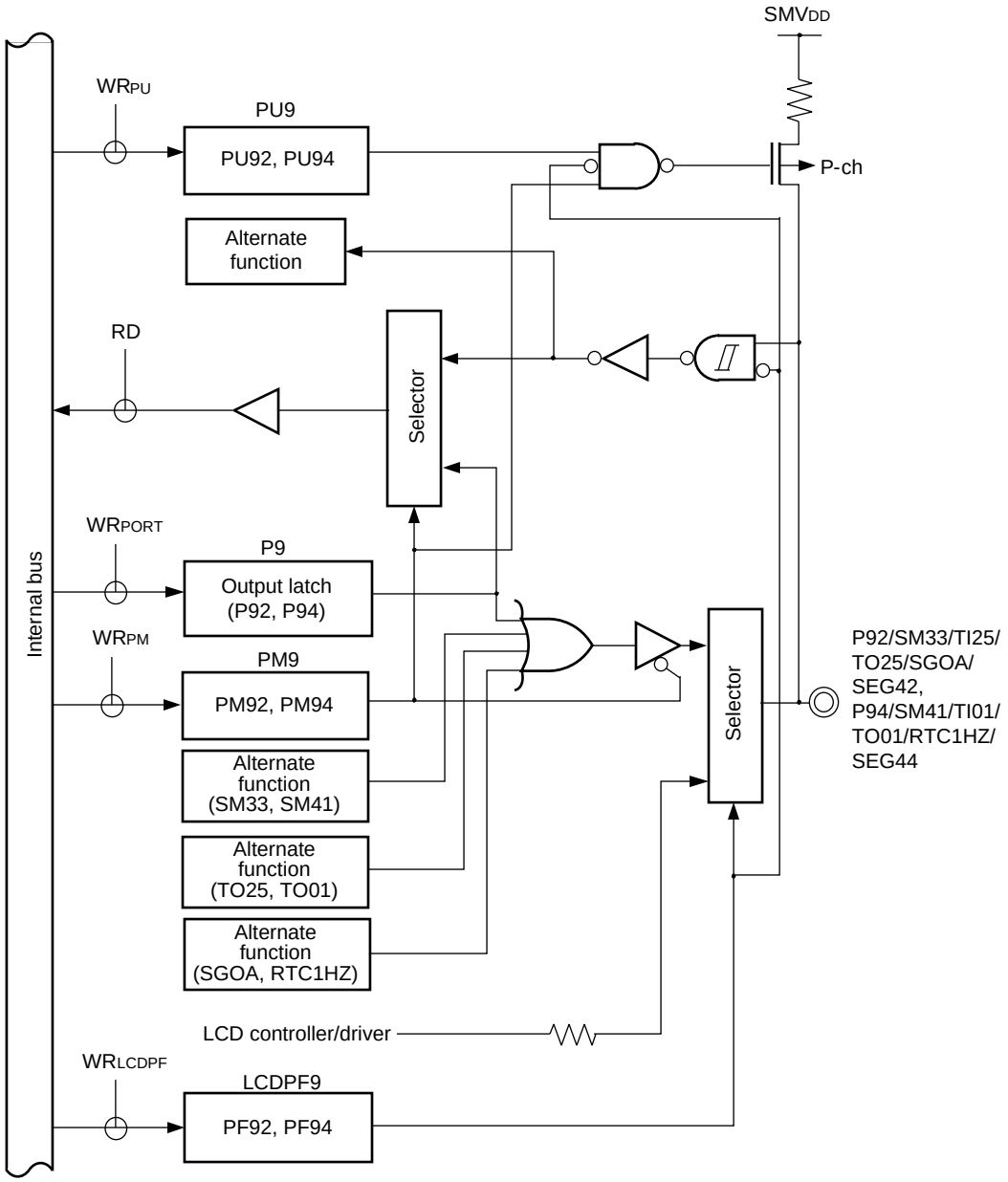


P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
LCDPF9: LCD port function registers 9
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SM31, SM32, SM42, SM43, TO21, TO23, TO03, or TO05, set the port latch to 0.

When using P90, P91, P95, or P96 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

Figure 4-39. Block Diagram of P92, P94

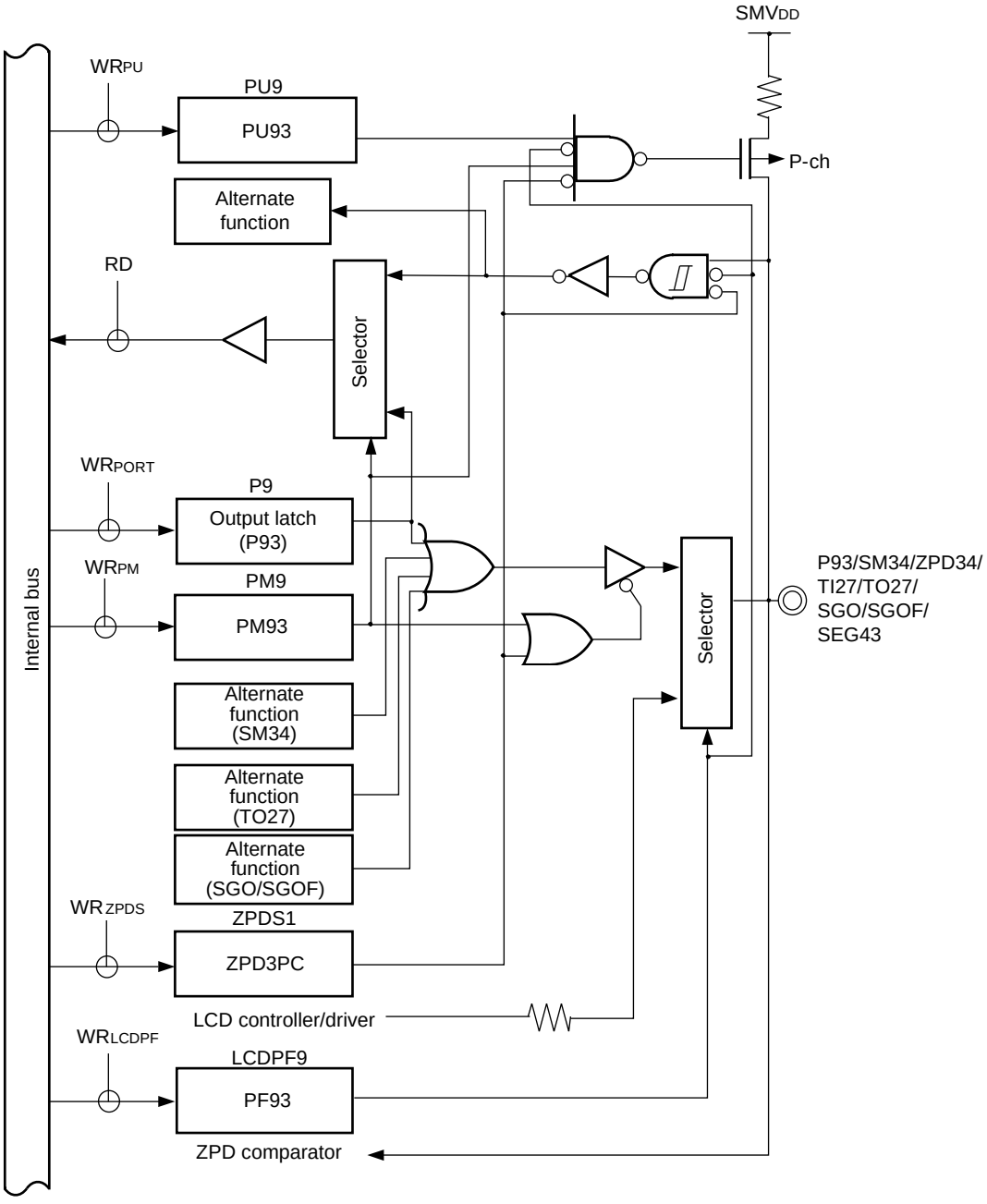


P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
LCDPF9: LCD port function registers 9
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SM33, SM41, TO25, TO01, SGOA, or RTC1HZ, set the port latch to 0.

When using P92 or P94 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

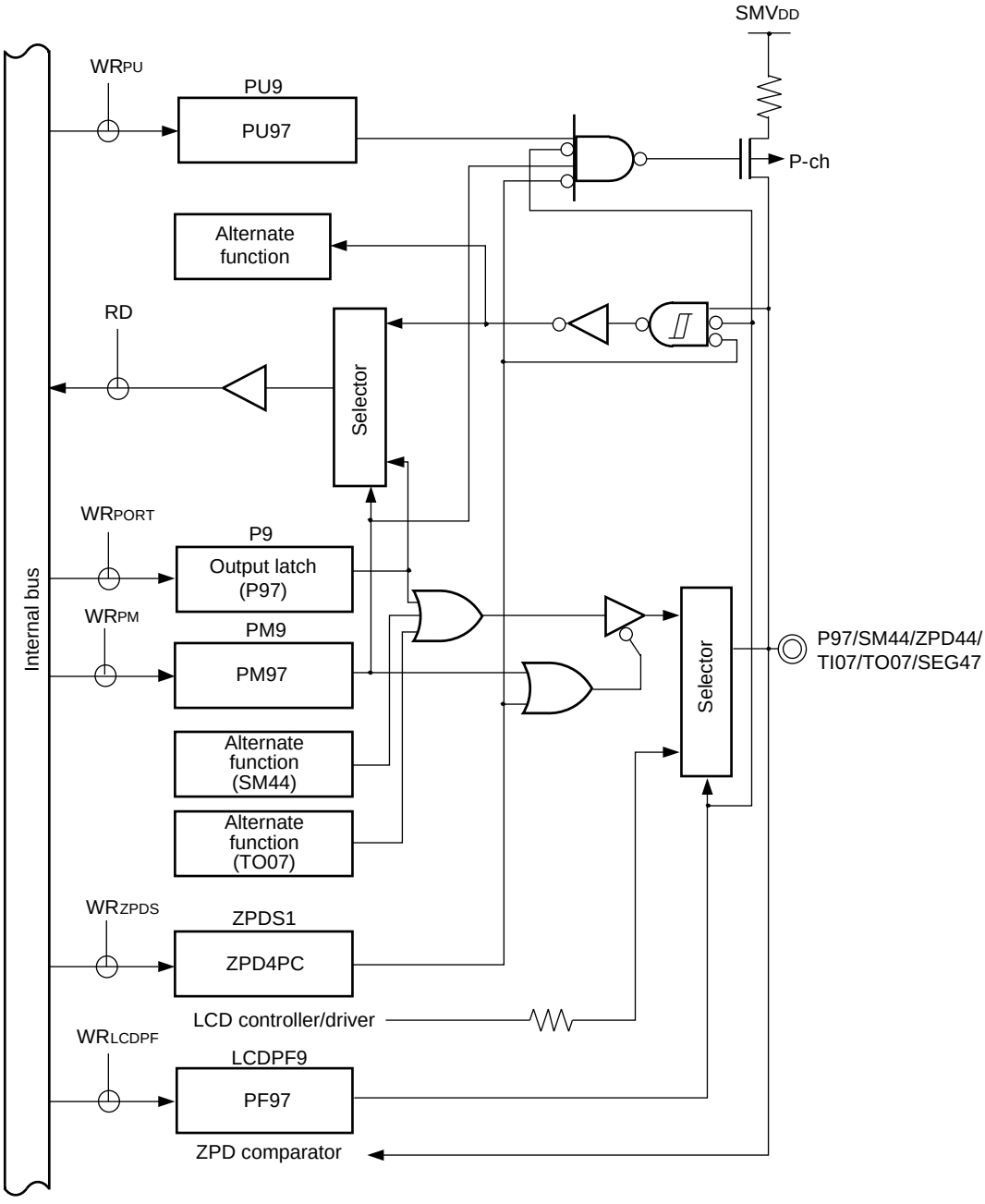
Figure 4-40. Block Diagram of P93



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
LCDPF9: LCD port function registers 9
ZPDS1: ZPD detection voltage setting register1/ZPD analog input control register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SM34, TO27 or SGO/SGOF, set the port latch to 0.
When using P93 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

Figure 4-41. Block Diagram of P97



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
LCDPF9: LCD port function registers 9
ZPDS1: ZPD detection voltage setting register1/ZPD analog input control register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SM44 or TO07 set the port latch to 0.
When using P97 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed 0.

<R> **4.2.11 Port 10**

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: Not provided
128-pin products: P100 to P107 function as an 8-bit I/O port.

Port 10 is an 8-bit I/O port with an output latch. Port 10 can be set to the input mode or output mode in 1-bit units using port mode register 10 (PM10). When the P100 to P107 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 10(PU10)

These pins also function as timer I/O and segment signal outputs for the LCD controller/driver.

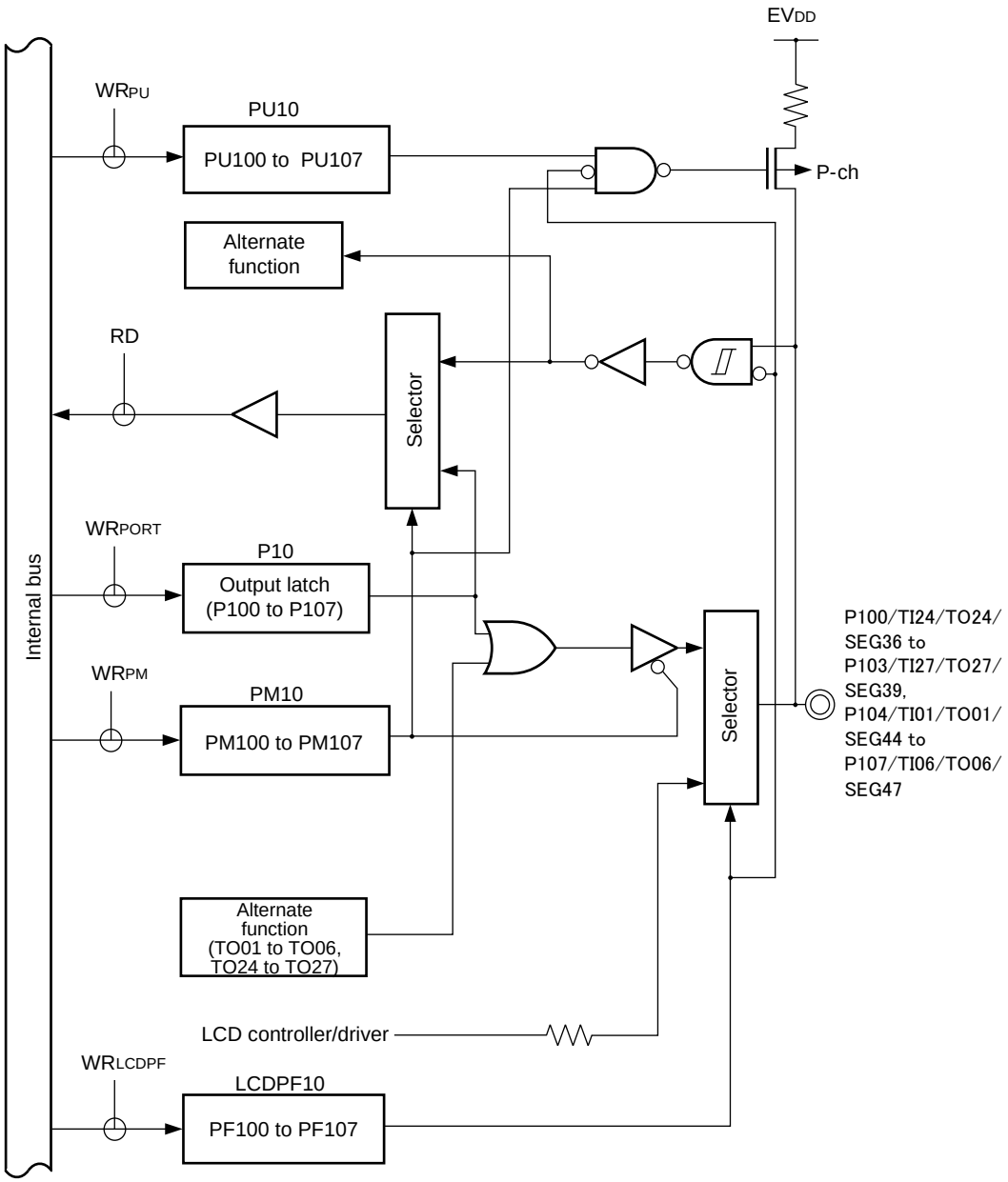
To use P100 to P107 as the port function, refer Table 4-12.

Table 4-12. Setting of P100 to P107 Pins to port function

P100 to P107 Pins		LCDPF10 Register	Alternate function		PM10 Register	PIM10 Register	POM Register	Remarks
port	function		Timer	Serial				
P100	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P101	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P102	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P103	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P104	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P105	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P106	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P107	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 10 to input mode.
Figure 4-42 shows a block diagram of port 10.

Figure 4-42. Block Diagram of P100 to P107



P10: Port register 10
PU10: Pull-up resistor option register 10
PM10: Port mode register 10
LCDPF10: LCD port function registers 10
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO01, TO02, TO05, TO06, or TO24 to TO27, set the port latch to 0. When using P100 to P107 as general-purpose ports, specify the port settings so that the alternate function output is fixed 0.

<R> **4.2.12 Port 11**

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: Not provided
128-pin products: P110 to P117 function as an 8-bit I/O port.

Port 11 is an 8-bit I/O port with an output latch. Port 11 can be set to the input mode or output mode in 1-bit units using port mode register 11 (PM11). When the P110 to P117 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 11(PU11)

Input to the P110 to P117 pins can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer, using port input mode register 11 (PIM11).

These pins also function as timer I/O, serial interface data I/O, clock I/O, LCD bus interface data I/O, and segment signal outputs for the LCD controller/driver.

To use P110 to P117 as the port function, refer Table 4-13.

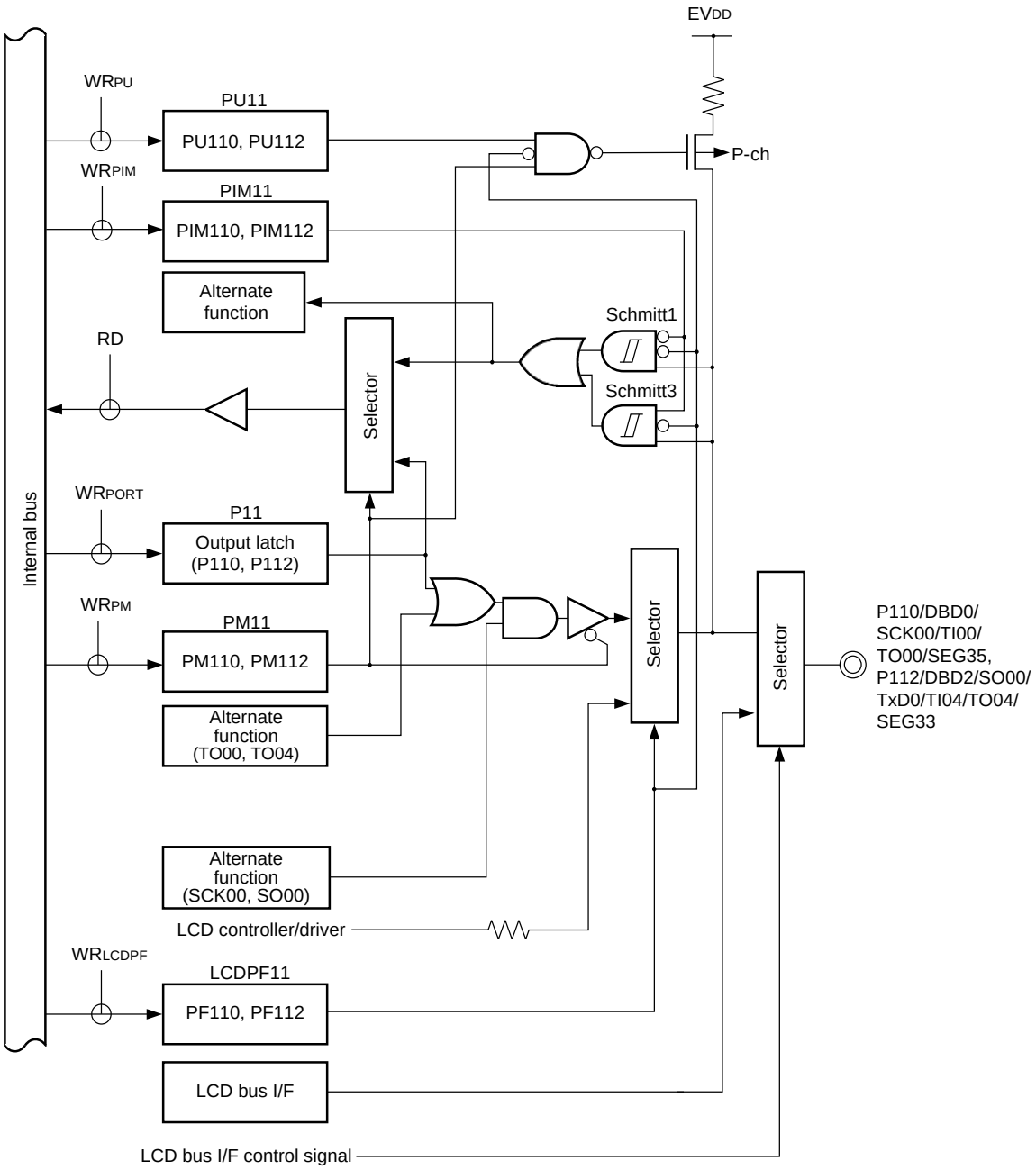
Table 4-13. Setting of P110 to P117 Pins to port function

P110 to P117 Pins		LCDPF11 Register	Alternate function		PM11 Register	PIM11 Register	POM Register	Remarks
port	function		Timer	Serial				
P110	Input port	Digital I/O selection	-	-	Input mode	0	N/A	Schmitt1 input
	Output port		0	1	Output mode	1		Schmitt3 input
P111	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
P112	Input port	Digital I/O selection	-	-	Input mode	0	N/A	Schmitt1 input
	Output port		0	1	Output mode	1		Schmitt3 input
P113	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
P114	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
P115	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
P116	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
P117	Input port	Digital I/O selection	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 11 to input mode.

Figures 4-43 and 4-44 show block diagrams of port 11.

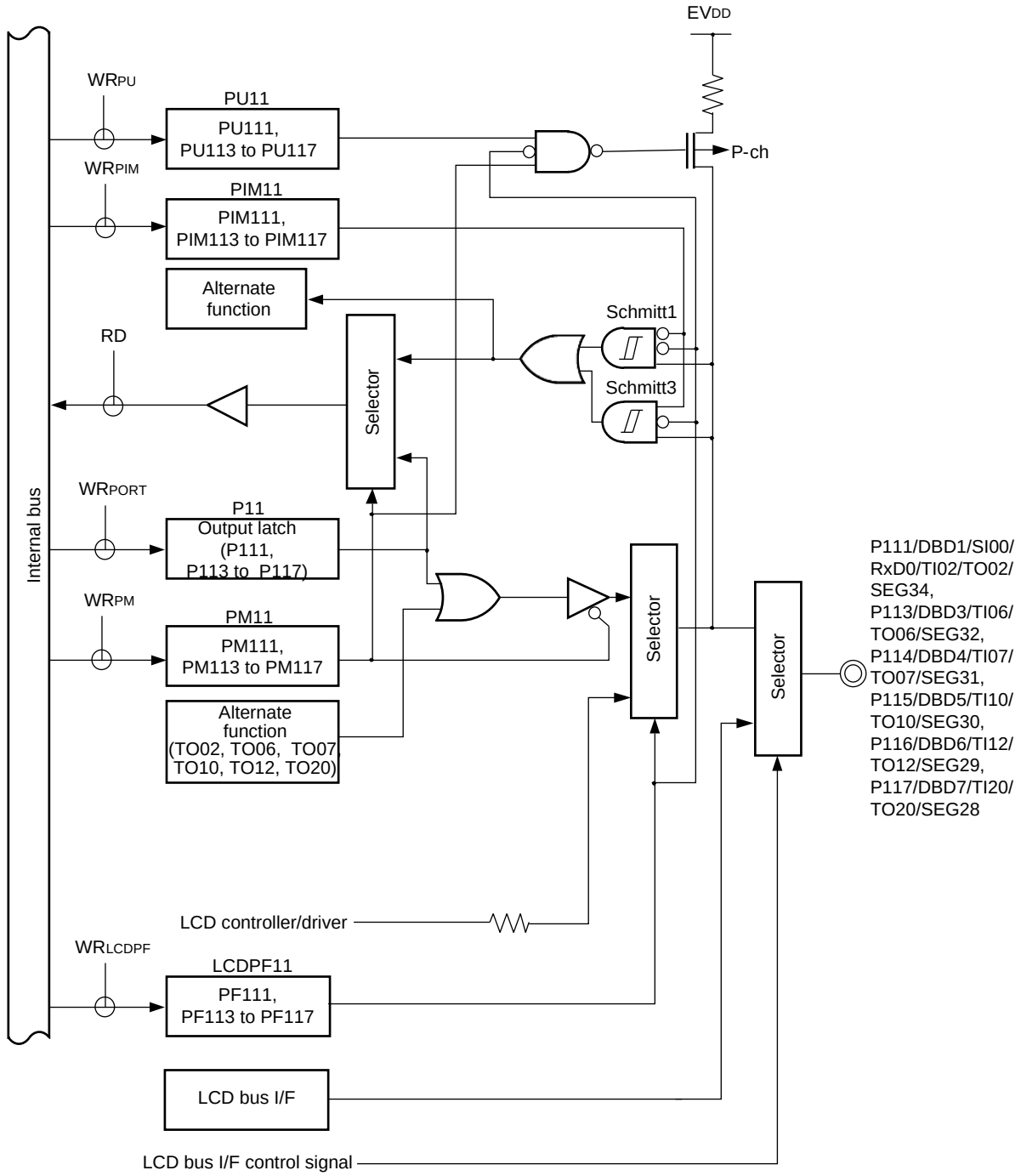
Figure 4-43. Block Diagram of P110 and P112



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PIM11: Port Input mode register 11
LCDPF11: LCD port function registers 11
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO00 or TO04, set the port latch to 0.
When using the alternate function SCK00, SO00, or TxD0, set the port latch to 1.
When using the alternate function DBD0 or DBD2, set the port latch to 0 and the port mode register to 1 (input mode).
When using P110 or P112 as a general-purpose port, specify the port settings so that the alternate function output is fixed (Timer to 0 and Serial to 1).

Figure 4-44. Block Diagram of P111, P113 to P117



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PIM11: Port Input mode register 11
LCDPF11: LCD port function registers 11
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO02, TO06, TO07, TO10, TO12 or TO20, set the port latch to 0.
When using the alternate function DBD1 or DBD3 to DBD7, set the port latch to 0 and the port mode register to 1 (input mode).
When using P111 or P113 to P117 as a general-purpose port, specify the port settings so that the alternate function output is fixed 0.

<R> **4.2.13 Port 12**

48-pin products: P121 to P124 function as a 4-bit Input port.
64-pin products: P121 to P124 function as a 4-bit Input port.
80-pin products: P121 to P124 function as a 4-bit Input port.
100-pin products: P121 to P124 function as a 4-bit Input port.
128-pin products: P121 to P124 function as a 4-bit Input port, P125 to P127 function as a 3-bit I/O port.

P121 to P124 is a 4-bit Input port.

P125 to P127 is a 3-bit I/O port with an output latch. P125 to P127 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P125 to P127 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 12 (PU12).

P121 to P124 pins also function as external clock input for main system clock, external clock input for subsystem clock.
P125 to P127 pins also function as timer I/O and segment signal outputs for the LCD controller/driver.

To use P121 to P127 as the port function, refer Table 4-14 and 4-15.

Table 4-14. Setting of P121 to P124 Pins to port function

P121 to P124 Pins		CMC Register			Remarks
port	function	EXCLK	OSCSEL	OSCSELS	
P121	Input port	-	0	-	
		1	1		
P122	Input port	-	0	-	
P123	Input port	-	-	0	
P124	Input port	-	-	0	

Table 4-15. Setting of P125 to P127 Pins to port function

P125 to P127 Pins		LCDPF12 Register	Alternate function		PM12 Register	PIM12 Register	POM Register	Remarks
port	function		Timer	Serial				
P125	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P126	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P127	Input port	Digital I/O selection	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
-	-	LCD Segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 12 to input mode.
Figure 4-45 and 4-47 show block diagrams of port 12.

Figure 4-45. Block Diagram of P121, P122

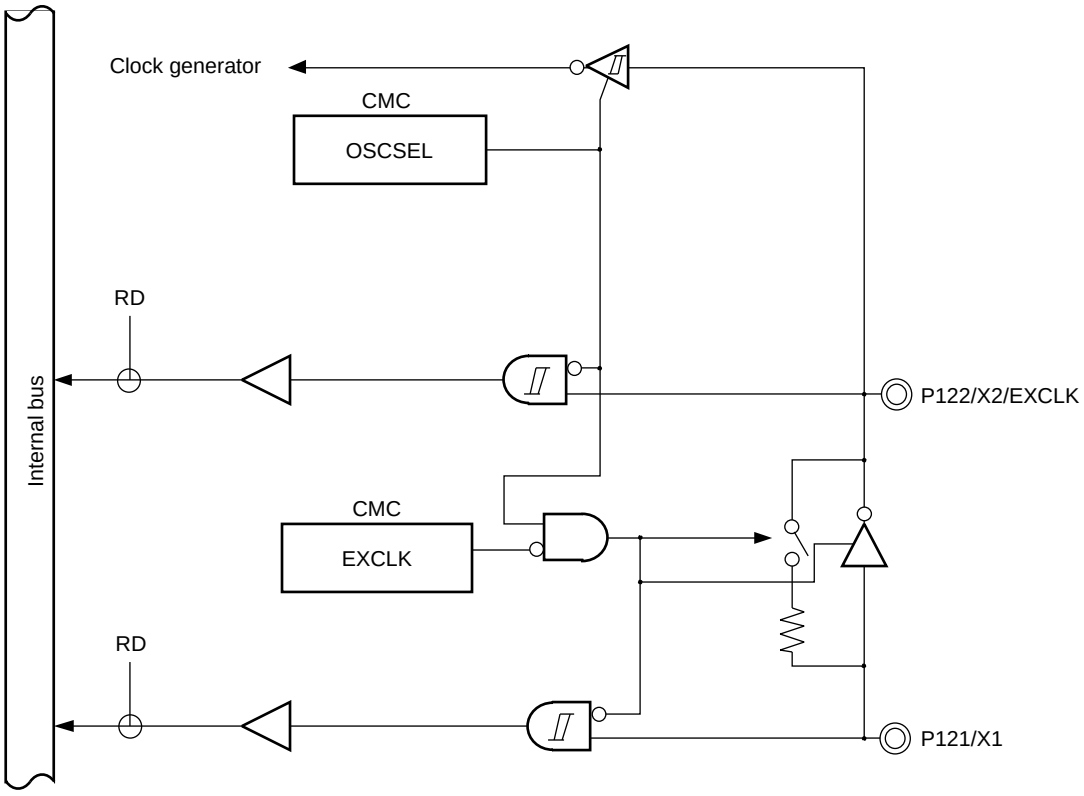


Figure 4-46. Block Diagram of P123, P124

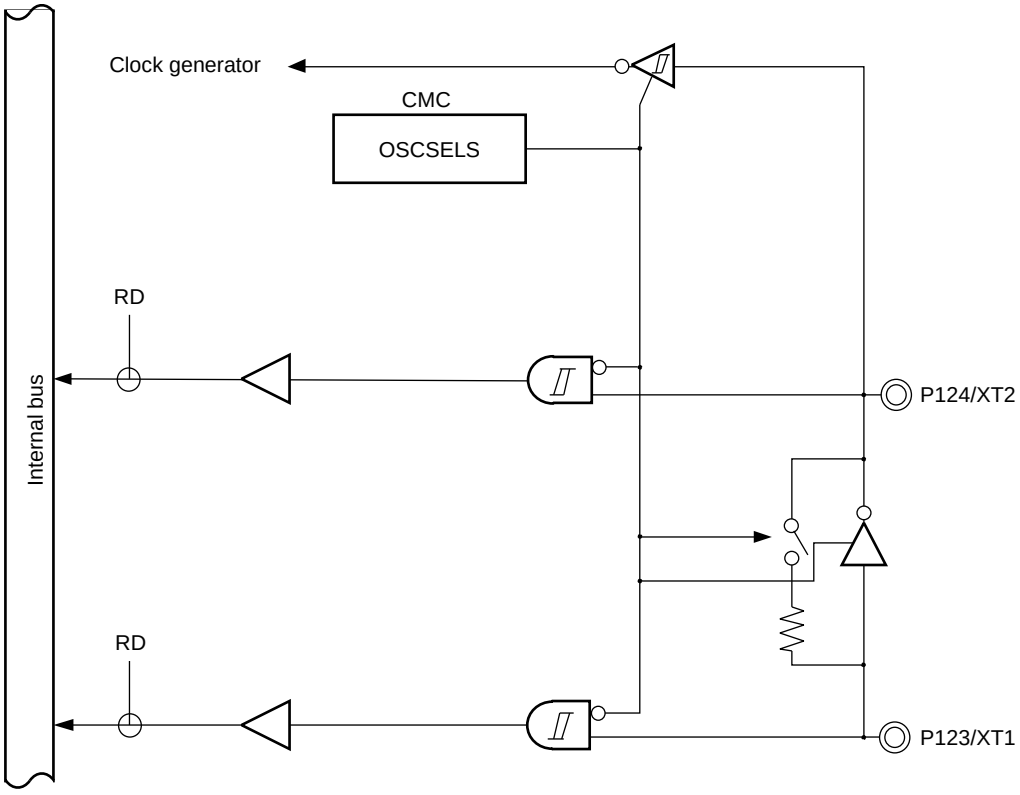
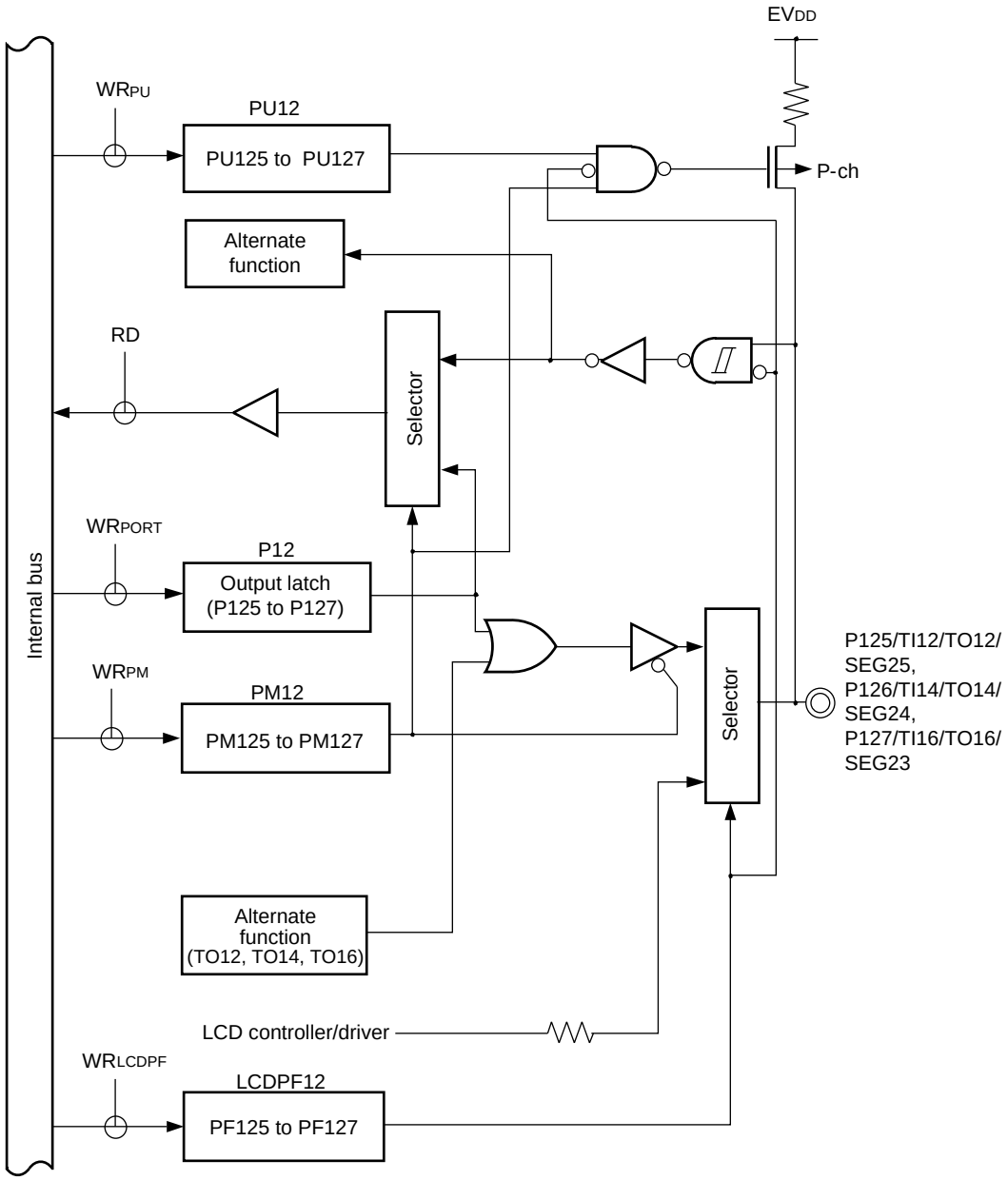


Figure 4-47. Block Diagram of P125 to P127



P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
LCDPF12: LCD port function registers 12
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO12, TO14, or TO16, set the port latch to 0.
When using P125 to P127 as general-purpose ports, specify the port settings so that the alternate function output is fixed 0.

4.2.14 Port 13

48-pin products: P137 functions as a 1-bit Input port.

64-pin products: P137 functions as a 1-bit Input port.

80-pin products: P137 functions as a 1-bit Input port.

100-pin products: P130 functions as a 1-bit Output port, P131 to P136 function as a 6-bit I/O port, and P137 functions as a 1-bit Input port.

<R> 128-pin products: P130 functions as a 1-bit Output port, P131 to P136 function as a 6-bit I/O port, and P137 functions as a 1-bit Input port.

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

P131 to P136 is a 6-bit I/O port with an output latch. P131 to P136 can be set to the input mode or output mode in 1-bit units using port mode register 13 (PM13). When the P131 to P136 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 13 (PU13).

P137 is a 1-bit input-only port.

Input to the P135 pin can be specified through a normal Schmitt3 input buffer or a Schmitt1 input buffer, using port input mode register 13 (PIM13).

Output from the P136 pin can be specified as normal CMOS output or N-ch open-drain output (VDD tolerance), using port output mode register (POM).

These pins also function as timer I/O, output pins for the sound generator, serial interface data I/O ,and segment signal outputs for the LCD controller/driver.

To use P131 to P136 as the port function, refer Table 4-16.

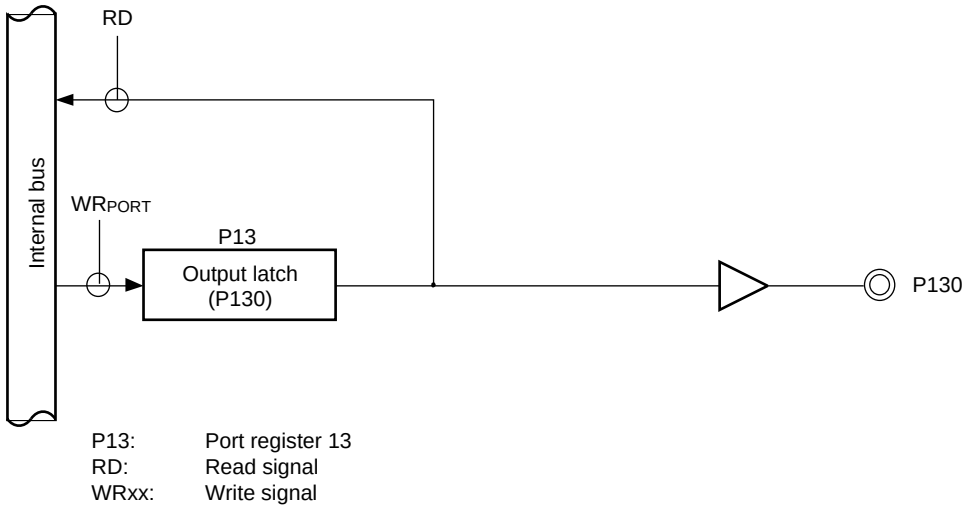
Table 4-16. Setting of P131 to P136 Pins to port function

P131 to P136 Pins		LCDPF13 Register	Alternate function		PM13 Register	PIM13 Register	POM Register	Remarks
port	function		Timer/SGO	Serial				
P131	Input port	N/A	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P132	Input port	N/A	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			
P133	Input port	N/A	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P134	Input port	N/A	-	-	Input mode	N/A	N/A	
	Output port		0	1	Output mode			
P135	Input port	N/A	-	N/A	Input mode	0	N/A	Schmitt1 input
	Output port		0		Output mode	1		Schmitt3 input
P136	Input port	Digital I/O selection	-	N/A	Input mode	N/A	0	CMOS output
	Output port		0		Output mode		1	N-ch OD output
	-	LCD segment output selection	-	-	-	-	-	LCD segment output

Reset signal generation sets port 13 to input mode.

Figures 4-48 to 4-55 show block diagrams of port 13.

Figure 4-48. Block Diagram of P130



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

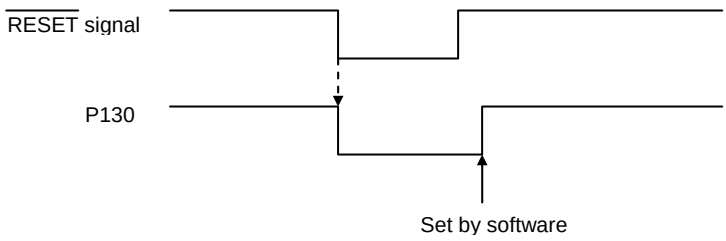
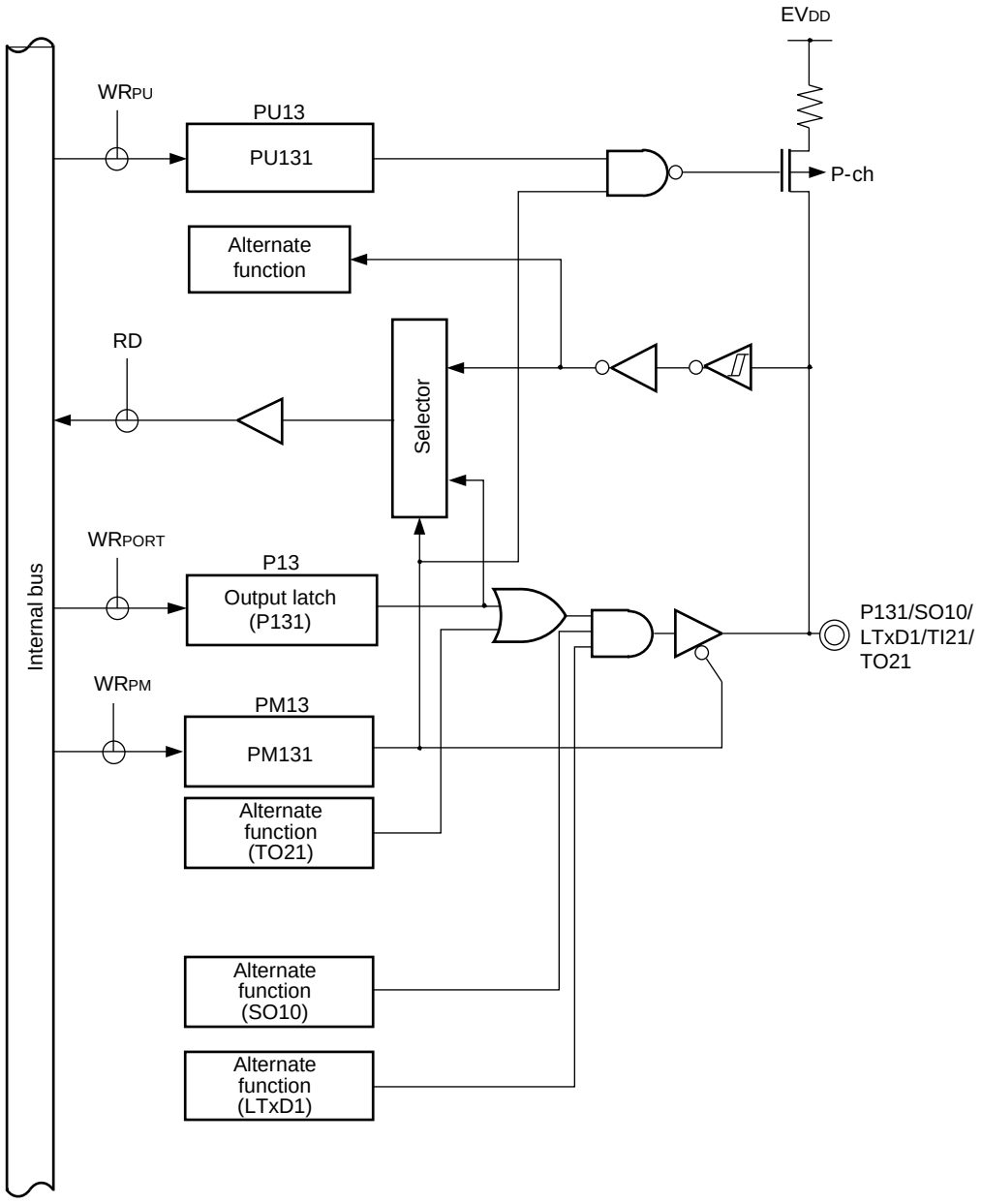


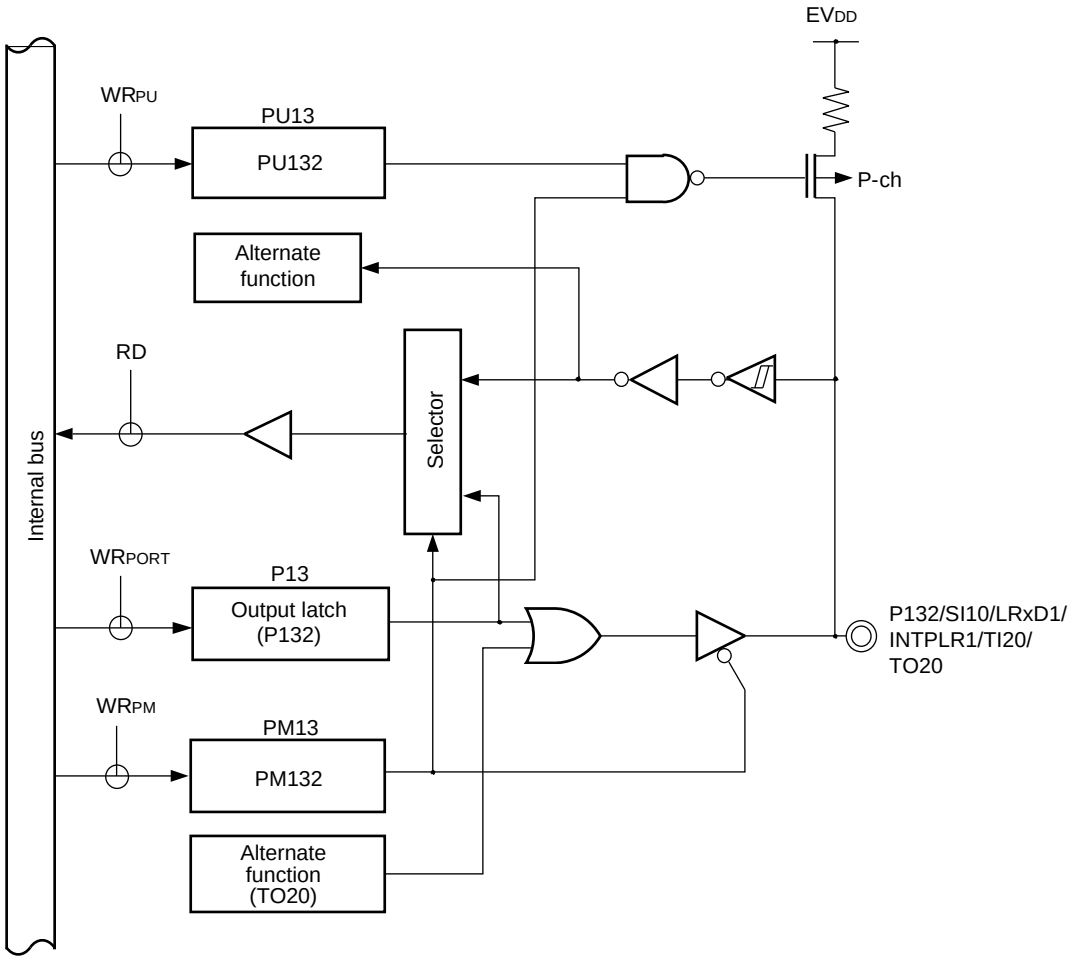
Figure 4-49. Block Diagram of P131



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO21, set the port latch to 0.
When using the alternate function SO10 or LTxD1, set the port latch to 1.
When using P131 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-50. Block Diagram of P132

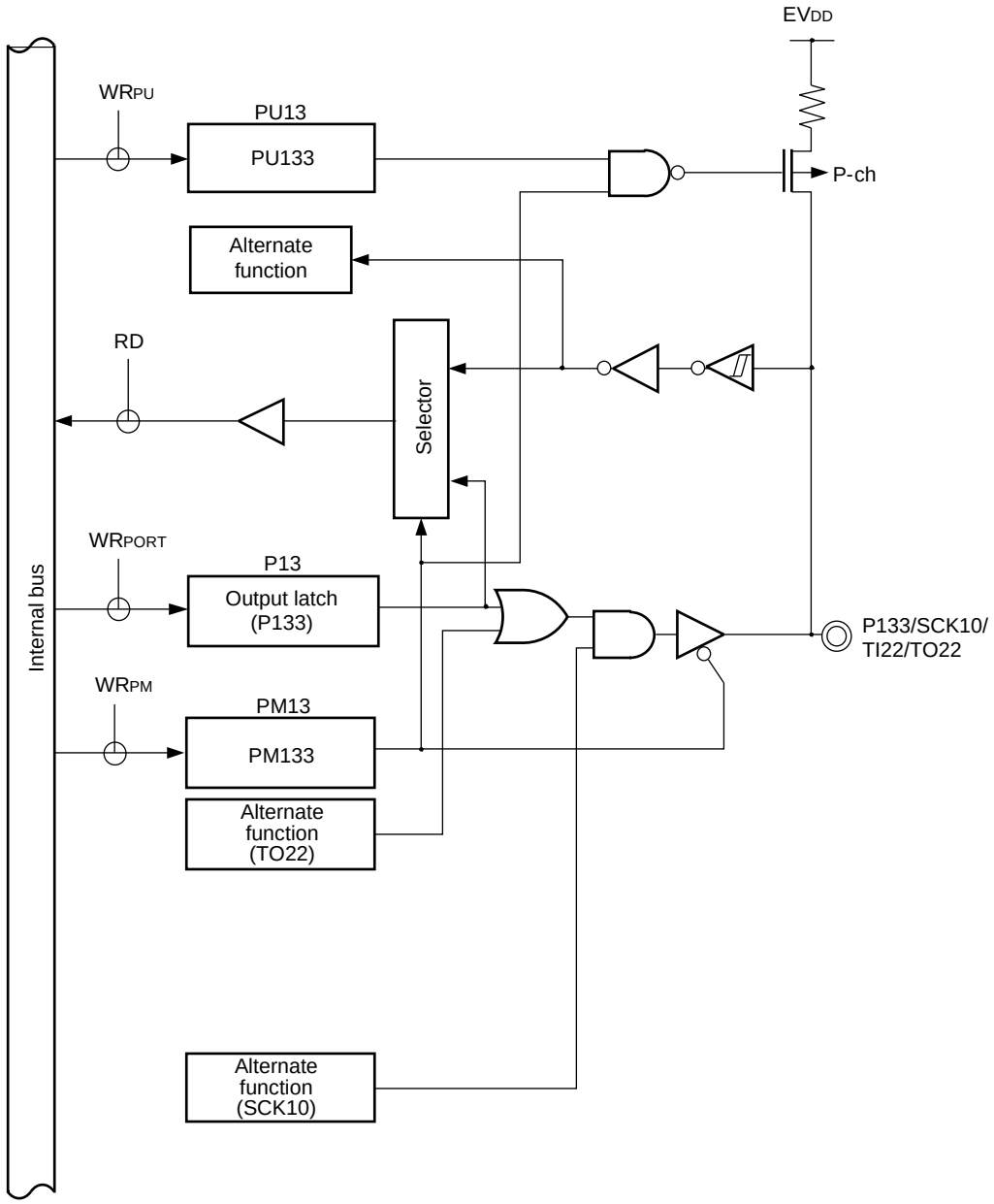


P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO20, set the port latch to 0.

When using P132 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

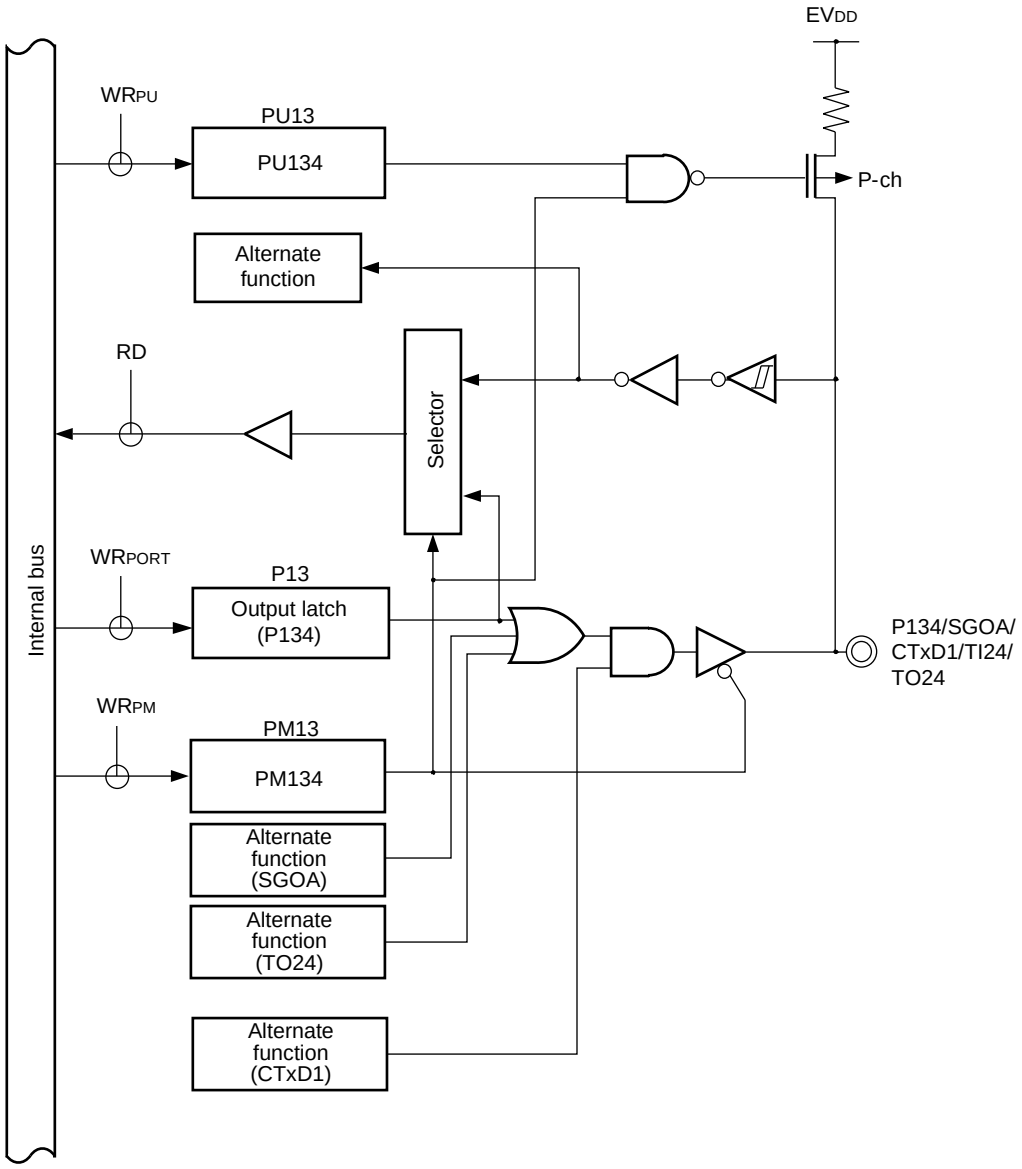
Figure 4-51. Block Diagram of P133



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO22, set the port latch to 0.
When using the alternate function SCK10, set the port latch to 1.
When using P133 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

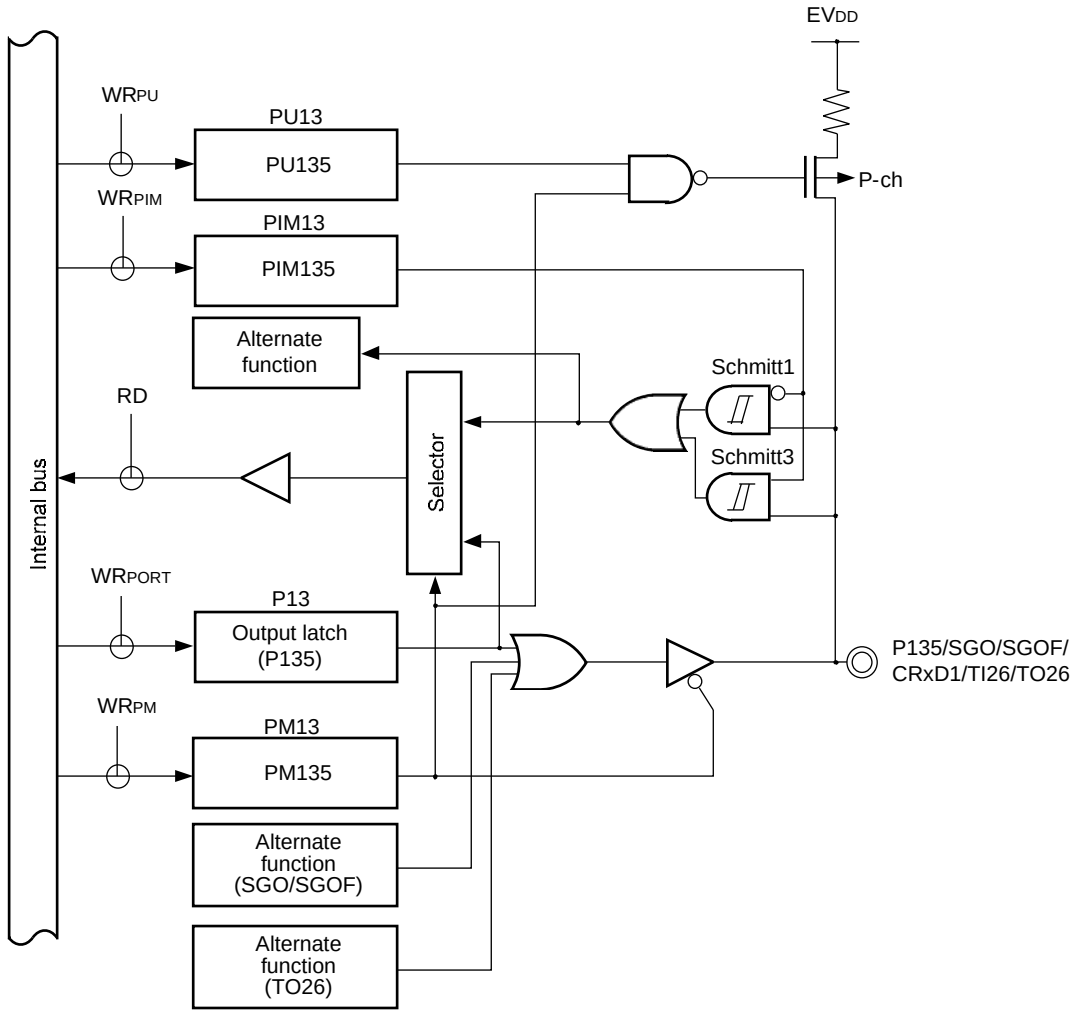
Figure 4-52. Block Diagram of P134



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO24, set the port latch to 0.
When using the alternate function CTxD1, set the port latch to 1.
When using P134 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

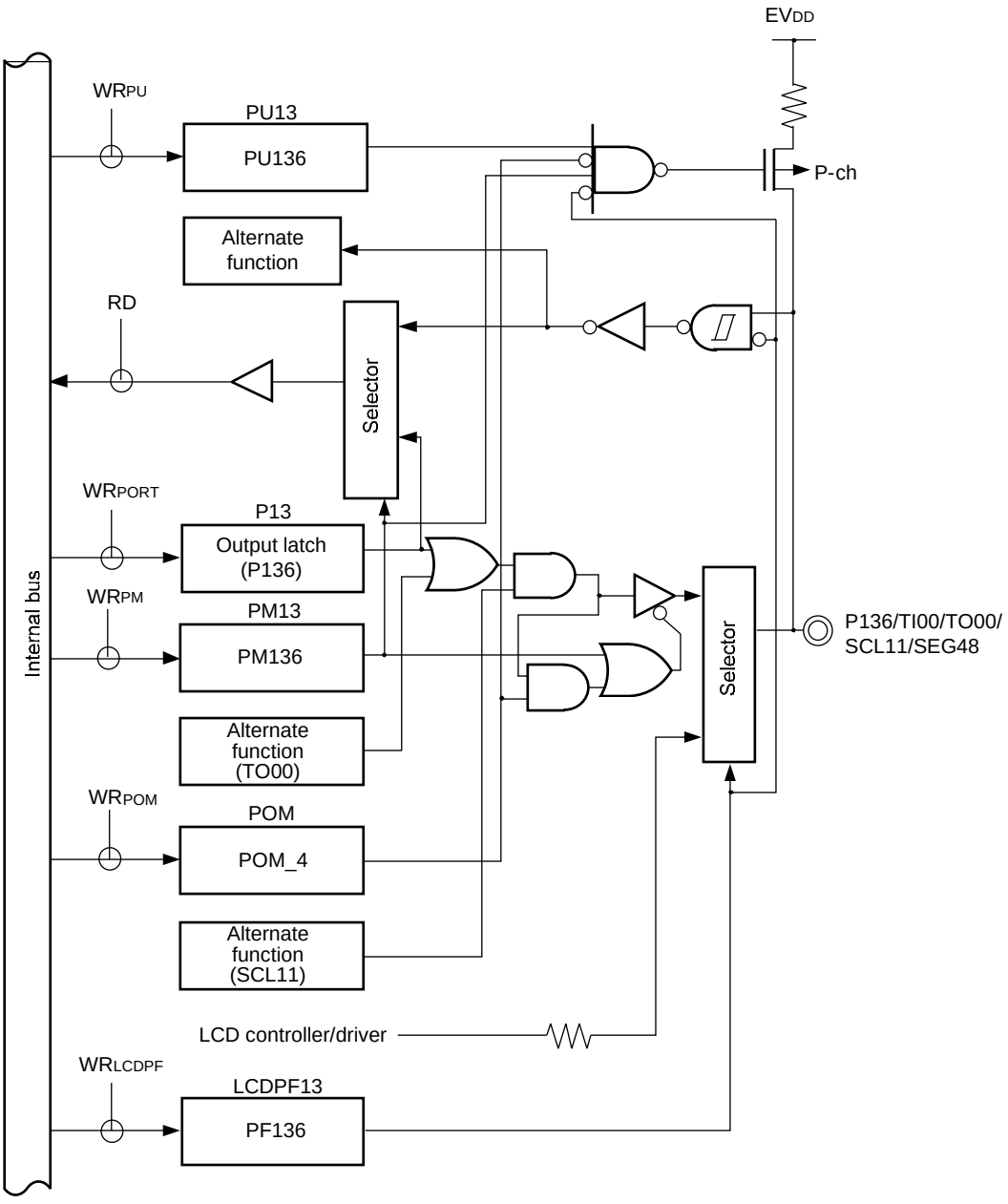
Figure 4-53. Block Diagram of P135



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WRxx: Write signal

Caution When using the alternate function SGO/SGOF or TO26, set the port latch to 0.
When using P135 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed to 0.

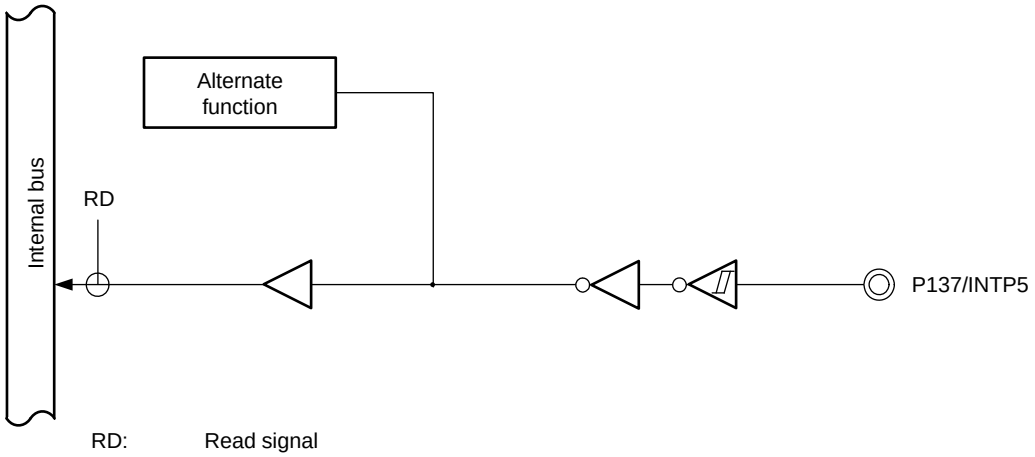
Figure 4-54. Block Diagram of P136



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
LCDPF13: LCD port function registers 13
POM: Port Output mode register
RD: Read signal
WRxx: Write signal

Caution When using the alternate function TO00, set the port latch to 0.
When using the alternate function SCL11, set the port latch to 1.
When using P136 as a general-purpose port, specify the port settings so that the alternate function outputs are fixed (Timer to 0 and Serial to 1).

Figure 4-55. Block Diagram of P137



4.2.15 Port 14

<R> 48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: P140 functions as a 1-bit I/O port.
128-pin products: P140 functions as a 1-bit I/O port.

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

This pin also functions as timer I/O.

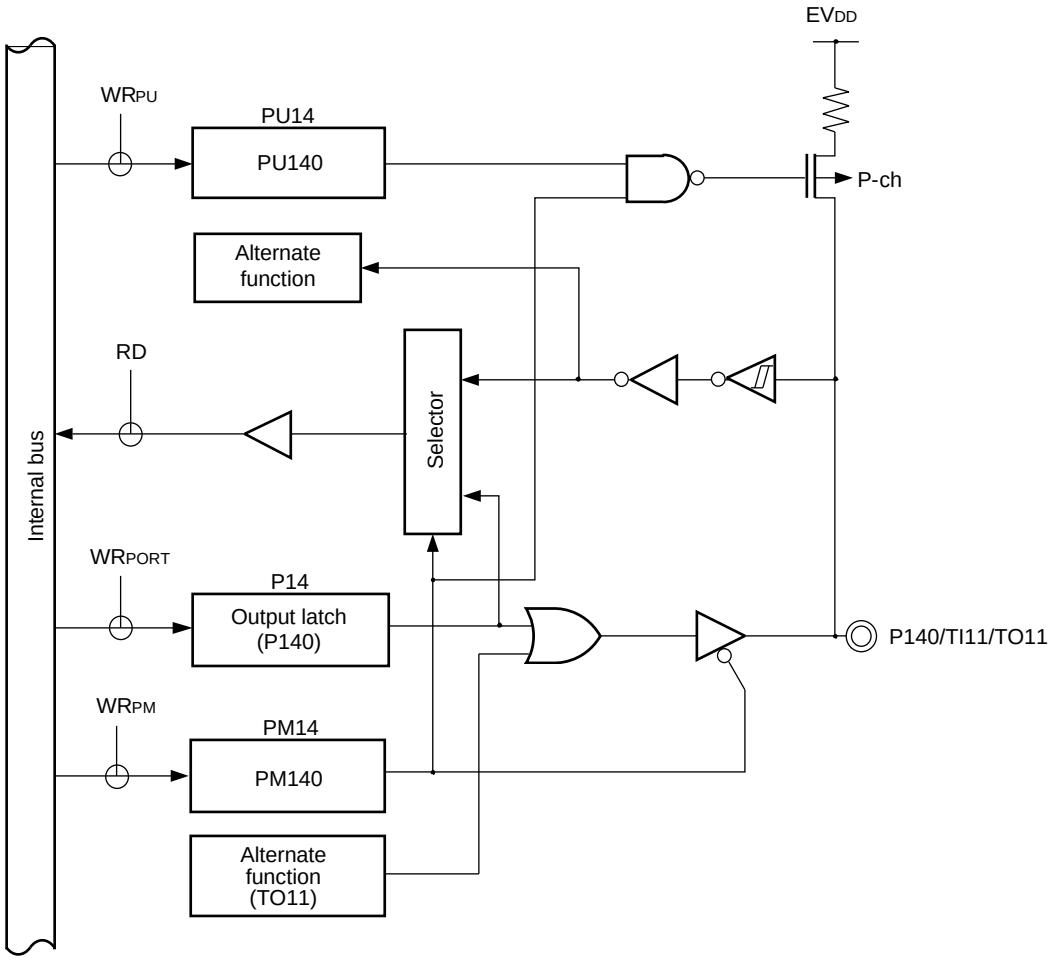
To use P140 as the port function, refer Table 4-17.

Table 4-17. Setting of P140 Pin to port function

P140 Pin		LCDPF14 Register	Alternate function		PM14 Register	PIM Register	POM Register	Remarks
port	function		Timer	Serial				
P140	Input port	N/A	-	N/A	Input mode	N/A	N/A	
	Output port		0		Output mode			

Reset signal generation sets port 14 to input mode.
Figure 4-56 shows block diagram of port 14.

Figure 4-56. Block Diagram of P140



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

Caution When using the alternate function TO11, set the port latch to 0.
When using P140 as a general-purpose port, specify the port settings so that the alternate function output is fixed to 0.

<R> **4.2.16 Port 15**

48-pin products: Not provided
64-pin products: Not provided
80-pin products: Not provided
100-pin products: P150 functions as a 1-bit I/O port.
128-pin products: P150 to P152 function as a 3-bit I/O port

Port 15 is a 3-bit port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

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

To use P150/ANI8 to P152/ANI10 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM15 register. Use these pins starting from the upper bit.

To use P150/ANI8 to P152/ANI10 as digital output pins, set them in the digital I/O mode by using the ADPC register and in the output mode by using the PM15 register. Use these pins starting from the upper bit.

To use P150/ANI8 to P152/ANI10 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM15 register. Use these pins starting from the lower bit.

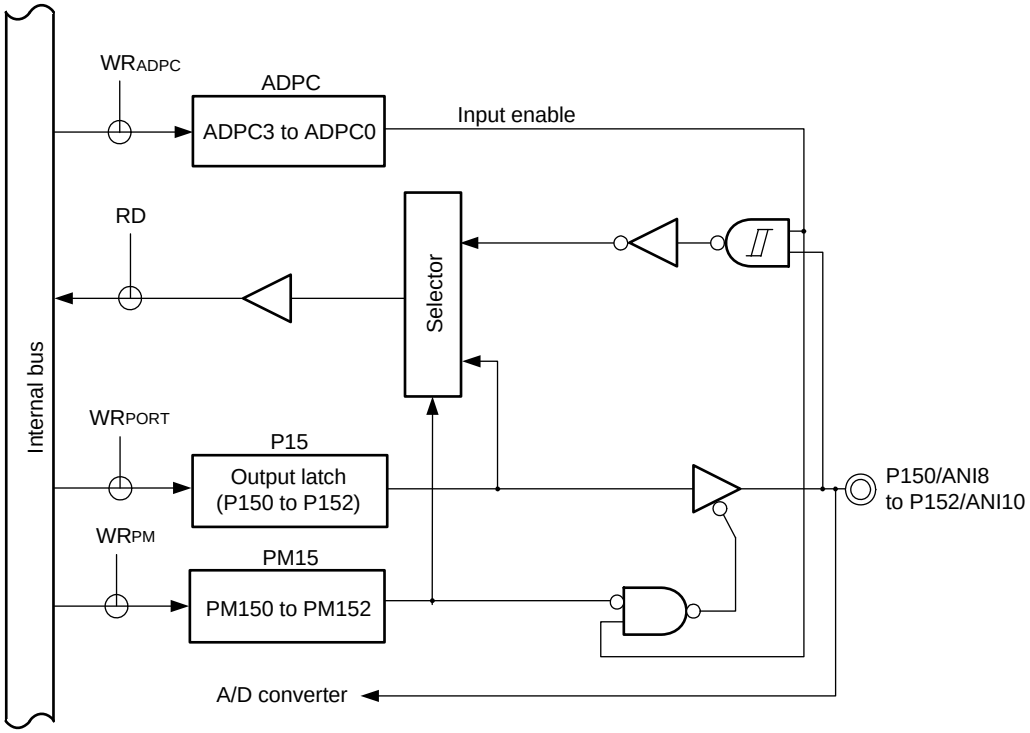
Table 4-18. Setting of P150 to P152 Pins

P150 Pin		ADPC Register	PM2 Register	Remarks
port	function			
P150	Input port	0001 to 1001	Input mode	
	Output port		Output mode	
P151	Input port	0001 to 1010	Input mode	
	Output port		Output mode	
P152	Input port	0001 to 1011	Input mode	
	Output port		Output mode	

All P150/ANI8 to P152/ANI10 are set in the analog input mode when the reset signal is generated.

Figure 4-57 shows block diagram of port 15.

Figure 4-57. Block Diagram of P150 to P152



P15: Port register 15
PM15: Port mode register 15
ADPC: A/D port configuration register
RD: Read signal
WRxx: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POM)
- LCD port function register (LCDPFxx)
- A/D port configuration register (ADPC)
- Stepper motor control register (SMPC)

Caution The undefined bits in each register vary by product and must be used with their initial value.

<R> **Table 4-19. Pxx, PMxx, PUxx, PIMxx, POM, LCDPFxx registers and the bits mounted on each products (1/4)**

Port		Bit name						128-pin	100-pin	80-pin	64-pin	48-pin
		Pxx register	PMxx register	PUxx register	PIMxx register	POM register	LCDPFxx register					
Port 0	0	√	PM00	PU00	N/A	N/A	LCDPF00	√	√	√	√	√
	1	√	PM01	PU01	PIM01		LCDPF01	√	√	√	√	√
	2	√	PM02	PU02	N/A		LCDPF02	√	√	√	√	N/A
	3	√	PM03	PU03			LCDPF03	√	√	√	√	N/A
	4	√	PM04	PU04			LCDPF04	√	√	√	√	N/A
	5	√	PM05	PU05			LCDPF05	√	√	√	√	N/A
	6	√	PM06	PU06			LCDPF06	√	√	√	N/A	N/A
	7	√	PM07	PU07			LCDPF07	√	√	√	√	N/A
Port 1	0	√	PM10	PU10	PIM10	N/A	LCDPF10	√	√	√	√	√
	1	√	PM11	PU11	PIM11		LCDPF11	√	√	√	√	√
	2	√	PM12	PU12	N/A		LCDPF12	√	√	√	√	√
	3	√	PM13	PU13			LCDPF13	√	√	√	√	√
	4	√	PM14	PU14			LCDPF14	√	√	√	√	√
	5	√	PM15	PU15			LCDPF15	√	√	√	√	N/A
	6	√	PM16	PU16			LCDPF16	√	√	√	N/A	N/A
	7	√	PM17	PU17	PIM17		LCDPF17	√	√	√	√	N/A
Port 2	0	√	PM20	N/A	N/A	N/A	N/A	√	√	√	√	√
	1	√	PM21					√	√	√	√	√
	2	√	PM22					√	√	√	√	√
	3	√	PM23					√	√	√	√	√
	4	√	PM24					√	√	√	N/A	N/A
	5	√	PM25					√	√	√	N/A	N/A
	6	√	PM26					√	√	√	N/A	N/A
	7	√	PM27					√	√	√	√	√

<R> **Table 4-19. Pxx, PMxx, PUxx, PIMxx, POM, LCDPFxx registers and the bits mounted on each products (2/4)**

Port		Bit name						128-pin	100-pin	80-pin	64-pin	48-pin
		Pxx register	PMxx register	PUxx register	PIMxx register	POM register	LCDPFxx register					
Port 3	0	P30	PM30	PU30	N/A	POM2	LCDPF30	√	√	√	√	√
	1	P31	PM31	PU31	PIM31	POM3	LCDPF31	√	√	√	√	√
	2	P32	PM32	PU32	N/A	N/A	LCDPF32	√	√	√	√	N/A
	3	P33	PM33	PU33			LCDPF33	√	√	√	√	√
	4	P34	PM34	PU34			LCDPF34	√	√	√	N/A	N/A
	5	P35	PM35	PU35			LCDPF35	√	√	√	N/A	N/A
	6	P36	PM36	PU36			LCDPF36	√	√	√	N/A	N/A
	7	P37	PM37	PU37			LCDPF37	√	√	√	N/A	N/A
Port 4	0	P40	PM40	PU40	N/A	N/A	N/A	√	√	√	√	√
	1	P41	PM41	PU41			N/A	√	N/A	N/A	N/A	N/A
	2	P42	PM42	PU42			LCDPF42	√	N/A	N/A	N/A	N/A
	3	P43	PM43	PU43			LCDPF43	√	N/A	N/A	N/A	N/A
	4	P44	PM44	PU44			LCDPF44	√	N/A	N/A	N/A	N/A
	5	P45	PM45	PU45			LCDPF45	√	N/A	N/A	N/A	N/A
	6	P46	PM46	PU46			LCDPF46	√	N/A	N/A	N/A	N/A
	7	P47	PM47	PU47			LCDPF47	√	N/A	N/A	N/A	N/A
Port 5	0	P50	PM50	PU50	PIM50	POM5	LCDPF50	√	√	N/A	N/A	N/A
	1	P51	PM51	PU51	PIM51	N/A	LCDPF51	√	√	N/A	N/A	N/A
	2	P52	PM52	PU52	PIM52		LCDPF52	√	√	N/A	N/A	N/A
	3	P53	PM53	PU53	N/A		LCDPF53	√	√	N/A	N/A	N/A
	4	P54	PM54	PU54			LCDPF54	√	√	√	√	√
	5	P55	PM55	PU55	PIM55		LCDPF55	√	√	√	√	√
	6	P56	PM56	PU56	PIM56		LCDPF56	√	√	√	√	√
	7	P57	PM57	PU57	PIM57		LCDPF57	√	√	√	√	√
Port 6	0	P60	PM60	PU60	N/A	POM0	N/A	√	√	√	√	√
	1	P61	PM61	PU61	PIM61	POM1		√	√	√	√	√
	2	P62	PM62	PU62	N/A	N/A		√	√	N/A	N/A	N/A
	3	P63	PM63	PU63	PIM63			√	√	N/A	N/A	N/A
	4	P64	PM64	PU64	N/A			√	√	N/A	N/A	N/A
	5	P65	PM65	PU65				√	√	√	N/A	N/A
	6	P66	PM66	PU66				√	√	√	N/A	N/A
Port 7	0	P70	PM70	PU70	PIM70	N/A	N/A	√	√	√	√	N/A
	1	P71	PM71	PU71	N/A			√	√	√	√	N/A
	2	P72	PM72	PU72			LCDPF72	√	√	√	√	√
	3	P73	PM73	PU73			LCDPF73	√	√	√	√	√
	4	P74	PM74	PU74			LCDPF74	√	√	√	√	√
	5	P75	PM75	PU75			LCDPF75	√	√	√	√	√

<R> **Table 4-19. Pxx, PMxx, PUxx, PIMxx, POM, LCDPFxx registers and the bits mounted on each products (3/4)**

Port		Bit name						128-pin	100-pin	80-pin	64-pin	48-pin
		Pxx register	PMxx register	PUxx register	PIMxx register	POM register	LCDPFxx register					
Port 8	0	P80	PM80	PU80	N/A	N/A	LCDPF80	√	√	√	√	√
	1	P81	PM81	PU81			LCDPF81	√	√	√	√	√
	2	P82	PM82	PU82			LCDPF82	√	√	√	√	√
	3	P83	PM83	PU83			LCDPF83	√	√	√	√	√
	4	P84	PM84	PU84			LCDPF84	√	√	√	√	N/A
	5	P85	PM85	PU85			LCDPF85	√	√	√	√	N/A
	6	P86	PM86	PU86			LCDPF86	√	√	√	√	N/A
	7	P87	PM87	PU87			LCDPF87	√	√	√	√	N/A
Port 9	0	P90	PM90	PU90	N/A	N/A	LCDPF90	√	√	√	√	√
	1	P91	PM91	PU91			LCDPF91	√	√	√	√	√
	2	P92	PM92	PU92			LCDPF92	√	√	√	√	√
	3	P93	PM93	PU93			LCDPF93	√	√	√	√	√
	4	P94	PM94	PU94			LCDPF94	√	√	√	√	√
	5	P95	PM95	PU95			LCDPF95	√	√	√	N/A	N/A
	6	P96	PM96	PU96			LCDPF96	√	√	√	N/A	N/A
	7	P97	PM97	PU97			LCDPF97	√	√	√	N/A	N/A
Port 10	0	P100	PM100	PU100	N/A	N/A	LCDPF100	√	N/A	N/A	N/A	N/A
	1	P101	PM101	PU101			LCDPF101	√	N/A	N/A	N/A	N/A
	2	P102	PM102	PU102			LCDPF102	√	N/A	N/A	N/A	N/A
	3	P103	PM103	PU103			LCDPF103	√	N/A	N/A	N/A	N/A
	4	P104	PM104	PU104			LCDPF104	√	N/A	N/A	N/A	N/A
	5	P105	PM105	PU105			LCDPF105	√	N/A	N/A	N/A	N/A
	6	P106	PM106	PU106			LCDPF106	√	N/A	N/A	N/A	N/A
	7	P107	PM107	PU107			LCDPF107	√	N/A	N/A	N/A	N/A
Port 11	0	P110	PM110	PU110	N/A	N/A	LCDPF110	√	N/A	N/A	N/A	N/A
	1	P111	PM111	PU111			LCDPF111	√	N/A	N/A	N/A	N/A
	2	P112	PM112	PU112			LCDPF112	√	N/A	N/A	N/A	N/A
	3	P113	PM113	PU113			LCDPF113	√	N/A	N/A	N/A	N/A
	4	P114	PM114	PU114			LCDPF114	√	N/A	N/A	N/A	N/A
	5	P115	PM115	PU115			LCDPF115	√	N/A	N/A	N/A	N/A
	6	P116	PM116	PU116			LCDPF116	√	N/A	N/A	N/A	N/A
	7	P117	PM117	PU117			LCDPF117	√	N/A	N/A	N/A	N/A
Port 12	1	P121	N/A	N/A	N/A	N/A	N/A	√	√	√	√	√
	2	P122						√	√	√	√	√
	3	P123						√	√	√	√	N/A
	4	P124						√	√	√	√	N/A

<R> **Table 4-19. Pxx, PMxx, PUxx, PIMxx, POM, LCDPFxx registers and the bits mounted on each products (4/4)**

Port		Bit name						128-pin	100-pin	80-pin	64-pin	48-pin
		Pxx register	PMxx register	PUxx register	PIMxx register	POM register	LCDPFxx register					
Port 13	0	P130	N/A	N/A	N/A	N/A	N/A	√	√	N/A	N/A	N/A
	1	P131	PM131	PU131				√	√	N/A	N/A	N/A
	2	P132	PM132	PU132				√	√	N/A	N/A	N/A
	3	P133	PM133	PU133				√	√	N/A	N/A	N/A
	4	P134	PM134	PU134				√	√	N/A	N/A	N/A
	5	P135	PM135	PU135	PIM135			√	√	N/A	N/A	N/A
	6	P136	PM136	PU136	N/A	POM4	LCDPF136	√	√	N/A	N/A	N/A
	7	P137	N/A	N/A		N/A	N/A	√	√	√	√	√
Port 14	0	P140	PM140	PU140	N/A	N/A	N/A	√	√	N/A	N/A	N/A
Port 15	0	P150	PM150	N/A	N/A	N/A	N/A	√	√	N/A	N/A	N/A
	1	P151	PM151					√	N/A	N/A	N/A	N/A
	2	P152	PM152					√	N/A	N/A	N/A	N/A

The format of each register is described in the following pages.

(1) Port mode registers (PM0 to PM9, PM13 to PM15)

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

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

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing 4.5 Settings of Port Mode Register, and Output Latch When Using Alternate Function.

Figure 4-58. Format of Port Mode Register (1/5)
(48-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20	FFH	R/W
PM1	1	1	1	PM14	PM13	PM12	PM11	PM10	FFF21	FFH	R/W
PM2	PM27	1	1	1	PM23	PM22	PM21	PM20	FFF22	FFH	R/W
PM3	1	1	1	1	PM33	1	PM31	PM30	FFF23	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24	FFH	R/W
PM5	PM57	PM56	PM55	PM54	1	1	1	1	FFF25	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	1	1	FFF27	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FFF28	FFH	R/W
PM9	1	1	1	PM94	PM93	PM92	PM91	PM90	FFF29	FFH	R/W
PM13	1	1	1	1	1	1	1	1	FFF2D	FEH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 9 and 13 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 2 to 7 of the PM0 register, bits 5 to 7 of the PM1 register, bits 4 to 6 of the PM2 register, bits 2 and 4 to 7 of the PM3 register, bits 1 to 7 of the PM4 register, bits 0 to 3 of the PM5 register, bits 2 to 7 of the PM6 register, bits 0 to 1 and 6 to 7 of the PM7 register, bits 4 to 7 of the PM8 register, bits 5 to 7 of the PM9 register and bits 0 to 7 of the PM13 register to "1".

Figure 4-58. Format of Port Mode Register (2/5)
(64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	1	PM05	PM04	PM03	PM02	PM01	PM00	FFF20	FFH	R/W
PM1	PM17	1	PM15	PM14	PM13	PM12	PM11	PM10	FFF21	FFH	R/W
PM2	PM27	1	1	1	PM23	PM22	PM21	PM20	FFF22	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FFF23	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24	FFH	R/W
PM5	PM57	PM56	PM55	PM54	1	1	1	1	FFF25	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28	FFH	R/W
PM9	1	1	1	PM94	PM93	PM92	PM91	PM90	FFF29	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 9 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 6 of the PM0 and PM1 register, bits 4 to 6 of the PM2 register, bits 4 to 7 of the PM3 register, bits 1 to 7 of the PM4 register, bits 0 to 3 of the PM5 register, bits 2 to 7 of the PM6 register, bits 6 and 7 of the PM7 register, and bits 5 to 7 of the PM9 register to “1”.

Figure 4-58. Format of Port Mode Register (3/5)
(80-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24	FFH	R/W
PM5	PM57	PM56	PM55	PM54	1	1	1	1	FFF25	FFH	R/W
PM6	1	PM66	PM65	1	1	1	PM61	PM60	FFF26	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 9 ; 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 the PM4 register, bits 0 to 3 of the PM5, bits 2 to 4 and 7 of the PM6 register, and bits 6 and 7 of the PM7 register to “1”.

Figure 4-58. Format of Port Mode Register (4/5)
(100-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25	FFH	R/W
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29	FFH	R/W
PM13	1	PM136	PM135	PM134	PM133	PM132	PM131	0	FFF2D	FEH	R/W
PM14	1	1	1	1	1	1	1	PM140	FFF2E	FFH	R/W
PM15	1	1	1	1	1	1	1	PM150	FFF2F	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 9 and 13 to 15 ; 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 the PM4 register, bits 7 of the PM6 register, bits 6 and 7 of the PM7 register, bits 7 of the PM13 register, and bits 1 to 7 of the PM14 and PM15 registers to “1” and bit 0 of the PM13 register to “0”.

<R>

Figure 4-58. Format of Port Mode Register (5/5)
(128-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25	FFH	R/W
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29	FFH	R/W
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100	FFF2A	FEH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFF2B	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	1	FFF2C	FFH	R/W
PM13	1	PM136P	PM135	PM134	PM133	PM132	PM131	0	FFF2D	FEH	R/W
PM14	1	1	1	1	1	1	1	PM140	FFF2E	FFH	R/W
PM15	1	1	1	1	1	PM152	PM151	PM150	FFF2F	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 15 ; 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 the PM4 register, bits 7 of the PM6 register, bits 6 and 7 of the PM7 register, bits 0 to 4 of the PM12 register, bits 7 of the PM13 register, bits 1 to 7 of the PM14, and bits 3 to 7 of the PM15 registers to “1” and bit 0 of the PM13 register to “0”.

<R> **(2) Port registers (P0 to P15)**

These registers set the output latch value of a port.

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

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

Reset signal generation clears these registers to 00H.

Note If P20 to P27 and P150 to P152 are set up as analog inputs of the A/D converter, when a port is read while in the input mode, 0 is always returned, not the pin level. And while in output mode, the output latch value is not output to port.

If P00 to P07, P10 to P17, P30 to P37, P42 to P47, P50 to P57, P72 to P75, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P125 to P127, and P136 are set up as the segment outputs of LCD controller/driver, when a port is read while in the input mode, 0 is always returned, not the pin level. And while in output mode, the output latch value is not output to port.

If P83, P87, P93, and P97 are set up to ZPD input, when a port is read while in the input mode, 0 is always returned, not the pin level. And while in output mode, the output latch value is not output to port.

Active alternate function	Port read in input mode	Port output in output mode
A/D converter input	0 is read	output latch value is not output
LCD segment output	0 is read	output latch value is not output
ZPD input	0 is read	output latch value is not output

Figure 4-59. Format of Port Register (1/5)
(48-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset (output latch)	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00	00H	R/W
P1	0	0	0	P14	P13	P12	P11	P10	FFF01	00H	R/W
P2	P27	0	0	0	P23	P22	P21	P20	FFF02	00H	R/W
P3	0	0	0	0	P33	0	P31	P30	FFF03	00H	R/W
P4	0	0	0	0	0	0	0	P40	FFF04	00H	R/W
P5	P57	P56	P55	P54	0	0	0	0	FFF05	00H	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06	00H	R/W
P7	0	0	P75	P74	P73	P72	0	0	FFF07	00H	R/W
P8	0	0	0	0	P83	P82	P81	P80	FFF08	00H	R/W
P9	0	0	0	P94	P93	P92	P91	P90	FFF09	00H	R/W
P12	0	0	0	0	0	P122	P121	0	FFF0C	00H	Read only
P13	P137	0	0	0	0	0	0	0	FFF0D	00H	Read only

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

Remark m = 0 to 9, 12, and 13 ; n = 0 to 7

Figure 4-59. Format of Port Register (2/5)
(64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset (output latch)	R/W
P0	P07	0	P05	P04	P03	P02	P01	P00	FFF00	00H	R/W
P1	P17	0	P15	P14	P13	P12	P11	P10	FFF01	00H	R/W
P2	P27	0	0	0	P23	P22	P21	P20	FFF02	00H	R/W
P3	0	0	0	0	P33	P32	P31	P30	FFF03	00H	R/W
P4	0	0	0	0	0	0	0	P40	FFF04	00H	R/W
P5	P57	P56	P55	P54	0	0	0	0	FFF05	00H	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06	00H	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07	00H	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08	00H	R/W
P9	0	0	0	P94	P93	P92	P91	P90	FFF09	00H	R/W
P12	0	0	0	P124	P123	P122	P121	0	FFF0C	00H	Read only
P13	P137	0	0	0	0	0	0	0	FFF0D	00H	Read only

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

Remark m = 0 to 9, 12, and 13 ; n = 0 to 7

Figure 4-59. Format of Port Register (3/5)
(80-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset (output latch)	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00	00H	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01	00H	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02	00H	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03	00H	R/W
P4	0	0	0	0	0	0	0	P40	FFF04	00H	R/W
P5	P57	P56	P55	P54	0	0	0	0	FFF05	00H	R/W
P6	0	P66	P65	0	0	0	P61	P60	FFF06	00H	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07	00H	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08	00H	R/W
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09	00H	R/W
P12	0	0	0	P124	P123	P122	P121	0	FFF0C	00H	Read only
P13	P137	0	0	0	0	0	0	0	FFF0D	00H	Read only

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

Remark m = 0 to 9, 12, and 13 ; n = 0 to 7

Figure 4-59. Format of Port Register (4/5)
(100-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset (output latch)	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00	00H	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01	00H	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02	00H	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03	00H	R/W
P4	0	0	0	0	0	0	0	P40	FFF04	00H	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05	00H	R/W
P6	0	P66	P65	P64	P63	P62	P61	P60	FFF06	00H	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07	00H	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08	00H	R/W
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09	00H	R/W
P12	0	0	0	P124	P123	P122	P121	0	FFF0C	00H	Read only
P13	P137	P136	P135	P134	P133	P132	P131	P130	FFF0D	00H	R/W[Note]
P14	0	0	0	0	0	0	0	P140	FFF0E	00H	R/W
P15	0	0	0	0	0	0	0	P150	FFF0F	00H	R/W

Pmn	m = 0 to 9 and 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 P137 is read only.

Remark m = 0 to 9 and 12 to 15 ; n = 0 to 7

<R>

Figure 4-59. Format of Port Registers (5/5)
(128-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset (output latch)	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00	00H	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01	00H	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02	00H	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03	00H	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04	00H	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05	00H	R/W
P6	0	P66	P65	P64	P63	P62	P61	P60	FFF06	00H	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07	00H	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08	00H	R/W
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09	00H	R/W
P10	P107	P106	P105	P104	P103	P102	P101	P100	FFF0A	00H	R/W
P11	P117	P116	P115	P114	P113	P112	P111	P110	FFF0B	00H	R/W
P12	P127	P126	P125	P124	P123	P122	P121	0	FFF0C	00H	Read only
P13	P137	P136	P135	P134	P133	P132	P131	P130	FFF0D	00H	R/W Note
P14	0	0	0	0	0	0	0	P140	FFF0E	00H	R/W
P15	0	0	0	0	0	P152	P151	P150	FFF0F	00H	R/W

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

Note P137 is read only.

Remark m = 0 to 15 ; n = 0 to 7

<R> **(3) Pull-up resistor option registers (PU0, PU1, PU3 to PU14)**

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, bits used as alternate-function output pins, bits used as alternate-function ZPD input pins, and POM is set to 1, regardless of the settings of these registers.

Table 4-20. on-chip pull-up resistor enable condition

operation mode		on-chip pull-up resistor
PM register setting	other setting	
output mode	-	Can not be connected
Input mode	alternate-function output mode	
	ZPD input mode	
	POM is set to 1	
Input mode	other than those above	usable

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

Reset signal generation clears these registers to 00H (Only PU4 is set to 01H)

Figure 4-60. Format of Pull-up resistor option Register (1/5)
(48-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030	00H	R/W
PU1	0	0	0	PU14	PU13	PU12	PU11	PU10	F0031	00H	R/W
PU3	0	0	0	0	PU33	0	PU31	PU30	F0033	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034	01H	R/W
PU5	PU57	PU56	PU55	PU54	0	0	0	0	F0035	00H	R/W
PU6	0	0	0	0	0	0	PU61	PU60	F0036	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	0	0	F0037	00H	R/W
PU8	0	0	0	0	PU83	PU82	PU81	PU80	F0038	00H	R/W
PU9	0	0	0	PU94	PU93	PU92	PU91	PU90	F0039	00H	R/W
PU13	0	0	0	0	0	0	0	0	F003D	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0 to 9, and 13 ; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 4-60. Format of Pull-up resistor option Register (2/5)
(64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	0	PU05	PU04	PU03	PU02	PU01	PU00	F0030	00H	R/W
PU1	PU17	0	PU15	PU14	PU13	PU12	PU11	PU10	F0031	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	F0033	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034	01H	R/W
PU5	PU57	PU56	PU55	PU54	0	0	0	0	F0035	00H	R/W
PU6	0	0	0	0	0	0	PU61	PU60	F0036	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038	00H	R/W
PU9	0	0	0	PU94	PU93	PU92	PU91	PU90	F0039	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0 to 9 ; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 4-60. Format of Pull-up resistor option Register (3/5)
(80-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	F0033	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034	01H	R/W
PU5	PU57	PU56	PU55	PU54	0	0	0	0	F0035	00H	R/W
PU6	0	PU66	PU65	0	0	0	PU61	PU60	F0036	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0 to 9 ; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Figure 4-60. Format of Pull-up resistor option Register (4/5)
(100-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	F0033	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035	00H	R/W
PU6	0	PU66	PU65	PU64	PU63	PU62	PU61	PU60	F0036	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039	00H	R/W
PU13	0	PU136	PU135	PU134	PU133	PU132	PU131	0	F003D	00H	R/W
PU14	0	0	0	0	0	0	0	PU140	F003E	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0 to 9,13, and 14 ; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

<R>

Figure 4-60. Format of Pull-up resistor option Registers (5/5)
(128-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	F0033	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035	00H	R/W
PU6	0	PU66	PU65	PU64	PU63	PU62	PU61	PU60	F0036	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039	00H	R/W
PU10	PU107	PU106	PU105	PU104	PU103	PU102	PU101	PU100	F003A	00H	R/W
PU11	PU117	PU116	PU115	PU114	PU113	PU112	PU111	PU110	F003B	00H	R/W
PU12	PU127	PU126	PU125	0	0	0	0	0	F003C	00H	R/W
PU13	0	PU136	PU135	PU134	PU133	PU132	PU131	0	F003D	00H	R/W
PU14	0	0	0	0	0	0	0	PU140	F003E	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 14 ; 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 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM11, PIM13)**

These registers set the input buffer of P01, P10, P11, P17, P31, P50 to P52, P55 to P57, P61, P63, P70, P110 to P117, and P135 in 1-bit units.

Schmitt1 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 4-61. Format of Port input mode Register (1/5)
(48-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040	00H	R/W
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	0	F0043	00H	R/W
PIM5	PIM57	PIM56	PIM55	0	0	0	0	0	F0045	00H	R/W
PIM6	0	0	0	0	0	0	PIM61	0	F0046	00H	R/W
PIM7	0	0	0	0	0	0	0	0	F0047	00H	R/W
PIM13	0	0	0	0	0	0	0	0	F004D	00H	R/W

PIMmn	PIMmn pin input threshold selection (m = 0, 1, 3, 5 to 7, and 13 ; n = 0 to 7)
0	Schmit1 input mode
1	Schmit3 input mode

Figure 4-61. Format of Port input mode Register (2/5)
(64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040	00H	R/W
PIM1	PIM17	0	0	0	0	0	PIM11	PIM10	F0041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	0	F0043	00H	R/W
PIM5	PIM57	PIM56	PIM55	0	0	0	0	0	F0045	00H	R/W
PIM6	0	0	0	0	0	0	PIM61	0	F0046	00H	R/W
PIM7	0	0	0	0	0	0	0	PIM70	F0047	00H	R/W

PIMmn	PIMmn pin input threshold selection (m = 0, 1, 3, and 5 to 7 ; n = 0 to 7)
0	Schmit1 input mode
1	Schmit3 input mode

Figure 4-61. Format of Port input mode Register (3/5)
(80-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040	00H	R/W
PIM1	PIM17	0	0	0	0	0	PIM11	PIM10	F0041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	0	F0043	00H	R/W
PIM5	PIM57	PIM56	PIM55	0	0	0	0	0	F0045	00H	R/W
PIM6	0	0	0	0	0	0	PIM61	0	F0046	00H	R/W
PIM7	0	0	0	0	0	0	0	PIM70	F0047	00H	R/W

PIMmn	PIMmn pin input threshold selection (m = 0, 1, 3, and 5 to 7 ; n = 0 to 7)
0	Schmit1 input mode
1	Schmit3 input mode

Figure 4-61. Format of Port input mode Register (4/5)
(100-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040	00H	R/W
PIM1	PIM17	0	0	0	0	0	PIM11	PIM10	F0041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	0	F0043	00H	R/W
PIM5	PIM57	PIM56	PIM55	0	0	PIM52	PIM51	PIM50	F0045	00H	R/W
PIM6	0	0	0	0	PIM63	0	PIM61	0	F0046	00H	R/W
PIM7	0	0	0	0	0	0	0	PIM70	F0047	00H	R/W
PIM13	0	0	PIM135	0	0	0	0	0	F004D	00H	R/W

PIMmn	PIMmn pin input threshold selection (m = 0, 1, 3, 5 to 7, and 13 ; n = 0 to 7)
0	Schmit1 input mode
1	Schmit3 input mode

Bit name	PIM01	PIM17	PIM11	PIM10	PIM31	PIM57	PIM56	PIM55
Port input function	P01/ CRxD0	P17	P11/ LRxD1 SI00	P10/ SCK00	P31/ SDA11	P57	P56/ SCK01	P55/ SI01
Bit name	PIM52	PIM51	PIM50	PIM63	PIM61	PIM70	PIM135	-
Port input function	P52/ SI10	P51/ SCK10	P50/ SDA11	P63/ CRxD1	P61/ SDA11	P70/ CRxD0 LRxD0	P135/ CRxD1	-

<R>

Figure 4-61. Format of Port Input Mode Registers (5/5)
(128-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040	00H	R/W
PIM1	PIM17	0	0	0	0	0	PIM11	PIM10	F0041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31	0	F0043	00H	R/W
PIM5	PIM57	PIM56	PIM55	0	0	PIM52	PIM51	PIM50	F0045	00H	R/W
PIM6	0	0	0	0	PIM63	0	PIM61	0	F0046	00H	R/W
PIM7	0	0	0	0	0	0	0	PIM70	F0047	00H	R/W
PIM11	PIM117	PIM116	PIM115	PIM114	PIM113	PIM112	PIM111	PIM110	F004B	00H	R/W
PIM13	0	0	PIM135	0	0	0	0	0	F004D	00H	R/W

PIMmn	PIMmn pin input threshold selection (m = 0, 1, 3, 5 to 7, 11, and 13 ; n = 0 to 7)
0	Schmit1 input mode
1	Schmit3 input mode

Bit name	PIM01	PIM17	PIM11	PIM10	PIM31	PIM57	PIM56	PIM55
Port input function	P01/ CRxD0	P17	P11/ LRxD1/ SI00	P10/ SCK00	P31/ SDA11	P57	P56/ SCK01	P55/ SI01
Bit name	PIM52	PIM51	PIM50	PIM63	PIM61	PIM70	PIM135	-
Port input function	P52/ SI10	P51/ SCK10	P50/ SDA11	P63/ CRxD1	P61/ SDA11	P70/ CRxD0/ LRxD0	P135/ CRxD1	-
Bit name	PIM117	PIM116	PIM115	PIM114	PIM113	PIM112	PIM111	PIM110
Port input function	P117/ DBD7	P116/ DBD6	P115/ DBD5	P114/ DBD4	P113/ DBD3	P112/ DBD2	P111/ DBD1	P110/ DBD0

(5) Port output mode register (POM)

This register sets the output mode of P30, P31, P50, P60, P61, P136 in 1-bit units.

N-ch open drain output (VDD tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA11 and SCL11 pins during simplified I^2C communication with an external device of the same potential.

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

Reset signal generation clears this register to 00H.

Figure 4-62. Format of Port input mode Register

(a) 48-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM	0	0	0	0	POM3	POM2	POM1	POM0	F006F	00H	R/W

(b) 64-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM	0	0	0	0	POM3	POM2	POM1	POM0	F006F	00H	R/W

(c) 80-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM	0	0	0	0	POM3	POM2	POM1	POM0	F006F	00H	R/W

(d) 100-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM	0	0	POM5	POM4	POM3	POM2	POM1	POM0	F006F	00H	R/W

(e) 128-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM	0	0	POM5	POM4	POM3	POM2	POM1	POM0	F006F	00H	R/W

POMn	Port output mode selection (n = 0 to 5)
0	Normal output (CMOS) mode
1	Nch-OD output (VDD tolerance) mode

Bit name	POM5	POM4	POM3	POM2	POM1	POM0
Port output function	P50/ TO02/ SDA11	P136/ TO00/ SCL11	P31/ TO21/ SDA11	P30/ TO20/ SCL11	P61/ TO21/ SDA11	P60/ TO20/ SCL11

Remark If use the alternate function of IIC, port output need to be set as Nch open-drain (Nch-OD) output. At that time, POM forces on-chip pull-up resistors should not be active (disabled by circuit).

<R> **(6) LCD port function register (LCDPF0, LCDPF1, LCDPF3, LCDPF5, LCDPF7 to LCDPF13)**

These registers specify the LCD segment signal output function for the port in 1-bit units.

Table 4-21. LCDPFmn register function

LCDPFmn	PMmn	Port pin in/output mode
0	0	Output mode
0	1	Input mode (default)
1	0	Segment output mode (on-chip pull-up resistors to disabled, and port output is disabled)
1	1	Segment output mode (on-chip pull-up resistors to disabled, and port read forced to 0)

Figure 4-63. Format of LCD port function Register (1/5)
(48-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF0	0	0	0	0	0	0	LCDPF01	LCDPF00	F0050	00H	R/W
LCDPF1	0	0	0	LCDPF14	LCDPF13	LCDPF12	LCDPF11	LCDPF10	F0051	00H	R/W
LCDPF3	0	0	0	0	LCDPF33	0	LCDPF31	LCDPF30	F0053	00H	R/W
LCDPF5	LCDPF57	LCDPF56	LCDPF55	LCDPF54	0	0	0	0	F0055	00H	R/W
LCDPF7	0	0	LCDPF75	LCDPF74	LCDPF73	LCDPF72	0	0	F0057	00H	R/W
LCDPF8	0	0	0	0	LCDPF83	LCDPF82	LCDPF81	LCDPF80	F0058	00H	R/W
LCDPF9	0	0	0	LCDPF94	LCDPF93	LCDPF92	LCDPF91	LCDPF90	F0059	00H	R/W
LCDPF13	0	0	0	0	0	0	0	0	F005D	00H	R/W

LCDPFmn	LCDPFmn register function (m = 0, 1, 3, 5, 7 to 9, 13)
0	Used as port or alternate function other than segment output
1	Used as LCD segment signal output

Figure 4-63. Format of LCD port function Register (2/5)
(64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF0	LCDPF07	0	LCDPF05	LCDPF04	LCDPF03	LCDPF02	LCDPF01	LCDPF00	F0050	00H	R/W
LCDPF1	LCDPF17	0	LCDPF15	LCDPF14	LCDPF13	LCDPF12	LCDPF11	LCDPF10	F0051	00H	R/W
LCDPF3	0	0	0	0	LCDPF33	LCDPF32	LCDPF31	LCDPF30	F0053	00H	R/W
LCDPF5	LCDPF57	LCDPF56	LCDPF55	LCDPF54	0	0	0	0	F0055	00H	R/W
LCDPF7	0	0	LCDPF75	LCDPF74	LCDPF73	LCDPF72	0	0	F0057	00H	R/W
LCDPF8	LCDPF87	LCDPF86	LCDPF85	LCDPF84	LCDPF83	LCDPF82	LCDPF81	LCDPF80	F0058	00H	R/W
LCDPF9	0	0	0	LCDPF94	LCDPF93	LCDPF92	LCDPF91	LCDPF90	F0059	00H	R/W

LCDPFmn	LCDPFmn register function (m = 0, 1, 3, 5, 7 to 9)
0	Used as port or alternate function other than segment output
1	Used as LCD segment signal output

Figure 4-63. Format of LCD port function Register (3/5)
(80-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF0	LCDPF07	LCDPF06	LCDPF05	LCDPF04	LCDPF03	LCDPF02	LCDPF01	LCDPF00	F0050	00H	R/W
LCDPF1	LCDPF17	LCDPF16	LCDPF15	LCDPF14	LCDPF13	LCDPF12	LCDPF11	LCDPF10	F0051	00H	R/W
LCDPF3	LCDPF37	LCDPF36	LCDPF35	LCDPF34	LCDPF33	LCDPF32	LCDPF31	LCDPF30	F0053	00H	R/W
LCDPF5	LCDPF57	LCDPF56	LCDPF55	LCDPF54	0	0	0	0	F0055	00H	R/W
LCDPF7	0	0	LCDPF75	LCDPF74	LCDPF73	LCDPF72	0	0	F0057	00H	R/W
LCDPF8	LCDPF87	LCDPF86	LCDPF85	LCDPF84	LCDPF83	LCDPF82	LCDPF81	LCDPF80	F0058	00H	R/W
LCDPF9	LCDPF97	LCDPF96	LCDPF95	LCDPF94	LCDPF93	LCDPF92	LCDPF91	LCDPF90	F0059	00H	R/W

LCDPFmn	LCDPFmn register function (m = 0, 1, 3, 5, 7 to 9)
0	Used as port or alternate function other than segment output
1	Used as LCD segment signal output

Figure 4-63. Format of LCD port function Register (4/5)
(100-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF0	LCDPF07	LCDPF06	LCDPF05	LCDPF04	LCDPF03	LCDPF02	LCDPF01	LCDPF00	F0050	00H	R/W
LCDPF1	LCDPF17	LCDPF16	LCDPF15	LCDPF14	LCDPF13	LCDPF12	LCDPF11	LCDPF10	F0051	00H	R/W
LCDPF3	LCDPF37	LCDPF36	LCDPF35	LCDPF34	LCDPF33	LCDPF32	LCDPF31	LCDPF30	F0053	00H	R/W
LCDPF5	LCDPF57	LCDPF56	LCDPF55	LCDPF54	LCDPF53	LCDPF52	LCDPF51	LCDPF50	F0055	00H	R/W
LCDPF7	0	0	LCDPF75	LCDPF74	LCDPF73	LCDPF72	0	0	F0057	00H	R/W
LCDPF8	LCDPF87	LCDPF86	LCDPF85	LCDPF84	LCDPF83	LCDPF82	LCDPF81	LCDPF80	F0058	00H	R/W
LCDPF9	LCDPF97	LCDPF96	LCDPF95	LCDPF94	LCDPF93	LCDPF92	LCDPF91	LCDPF90	F0059	00H	R/W
LCDPF13	0	LCDPF136	0	0	0	0	0	0	F005D	00H	R/W

LCDPFmn	LCDPFmn register function (m = 0, 1, 3, 5, 7 to 9, 13)
0	Used as port or alternate function other than segment output
1	Used as LCD segment signal output

<R>

Figure 4-63. Format of LCD port function Register (5/5)
(128-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF0	LCDPF07	LCDPF06	LCDPF05	LCDPF04	LCDPF03	LCDPF02	LCDPF01	LCDPF00	F0050	00H	R/W
LCDPF1	LCDPF17	LCDPF16	LCDPF15	LCDPF14	LCDPF13	LCDPF12	LCDPF11	LCDPF10	F0051	00H	R/W
LCDPF3	LCDPF37	LCDPF36	LCDPF35	LCDPF34	LCDPF33	LCDPF32	LCDPF31	LCDPF30	F0053	00H	R/W
LCDPF4	LCDPF47	LCDPF46	LCDPF45	LCDPF44	LCDPF43	LCDPF42	LCDPF41	LCDPF40	F0054	00H	R/W
LCDPF5	LCDPF57	LCDPF56	LCDPF55	LCDPF54	LCDPF53	LCDPF52	LCDPF51	LCDPF50	F0055	00H	R/W
LCDPF7	0	0	LCDPF75	LCDPF74	LCDPF73	LCDPF72	0	0	F0057	00H	R/W
LCDPF8	LCDPF87	LCDPF86	LCDPF85	LCDPF84	LCDPF83	LCDPF82	LCDPF81	LCDPF80	F0058	00H	R/W
LCDPF9	LCDPF97	LCDPF96	LCDPF95	LCDPF94	LCDPF93	LCDPF92	LCDPF91	LCDPF90	F0059	00H	R/W
LCDPF10	LCDPF107	LCDPF106	LCDPF105	LCDPF104	LCDPF103	LCDPF102	LCDPF101	LCDPF100	F005A	00H	R/W
LCDPF11	LCDPF117	LCDPF116	LCDPF115	LCDPF114	LCDPF113	LCDPF112	LCDPF111	LCDPF110	F005B	00H	R/W
LCDPF12	LCDPF127	LCDPF126	LCDPF125	0	0	0	0	0	F005C	00H	R/W
LCDPF13	0	LCDPF136	0	0	0	0	0	0	F005D	00H	R/W

LCDPFmn	LCDPFmn register function (m = 0, 1, 3 to 5, 7 to 13)
0	Used as port or alternate function other than segment output
1	Used as LCD segment signal output

Caution For 128pin production, 24 SEGxx re-direction function by PF registers is supported. For example, SEG47 can be output from P97 if setting PF97=1 or from P107 if setting PF107=1, but do not set PF97=1 and PF107=1 at the same time otherwise both P97 and P107 can output same segment signal, because there is not exlusive-active-control-logic for PF97=1 and PF107=1 at hardware, this case is setting prohibited. See below:

<Example>

SEG name	PF97	PF107	Function
SEG47	0	0	Both Port 97 and Port 107 are not used as LCD segment
	0	1	Port 107 is used as LCD segment. Port 97 is not.
	1	0	Port 97 is used as LCD segment. Port 107 is not.
	1	1	Setting prohibited (Both Port 97 and Port 107 can output segment signal)

About SEGxx that can be re-directed to output from two pins, see Table 4-22.

Table 4-22. SEGxx re-direction function

SEG name	Ports		SEG name	Ports		SEG name	Ports	
SEG7	P31	P42	SEG28	P17	P117	SEG36	P84	P100
SEG14	P00	P43	SEG29	P12	P116	SEG37	P85	P101
SEG15	P01	P44	SEG30	P11	P115	SEG38	P86	P102
SEG23	P15	P127	SEG31	P10	P114	SEG39	P87	P103
SEG24	P14	P126	SEG32	P80	P113	SEG44	P94	P104
SEG25	P13	P125	SEG33	P81	P112	SEG45	P95	P105
SEG26	P74	P47	SEG34	P82	P111	SEG46	P96	P106
SEG27	P75	P46	SEG35	P83	P110	SEG47	P97	P107

<R> **(7) A/D port configuration register (ADPC)**

This register switches the P20/ANI0/AVREFP to P27/ANI7, and P150/ANI8 to P152/ANI10 pins to digital I/O of port or analog input of A/D converter.

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

Reset signal generation sets this register to 00H.

Figure 4-64. Format of A/D port configuration Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADPC	0	0	0	0	ADPC3	ADPC2	ADPC1	ADPC0	F006E	00H	R/W

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching										
				ANI10/P152	ANI9/P151	ANI8/P150	ANI7/P27	ANI6/P26	ANI5/P25	ANI4/P24	ANI3/P23	ANI2/P22	ANI1/AVREFM/P21	ANI0/AVREFP/P20
0	0	0	0	A	A	A	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	D	D	D	A	A	A	A	A
0	1	1	1	D	D	D	D	D	A	A	A	A	A	A
1	0	0	0	D	D	D	D	A	A	A	A	A	A	A
1	0	0	1	D	D	D	A	A	A	A	A	A	A	A
1	0	1	0	D	D	A	A	A	A	A	A	A	A	A
1	0	1	1	D	A	A	A	A	A	A	A	A	A	A
Other than the above				Setting prohibited, all the channels ANI10-0 are alalog input on all hardware sepecification										

A(Analog): Digital functions (input/output) are disabled. PMmn setting is invalid.

D(Digital): Digital functions (input/output) are enabled. PMmn setting is valid.

(8) Stepper motor port control register (SMPC)

This register sets the output mode of stepper motor controller/driver.

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

Reset signal generation clears this register to 00H.

Figure 4-65. Format of Stepper motor port control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SMPC	MOD4	MOD3	MOD2	MOD1	EN4	EN3	EN2	EN1	FFF3F	00H	R/W

ENk	MODk	Port mode selection (k = 1 to 4)
0	-	Port mode All SMkm (m = 1 to 4) are set to port function
1	0	PWM full bridge mode SMkm (m = 1 to 4) set to FULL bridge output control
1	1	2pin Stepper motor mode and 2pin Port Mode SMkm set to PWM output control, depending on the DIRkn bit SMkm, (m = 2n - 1) are in PWM output mode and SMkm, (m = 2n) are in Port mode for DIRkn = 0, SMkm, (m = 2n) are in PWM output mode and SMkm, (m = 2n - 1) are in Port mode for DIRkn = 1

An example of settings when m = 1 is as follows:

EN1	MOD1	DIR11	DIR10	PWM Output Pin Control				Output Mode
				SM11 (sin+)	SM12 (sin-)	SM13 (cos+)	SM14 (cos-)	
0	-	-	-	port	port	port	port	Port mode
1	0	0	0	PWM	0	PWM	0	PWM mode
1	0	0	1	PWM	0	0	PWM	Full bridge
1	0	1	0	0	PWM	0	PWM	
1	0	1	1	0	PWM	PWM	0	
1	1	0	0	PWM	port	PWM	port	PWM mode
1	1	0	1	PWM	port	port	PWM	Half bridge
1	1	1	0	port	PWM	port	PWM	
1	1	1	1	port	PWM	PWM	port	

Caution Set port registers (Pn) and port mode registers (PMn) to 00H, whose pins are in the PWM full bridge mode.

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.

4.5 Settings of Registers, and Output Latch When Using Alternate Function

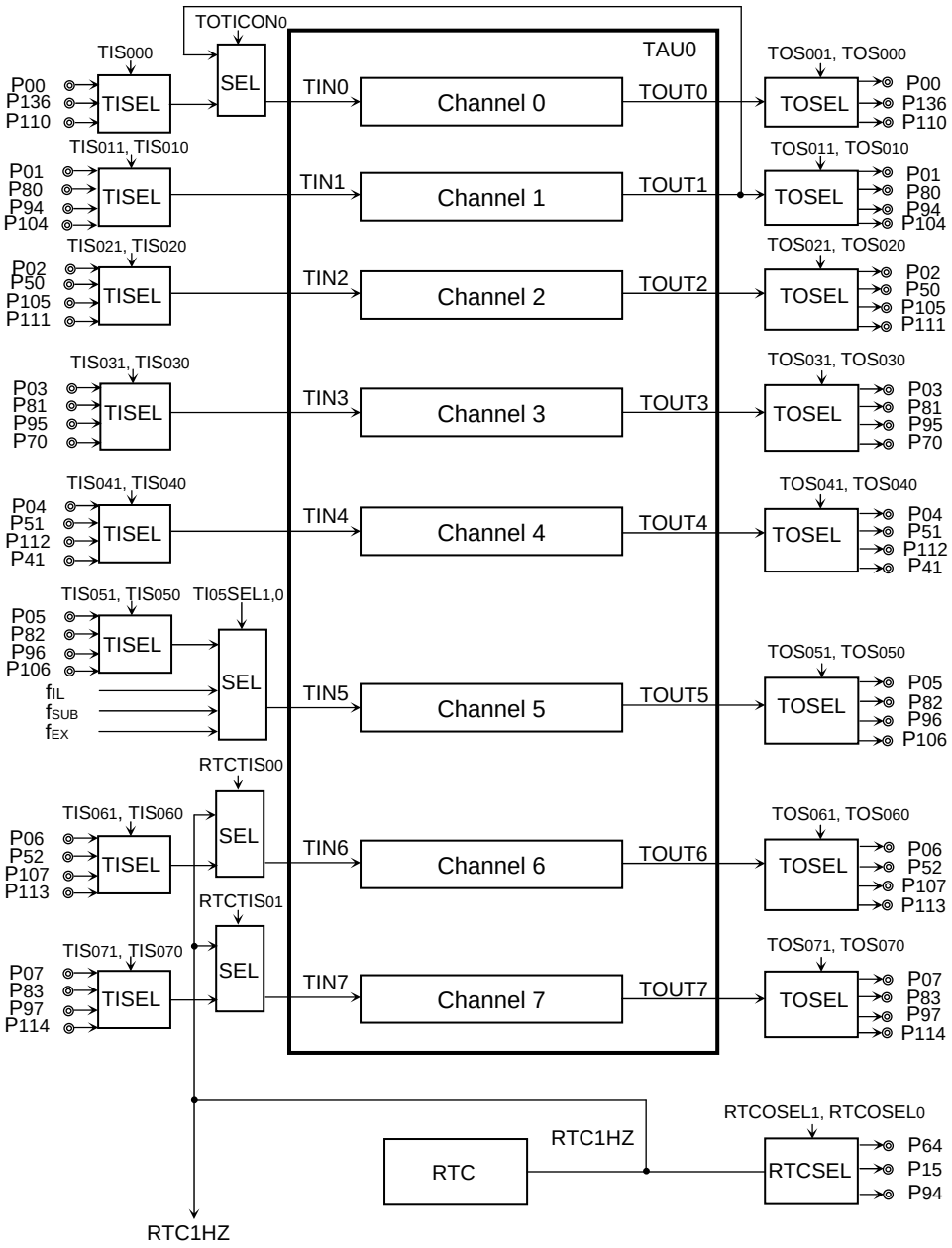
4.5.1 The relationship of alternate function and port

The alternate functions are connected to some port via selector. Figure 4-59 and 60 represent the connection of alternate function and port.

Note: Noise filters and so on are omitted in these figures.

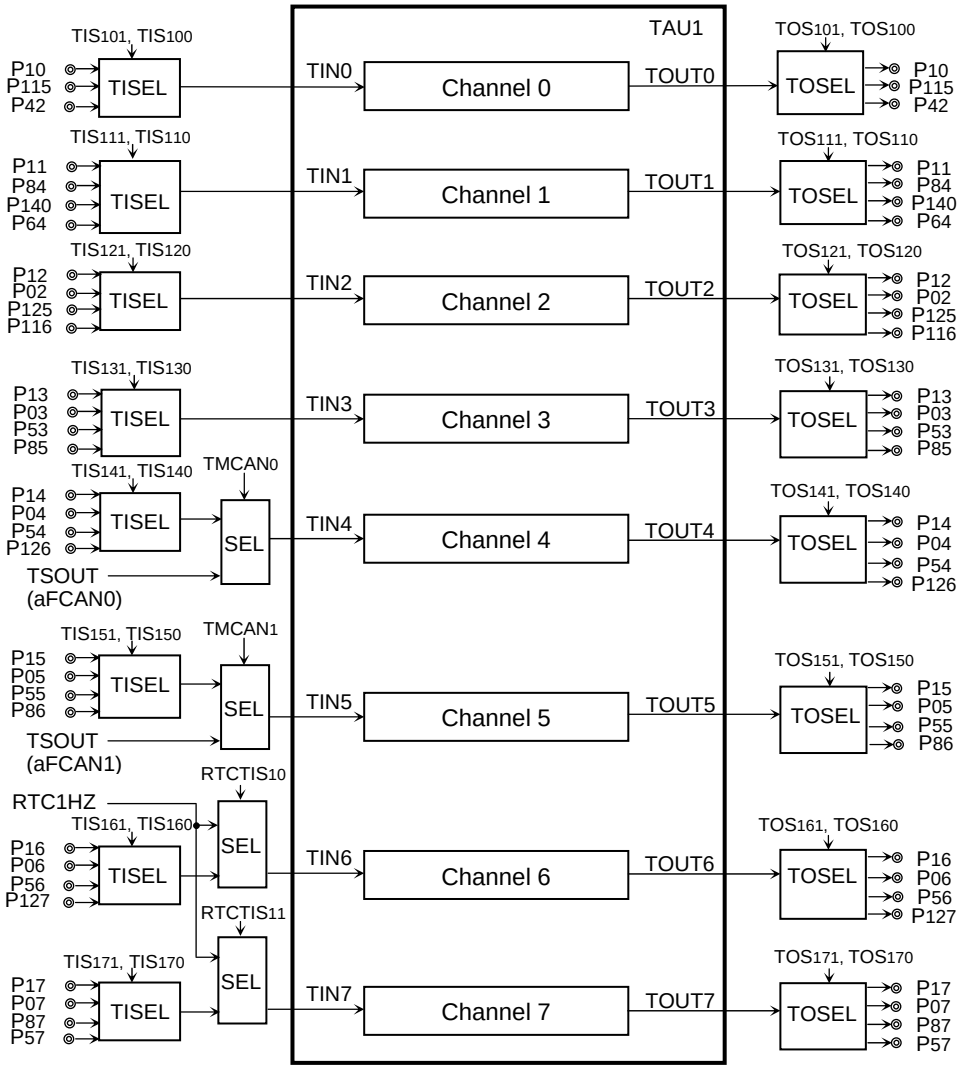
<R>

Figure 4-66. Timer Array unit and RTC I/O connection (128-pin products) (1/3)

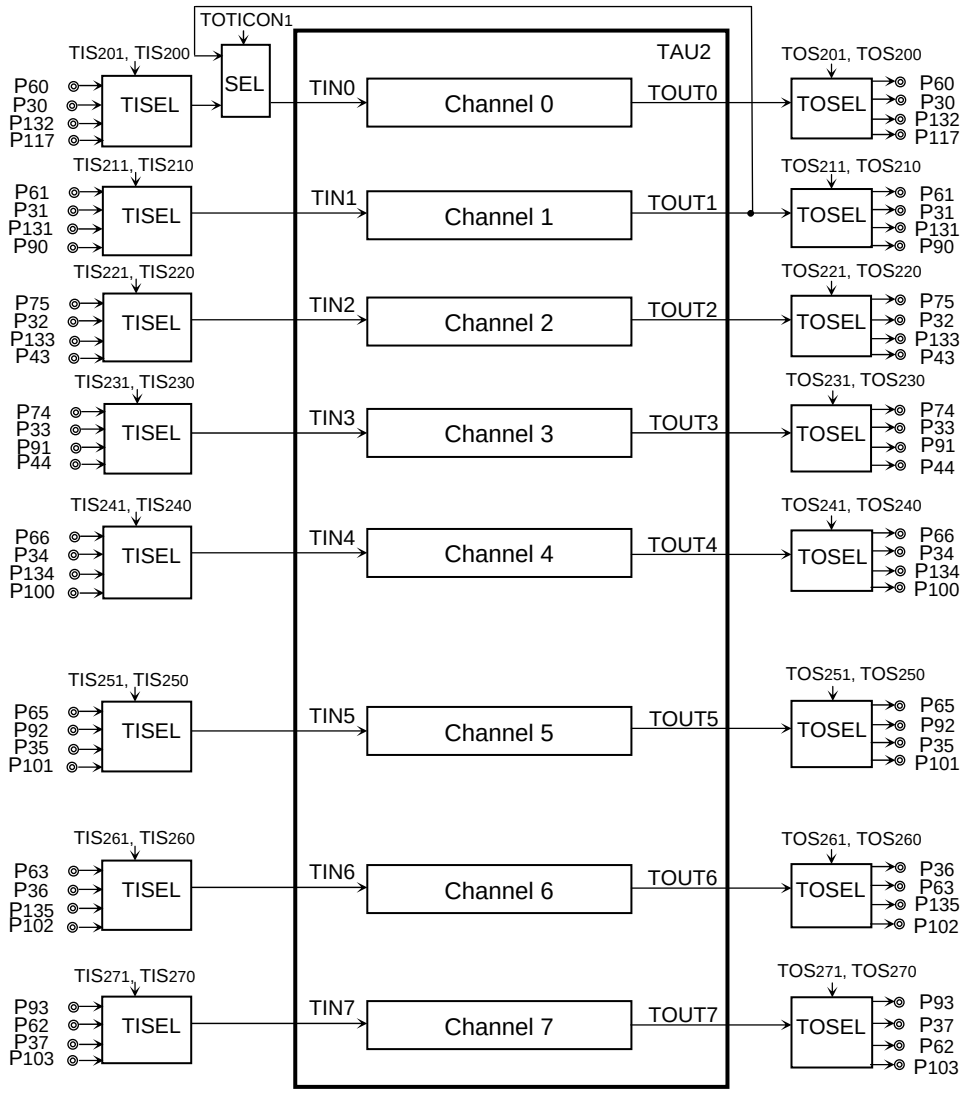


<R>

Figure 4-66. Timer Array unit and RTC I/O connection (128-pin products) (2/3)

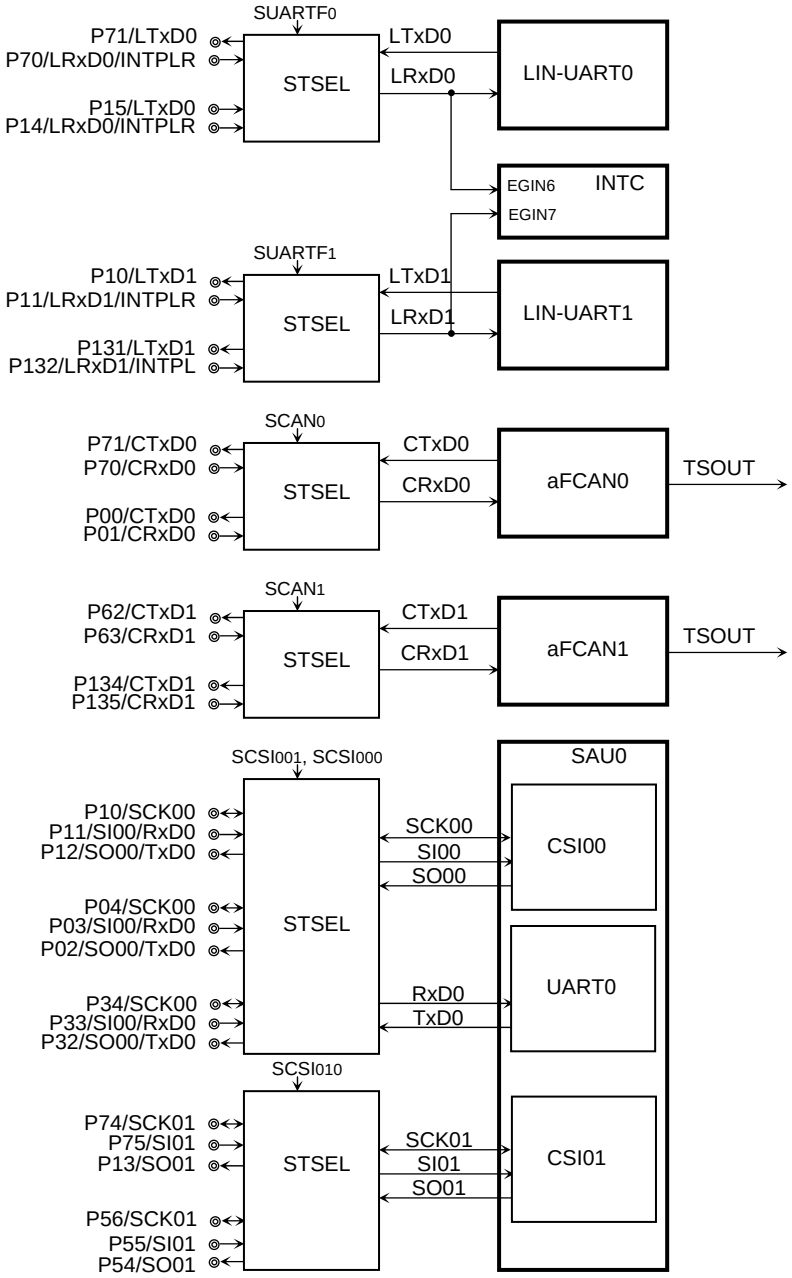


<R> **Figure 4-66. Timer Array unit and RTC I/O connection (128-pin products) (3/3)**

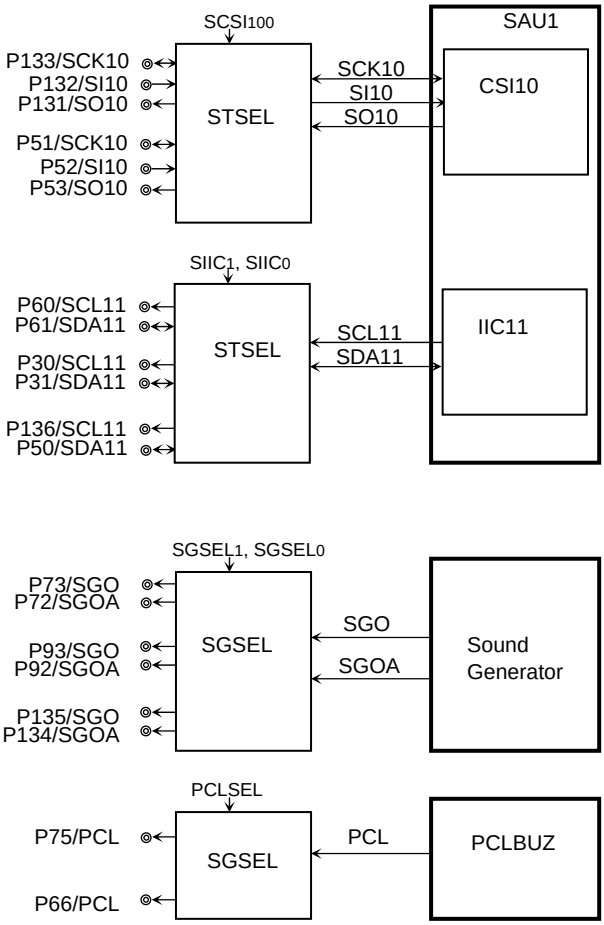


<R>

Figure 4-67. Serial unit, SG, and PCL connection (128-pin products) (1/2)



<R> **Figure 4-67. Serial unit, SG, and PCL connection (128-pin products) (2/2)**



4.5.2 Expanded Control Register of Port Function

(1) Timer input select register (TIS00, TIS01, TIS10, TIS11, TIS20, TIS21)

These registers are used for alternate switch of TAU input pins. TIS00 ~ TIS01 is for TAU unit0, TIS10~ TIS11 for TAU unit1, TIS20~ TIS21 for TAU unit2.

<R>

Figure 4-68. Format of TIS00 and TIS01 Registers (128-pin products) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TIS00	TIS031	TIS030	TIS021	TIS020	TIS011	TIS010	TIS001	TIS000	F0070	00H	R/W
TIS01	TIS071	TIS070	TIS061	TIS060	TIS051	TIS050	TIS041	TIS040	F0071	00H	R/W

TIS001	TIS000	TI00 (TAU unit0 CH0) alternate pin selection
0	0	P00
0	1	P136
1	0	P110
Other than the above		Setting prohibited (same as “00” setting)

TIS011	TIS010	TI01 (TAU unit0 CH1) alternate pin selection
0	0	P01
0	1	P80
1	0	P94
1	1	P104

TIS021	TIS020	TI02 (TAU unit0 CH2) alternate pin selection
0	0	P02
0	1	P50
1	0	P105
1	1	P111

TIS031	TIS030	TI03 (TAU unit0 CH3) alternate pin selection
0	0	P03
0	1	P81
1	0	P95
1	1	P70

TIS041	TIS040	TI04 (TAU unit0 CH4) alternate pin selection
0	0	P04
0	1	P51
1	0	P112
1	1	P41

TIS051	TIS050	TI05 (TAU unit0 CH5) alternate pin selection
0	0	P05
0	1	P82
1	0	P96
1	1	P106

TIS061	TIS060	TI06 (TAU unit0 CH6) alternate pin selection
0	0	P06
0	1	P52
1	0	P107
1	1	P113

Figure 4-68. Format of TIS00 and TIS01 Registers (128-pin products) (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TIS00	TIS031	TIS030	TIS021	TIS020	TIS011	TIS010	TIS001	TIS000	F0070	00H	R/W
TIS01	TIS071	TIS070	TIS061	TIS060	TIS051	TIS050	TIS041	TIS040	F0071	00H	R/W

TIS071	TIS070	TI07 (TAU unit0 CH7) alternate pin selection
0	0	P07
0	1	P83
1	0	P97
<R> | 1 | 1 | P114 |

<R> **Figure 4-69. Format of TIS10 and TIS11 Registers (128-pin products) (1/2)**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TIS10	TIS131	TIS130	TIS121	TIS120	TIS111	TIS110	TIS101	TIS100	F0072	00H	R/W
TIS11	TIS171	TIS170	TIS161	TIS160	TIS151	TIS150	TIS141	TIS140	F0073	00H	R/W

TIS111	TIS110	TI11 (TAU unit1 CH1) alternate pin selection
0	0	P11
0	1	P84
1	0	P140
1	1	P64

TIS121	TIS120	TI12 (TAU unit1 CH1) alternate pin selection
0	0	P12
0	1	P02
1	0	P125
1	1	P116

TIS131	TIS130	TI13 (TAU unit1 CH3) alternate pin selection
0	0	P13
0	1	P03
1	0	P53
1	1	P85

TIS141	TIS140	TI14 (TAU unit1 CH4) alternate pin selection
0	0	P14
0	1	P04
1	0	P54
1	1	P126

TIS151	TIS150	TI15 (TAU unit1 CH5) alternate pin selection
0	0	P15
0	1	P05
1	0	P55
1	1	P86

TIS161	TIS160	TI16 (TAU unit1 CH6) alternate pin selection
0	0	P16
0	1	P06
1	0	P56
1	1	P127

Figure 4-69. Format of TIS10 and TIS11 Registers (128-pin products) (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TIS10	TIS131	TIS130	TIS121	TIS120	TIS111	TIS110	TIS101	TIS100	F0072	00H	R/W
TIS11	TIS171	TIS170	TIS161	TIS160	TIS151	TIS150	TIS141	TIS140	F0073	00H	R/W

TIS171	TIS170	TI17 (TAU unit1 CH7) alternate pin selection
0	0	P17
0	1	P07
1	0	P87
1	1	P57

<R>

Figure 4-70. Format of TIS20 and TIS21 Registers (128-pin products) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TIS20	TIS231	TIS230	TIS221	TIS220	TIS211	TIS210	TIS201	TIS200	F0074	00H	R/W
TIS21	TIS271	TIS270	TIS261	TIS260	TIS251	TIS250	TIS241	TIS240	F0075	00H	R/W

TIS201	TIS200	TI21 (TAU unit2 CH1) alternate pin selection
0	0	P60
0	1	P30
1	0	P132
1	1	P117

TIS211	TIS210	TI21 (TAU unit2 CH1) alternate pin selection
0	0	P61
0	1	P31
1	0	P131
1	1	P90

TIS221	TIS220	TI22 (TAU unit2 CH2) alternate pin selection
0	0	P75
0	1	P32
1	0	P133
1	1	P43

TIS231	TIS230	TI23 (TAU unit2 CH3) alternate pin selection
0	0	P74
0	1	P33
1	0	P91
1	1	P44

TIS241	TIS240	TI24 (TAU unit2 CH4) alternate pin selection
0	0	P66
0	1	P34
1	0	P134
1	1	P100

TIS251	TIS250	TI25 (TAU unit2 CH5) alternate pin selection
0	0	P65
0	1	P92
1	0	P35
1	1	P101

TIS261	TIS260	TI26 (TAU unit2 CH6) alternate pin selection
0	0	P63
0	1	P36
1	0	P135
1	1	P102

Figure 4-70. Format of TIS20 and TIS21 Registers (128-pin products) (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TIS20	TIS231	TIS230	TIS221	TIS220	TIS211	TIS210	TIS201	TIS200	F0074	00H	R/W
TIS21	TIS271	TIS270	TIS261	TIS260	TIS251	TIS250	TIS241	TIS240	F0075	00H	R/W

	TIS271	TIS270	TI27 (TAU unit2 CH7) alternate pin selection
	0	0	P93
	0	1	P62
	1	0	P37
<R>	1	1	P103

(2) Timer output select register (TOS00, TOS01, TOS10, TOS11, TOS20, TOS21)

These registers are used for alternate switch of TAU output pins. TOS00 to TOS01 is for TAU unit0, TOS10 to TOS11 for TAU unit1, TOS20 to TOS21 for TAU unit2.

<R> **Figure 4-71. Format of TOS00 and TOS01 Registers (128-pin products) (1/2)**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TOS00	TOS031	TOS030	TOS021	TOS020	TOS011	TOS010	TOS001	TOS000	F0076	00H	R/W
TOS01	TOS071	TOS070	TOS061	TOS060	TOS051	TOS050	TOS041	TOS040	F0077	00H	R/W

TOS001	TIS000	TO00 (TAU unit0 CH0) alternate pin selection
0	0	P00
0	1	P136
1	0	P110
Other than the above		Setting prohibited (same as “00” setting)

TOS011	TOS010	TO01 (TAU unit0 CH1) alternate pin selection
0	0	P01
0	1	P80
1	0	P94
1	1	P104

TOS021	TOS020	TO02 (TAU unit0 CH2) alternate pin selection
0	0	P02
0	1	P50
1	0	P105
1	1	P111

TOS031	TOS030	TO03 (TAU unit0 CH3) alternate pin selection
0	0	P03
0	1	P81
1	0	P95
1	1	P70

TOS041	TOS040	TO04 (TAU unit0 CH4) alternate pin selection
0	0	P04
0	1	P51
1	0	P112
1	1	P41

TOS051	TOS050	TO05 (TAU unit0 CH5) alternate pin selection
0	0	P05
0	1	P82
1	0	P96
1	1	P106

<R>

Figure 4-71. Format of TOS00 and TOS01 Registers (128-pin products) (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TOS00	TOS031	TOS030	TOS021	TOS020	TOS011	TOS010	TOS001	TOS000	F0076	00H	R/W
TOS01	TOS071	TOS070	TOS061	TOS060	TOS051	TOS050	TOS041	TOS040	F0077	00H	R/W

TOS061	TOS060	TO06 (TAU unit0 CH6) alternate pin selection
0	0	P06
0	1	P52
1	0	P107
1	1	P113

TOS071	TOS070	TO07 (TAU unit0 CH7) alternate pin selection
0	0	P07
0	1	P83
1	0	P97
1	1	P114

<R> **Figure 4-72. Format of TOS10 and TOS11 Registers (128-pin products) (1/2)**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TOS10	TOS131	TOS130	TOS121	TOS120	TOS111	TOS110	TOS101	TOS100	F0079	00H	R/W
TOS11	TOS171	TOS170	TOS161	TOS160	TOS151	TOS150	TOS141	TOS140	F007A	00H	R/W

TOS101	TOS100	TO10 (TAU unit1 CH0) alternate pin selection
0	0	P10
0	1	P115
1	0	P42
Other than the above		Setting prohibited (same as “00” setting)

TOS111	TOS110	TO11 (TAU unit1 CH1) alternate pin selection
0	0	P11
0	1	P84
1	0	P140
1	1	P64

TOS121	TOS120	TO12 (TAU unit1 CH2) alternate pin selection
0	0	P12
0	1	P02
1	0	P125
1	1	P116

TOS131	TOS130	TO13 (TAU unit1 CH3) alternate pin selection
0	0	P13
0	1	P03
1	0	P53
1	1	P85

TOS141	TOS140	TO14 (TAU unit1 CH4) alternate pin selection
0	0	P14
0	1	P04
1	0	P54
1	1	P126

TOS151	TOS150	TO15 (TAU unit1 CH5) alternate pin selection
0	0	P15
0	1	P05
1	0	P55
1	1	P86

TOS161	TOS160	TO16 (TAU unit1 CH6) alternate pin selection
0	0	P16
0	1	P06
1	0	P56
1	1	P127

Figure 4-72. Format of TOS10 and TOS11 Registers (128-pin products) (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TOS10	TOS131	TOS130	TOS121	TOS120	TOS111	TOS110	TOS101	TOS100	F0079	00H	R/W
TOS11	TOS171	TOS170	TOS161	TOS160	TOS151	TOS150	TOS141	TOS140	F007A	00H	R/W

TOS171	TOS170	TO17 (TAU unit1 CH7) alternate pin selection
0	0	P17
0	1	P07
1	0	P87
1	1	P57

<R> **Figure 4-73. Format of TOS20 and TOS21 Registers (128-pin products) (1/2)**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TOS20	TOS231	TOS230	TOS221	TOS220	TOS211	TOS210	TOS201	TOS200	F007B	00H	R/W
TOS21	TOS271	TOS270	TOS261	TOS260	TOS251	TOS250	TOS241	TOS240	F007C	00H	R/W

TOS201	TOS200	TO20 (TAU unit2 CH0) alternate pin selection
0	0	P60
0	1	P30
1	0	P132
1	1	P117

TOS211	TOS210	TO21 (TAU unit2 CH1) alternate pin selection
0	0	P61
0	1	P31
1	0	P131
1	1	P90

TOS221	TOS220	TO22 (TAU unit2 CH2) alternate pin selection
0	0	P75
0	1	P32
1	0	P133
1	1	P43

TOS231	TOS230	TO23 (TAU unit2 CH3) alternate pin selection
0	0	P74
0	1	P33
1	0	P91
1	1	P44

TOS241	TOS240	TO24 (TAU unit2 CH4) alternate pin selection
0	0	P66
0	1	P34
1	0	P134
1	1	P100

TOS251	TOS250	TO25 (TAU unit2 CH5) alternate pin selection
0	0	P65
0	1	P92
1	0	P35
1	1	P101

TOS261	TOS260	TO26 (TAU unit2 CH6) alternate pin selection
0	0	P36
0	1	P63
1	0	P135
1	1	P102

Figure 4-73. Format of TOS20 and TOS21 Registers (128-pin products) (2/2)

TOS271	TOS270	TO27 (TAU unit2 CH7) alternate pin selection
0	0	P93
0	1	P37
1	0	P62
<R> 1	1	P103

Considering direct LED driving, or other large current application, 16-bit resolution PWM outputs are also alternated to the SM pins. When configure pin function, the policy is that odd TO (ch1,3,5,7) of TAU should be output with higher priority. In addition, 4 kinds of output with different periods using different master CH's can be achieved if by this means.

(3) Timer input select else register (TISELSE)

This register provides below selection function.

(a) TAU unit 0 channel5 input selection

The input source can be timer input signal (TI05) from port or internal/external clock.

(b) Timer conjunction function of timer output to timer input just like 78K0/Dx2.

TAU unit0 CH1 output can be connected to TAU unit0 CH0. This function is controlled by bit6.
TAU unit2 CH1 output can be connected to TAU unit2 CH0. This function is controlled by bit7.
This function is used for measuring speed or taco pulse. If only use timer caupture function to measure, there will be too many interrupts and increase the loading of software when input is higher (about 8kHz, 125us interrupt interval). So division of interrupt is necessary. At this usage, one timer is used as capture mode, its output is internally connected to another timer (operated as external event mode) to generate divided interrupt.
(Refer to TMP2 and TMP3 conjunction function of 78K0/Dx2)

Figure 4-74. Format of TISELSE Registers

Address: FFF3E After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TISELSE	TOTICON1	TOTICON0	0	0	0	0	TI05SEL1	TI05SEL0

TI05SEL1	TI05SEL0	TIS051	TIS050	TAU unit0 CH5 input alternate selection
0	0	0	0	P05
0	0	0	1	P82
0	0	1	0	P96
0	1	x	x	Low-speed on-chip clock (f_{IL})
1	0	x	x	Sub system clock (f_{SUB})
1	1	x	x	Main external clock (f_{EX})
Other than the above				Setting prohibited (same as "0000" setting)

Considering the below purposes, every peripheral clock is connected to TI05 of TAU0

- Low-speed on-chip clock: For Frequency Detection of Safefy Function.
- Sub system clock: For the ultra accuracy trimming of high-speed on-chip oscillator Note
- External main clock: For the ultra accuracy trimming of high-speed on-chip oscillator without sub system clock Note

Note Count present operation frequency by timer. It is possible to change trimming code by access HIOTRM register.

TOTICON0	Timer conjunction function control
0	Cut off the connection of TAU unit0 CH1 output to CH0 input
1	Connect TAU unit0 CH1 output to TAU unit0 CH0 input

TOTICON1	Timer conjunction function control
0	Cut off the connection of TAU unit2 CH1 output to CH0 input
1	Connect TAU unit2 CH1 output to TAU unit2 CH0 input

The connection with TOTICONn is used to count external event (pulse) input to TI01/TI21 in long term such as 16-bit counter is overflowed. Timer array unit 0 channel 1/timer array unit 2 channel 1 is worked as divider of input pulse and generates slower pulse to TO01/TO21. Timer array unit 0 channel 0/timer array unit 2 channel 0 is worked as external event counter. Its expected value should be made the calculated value according to timer array unit 0 channel 1/timer array unit 2 channel 1 divider setting.

(4) Serial communication pin select register (STSEL0, STSEL1)

These registers are used for alternate switch of serial input/output pins.

<R>

Figure 4-75. Format of STSEL0 Register

Address: FFF3C After reset: 00H R/W

Symbol	7	<6>	5	<4>	<3>	<2>	<1>	<0>
STSEL0	0	SCSI100	0	SCSI010	SCSI001	SCSI000	SUARTF1	SUARTF0

SUARTF0	Communication pin selection of UARTF0	
	LTxD0	LRxD0
0	P71	P70
1	P15	P14

SUARTF1	Communication pin selection of UARTF1	
	LTxD1	LRxD1
0	P10	P11
1	P131	P132

SCSI001	SCSI000	CSI00 communication pin selection		
		SCK00	SI00	SO00
0	0	P10	P11	P12
0	1	P04	P03	P02
1	0	P34	P33	P32
1	0	P110 Note	P111 Note	P112 Note

Note 128-pin products only (same as "00" setting for other products).

SCSI010	CSI01 communication pin selection		
	SCK01	SI01	SO01
0	P74	P75	P13
1	P56	P55	P54

SCSI100	CSI10 communication pin selection		
	SCK10	SI10	SO10
0	P133	P132	P131
1	P51	P52	P53

Figure 4-76. Format of STSEL1 Register

Address: FFF3D After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL1	SIIC1	SIIC0	0	0	SCAN1	SCAN0	TMCAN1	TMCAN0

TMCAN0	Input source switch of TAU unit1 CH4
0	Input from TI14 (after selected by TIS141~0 bits)
1	TSOUT of aFCAN0 (CAN0 time stamp function)

TMCAN1	Input source switch of TAU unit1 CH5
0	Input from TI15 (after selected by TIS151~0 bits)
1	TSOUT of aFCAN1 (CAN1 time stamp function)

SCAN0	Communication pin selection of aFCAN0	
	CTxD0	CRxD0
0	P71	P70
1	P00	P01

SCAN1	Communication pin selection of aFCAN1	
	CTxD1	CRxD1
0	P62	P63
1	P134	P135

SIIC1	SIIC0	Communication pin selection of IIC11	
		SCL11	SDA11
0	0	P60	P61
0	1	P30	P31
1	0	P136	P50
Other than the above		Setting prohibited	

(5) Sound generator and PCL pin select register (SGSEL)

This register is used for alternate switch of sound generator and PCL output pins.

SGOA output can be stopped when it is not used if SGSEL_2 is set to “1”.

Figure 4-77. Format of SGSEL Register

Address: FFF3F After reset: 00H R/W

<R>

Symbol	7	6	5	4	<3>	2	1	0
SGSEL	0	0	0	0	PCLSEL	SGSEL2	SGSEL1	SGSEL0

SGSEL2	SGSEL1	SGSEL0	Pin select of sound generator outputs	
			SGO/SGOF	SGOA
0	0	0	P73	P72
0	0	1	P93	P92
0	1	0	P135	P134
0	1	1	Setting prohibit	
1	0	0	P73	No port is selected (output disabled)
1	0	1	P93	
1	1	0	P135	
1	1	1	Setting prohibit	

Note: The driving capability of SGO/SGOF alternate pin (P73, P93, P135) is larger than normal buffer.
P93 is also alternated as Stepper-Motor function, so its driving characteristics is the same as SM buffer.
P73 and P135 are the same as SG buffer of 78K0/Dx2.

PCLSEL	PCL output pin selection
0	P75 (default, be available for 48/64/80/100pin)
1	P66 (option for 80/100pin)

(6) RTC1HZ pin select register (RTCSEL)

This register includes two kinds of control function.

- Control of switching RTC1Hz output to TAU TI input.
- Control of switching RTC1Hz output to different port.

Figure 4-78. Format of RTCSEL Register

Address: FFF36 After reset: 00H R/W

<R>

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
RTCSEL	RTCOSEL1	RTCOSEL0	0	0	RTCTIS11	RTCTIS10	RTCTIS01	RTCTIS00

RTCTIS00	Switch RTC1Hz output to TAU TI06 input or not
0	Disconnected to TI06
1	Connected to TI06

RTCTIS01	Switch RTC1Hz output to TAU TI07 input or not
0	Disconnected to TI07
1	Connected to TI07

RTCTIS10	Switch RTC1Hz output to TAU TI16 input or not
0	Disconnected to TI16
1	Connected to TI16

RTCTIS11	Switch RTC1Hz output to TAU TI17 input or not
0	Disconnected to TI17
1	Connected to TI17

RTCOSEL1	RTCOSEL0	RTC1Hz output pin selection
0	0	P64
0	1	P15
1	0	P94
1	1	No port is selected (Output disabled)

To measure 1Hz, two channels of TAU should be used because 16-bit counter will be overflowed if F_{CLK} is fast frequency. A channel is operated in pulse width measurement mode. Low-level or high-level width of 1Hz pulse is typically 500ms. Another channel is operated in interval timer mode (start trigger is set to TIN edge) and number of overflow should be counted by software at the interrupt timing. The measurement is finished when interrupt by capture channel is occurred. The interval time can be calculated by software-overflow-counter and TDR register of capture channel.

4.5.3 The setting to use alternate function

To use the alternate function of a port pin, set the port mode register, output latch, port output mode register, LCD port function register, A/D port configuration register, and Stepper motor port mode control register as shown in Table 4-23.

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (1/16)

(a) Alternate function of P0 (1/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P00	TI00	Input	1	x	0	TIS00.0 = 0 TISELSE.6 = 0	-
	TO00	Output	0	0	0	TOS00.0 = 0	STSEL1.2 = 0
	CTxD0	Output	0	1	0	STSEL1.2 = 1	-
	SEG14	Output	x	x	1	-	-
P01	TI01	Input	1	x	0	TIS00.3,2 = 00	-
	TO01	Output	0	0	0	TOS00.3,2 = 00	-
	CRxD0	Input	1	x	0	STSEL1.2 = 1	-
	SEG15	Output	x	x	1	-	-
P02	TI02	Input	1	x	0	TIS00.4 = 0	-
	TO02	Output	0	0	0	TOS00.4 = 0	STSEL0.3,2 = 00/10 TOS10.4 = 0
	TI12	Input	1	x	0	TIS10.4 = 1	-
	TO12	Output	0	0	0	TOS10.4 = 1	STSEL0.3,2 = 00/10 TOS00.4 = 0
	SO00	Output	0	1	0	STSEL0.3,2 = 01	-
	SEG16	Output	x	x	1	-	-
P03	TI03	Input	1	x	0	TIS00.7,6 = 00	-
	TO03	Output	0	0	0	TOS00.7,6 = 00	TOS10.7,6 = 00/10/11
	TI13	Input	1	x	0	TIS10.7,6 = 01	-
	TO13	Output	0	0	0	TOS10.7,6 = 01	TOS00.7,6 = 01/10/11
	SI00	Input	1	x	0	STSEL0.3,2 = 01	-
	SEG17	Output	x	x	1	-	-
P04	TI04	Input	1	x	0	TIS01.0 = 0	
	TO04	Output	0	0	0	TOS01.0 = 0	STSEL0.3,2 = 00/10 TOS11.1,0 = 00/10
	TI14	Input	1	x	0	TIS11.1,0 = 01 STSEL1.0 = 0	-
	TO14	Output	0	0	0	TOS11.1,0 = 01	STSEL0.3,2 = 00/10 TOS01.0 = 1
	SCK00	Output	0	1	0	STSEL0.3,2 = 01	-
		Input	1	x			
	SEG18	Output	x	x	1	-	-
P05	TI05	Input	1	x	0	TIS01.3,2 = 00 TISELSE.1,0 = 00	-
	TO05	Output	0	0	0	TOS01.3,2 = 00	TOS11.3,2 = 00/10/11
	TI15	Input	1	x	0	TIS11.3,2 = 01 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 01	TOS01.3,2 = 01/10
	SEG19	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (2/16)

(a) Alternate function of P0 (2/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P06	TI06	Input	1	x	0	TIS01.4 = 0 RTCSEL.0 = 0	-
	TO06	Output	0	0	0	TOS01.4 = 0	TOS11.5,4 = 00/10
	TI16	Input	1	x	0	TIS11.5,4 = 01 RTCSEL.2 = 0	-
	TO16	Output	0	0	0	TOS11.5,4 = 01	TOS01.4 = 1
	SEG20	Output	x	x	1	-	-
P07	TI07	Input	1	x	0	TIS01.7,6 = 00 RTCSEL.1 = 0	-
	TO07	Output	0	0	0	TOS01.7,6 = 00	TOS11.7,6 = 00/10/11
	TI17	Input	1	x	0	TIS11.7,6 = 01 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 01	TOS01.7,6 = 01/10
	SEG21	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (3/16)

(b) Alternate function of P1 (1/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P10	INTP4	Input	1	x	0	-	-
	TI10	Input	1	x	0	-	-
	TO10	Output	0	0	0	-	STSEL0.1 = 1 STSEL0.3,2 = 01/10
	LTxD1	Output	0	1	0	STSEL0.1 = 0	STSEL0.3,2 = 01/10
	SCK00	Output	0	1	0	STSEL0.3,2 = 00	STSEL0.1 = 1
		Input	1	x			-
	SEG31	Output	x	x	1	-	-
P11	INTPLR1	Input	1	x	0	STSEL0.1 = 0	-
	TI11	Input	1	x	0	TIS10.3,2 = 00	-
	TO11	Output	0	0	0	TOS10.3,2 = 00	-
	LRxD1	Input	1	x	0	STSEL0.1 = 0	-
	SI00	Input	1	x	0	STSEL0.3,2 = 00	-
	SEG30	Output	x	x	1	-	-
P12	INTP2	Input	1	x	0	-	-
	TI12	Input	1	x	0	TIS10.4 = 0	-
	TO12	Output	0	0	0	TOS10.4 = 0	STSEL0.3,2 = 01/10
	SO00	Output	0	1	0	STSEL0.3,2 = 00	TOS10.4 = 1
	SEG29	Output	x	x	1	-	-
P13	TI13	Input	1	x	0	TIS10.7,6 = 00	-
	TO13	Output	0	0	0	TOS10.7,6 = 00	STSEL0.4 = 1
	SO01	Output	0	1	0	STSEL0.4 = 0	TOS10.7,6 = 01/10/11
	SEG25	Output	x	x	1	-	-
P14	INTPLR0	Input	1	x	0	STSEL0.0 = 1	-
	TI14	Input	1	x	0	TIS11.1,0 = 00 STSEL1.0 = 0	-
	TO14	Output	0	0	0	TOS11.1,0 = 00	-
	LRxD0	Input	1	x	0	STSEL0.0 = 1	-
	SEG24	Output	x	x	1	-	-
P15	TI15	Input	1	x	0	TIS11.3,2 = 00 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 00	RTCSEL.7,6 = 00/10/11 STSEL0.0 = 0
	RTC1HZ	Output	0	0	0	RTCSEL.7,6 = 01	TOS11.3,2 = 01/10/11 STSEL0.0 = 0
	LTxD0	Output	0	1	0	STSEL0.0 = 1	TOS11.3,2 = 01/10/11 RTCSEL.7,6 = 00/10/11
	SEG23	Output	x	x	1	-	-
P16	TI16	Input	1	x	0	TIS11.5,4 = 00 RTCSEL.2 = 0	-
	TO16	Output	0	0	0	TOS11.5,4 = 00	-
	SEG22	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (4/16)

(b) Alternate function of P1 (2/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P17	INTP0	Input	1	x	0	-	-
	TI17	Input	1	x	0	TIS11.7,6 = 00 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 00	STSEL0.3,2 = 01/10
	SEG28	Output	x	x	1	-	-

(c) Alternate function of P2

port	Alternate function		PMxx	Pxx	ADPC (bit 3 to 0)
	Function name	I/O			
P20	AVREFP	Input	-	-	0000/0010 to 1001
	ANI0	Input	-	-	
P21	AVREFM	Input	-	-	0000/0011 to 1001
	ANI1	Input	-	-	
P22	ANI2	Input	-	-	0000/0100 to 1001
P23	ANI3	Input	-	-	0000/0101 to 1001
P24	ANI4	Input	-	-	0000/0110 to 1001
P25	ANI5	Input	-	-	0000/0111 to 1001
P26	ANI6	Input	-	-	0000/1000/1001
P27	ANI7	Input	-	-	0000/1001

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (5/16)

(d) Alternate function of P3

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P30	TI20	Input	1	x	0	TIS20.1,0 = 01 TISELSE.7 = 0	-
	TO20	Output	0	0	0	TOS20.1,0 = 01	STSEL1.7,6 = 00/10
	SCL11	Output	0	1	0	STSEL1.7,6 = 01	TOS20.1,0 = 00/10
	SEG6	Output	x	x	1	-	-
P31	TI21	Input	1	x	0	TIS20.3,2 = 01	-
	TO21	Output	0	0	0	TOS20.3,2 = 01	STSEL1.7,6 = 00/10
	SDA11	I/O	0	1	0	STSEL1.7,6 = 01	TOS20.3,2 = 00/10/11
	SEG7	Output	x	x	1	-	-
P32	TI22	Input	1	x	0	TIS20.5,4 = 01	-
	TO22	Output	0	0	0	TOS20.5,4 = 01	STSEL0.3,2 = 00/01
	SO00	Output	0	1	0	STSEL0.3,2 = 10	TOS20.5,4 = 00/10
<R>	TxD0	Output	0	1	0	STSEL0.3,2 = 10	TOS20.5,4 = 00/10
	SEG8	Output	x	x	1	-	-
P33	TI23	Input	1	x	0	TIS20.7,6 = 01	-
	TO23	Output	0	0	0	TOS20.7,6 = 01	STSEL0.3,2 = 00/01
	SI00	Input	1	x	0	STSEL0.3,2 = 10	TOS20.7,6 = 00/10
<R>	RxD0	Input	1	x	0	STSEL0.3,2 = 10	TOS20.7,6 = 00/10
	SEG9	Output	x	x	1	-	-
P34	TI24	Input	1	x	0	TIS21.1,0 = 01	-
	TO24	Output	0	0	0	TOS21.1,0 = 01	STSEL0.3,2 = 00/01
	SCK00	Output	0	1	0	STSEL0.3,2 = 10	TOS21.1,0 = 00/10
		Input	1	x			-
	SEG10	Output	x	x	1	-	-
P35	TI25	Input	1	x	0	TIS21.3,2 = 10	-
	TO25	Output	0	0	0	TOS21.3,2 = 10	STSEL0.3,2 = 00/01
	SEG11	Output	x	x	1	-	-
P36	TI26	Input	1	x	0	TIS21.5,4 = 01	-
	TO26	Output	0	0	0	TOS21.5,4 = 00	-
	SEG12	Output	x	x	1	-	-
P37	TI27	Input	1	x	0	TIS21.7,6 = 10	-
	TO27	Output	0	0	0	TOS21.7,6 = 01	-
	SEG13	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (6/16)

<R> **(e) Alternate function of P4**

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P41	TI04	Input	1	x	0	TIS01.1,0 = 11	-
	TO04	Output	0	0	0	TOS01.1,0 = 11	-
	STOPST	Output	0	0	0	-	-
P42	TI10	Input	1	x	0	TIS10.1,0 = 10	-
	TO10	Output	0	0	0	TOS10.1,0 = 10	-
	SEG7	Output	x	x	1	-	-
P43	TI22	Input	1	x	0	TIS20.5,4 = 11	-
	TO22	Output	0	0	0	TOS20.5,4 = 11	-
	SEG14	Output	x	x	1	-	-
P44	TI23	Input	1	x	0	TIS20.7,6 = 11	-
	TO23	Output	0	0	0	TOS20.7,6 = 11	-
	SEG15	Output	x	x	1	-	-
P45	SEG53	Output	x	x	1	-	-
P46	$\overline{\text{DBWR}}$	Output	0	1	0	-	-
	SEG27	Output	x	X	1	-	-
P47	$\overline{\text{DBRD}}$	Input	0	1	0	-	-
	SEG26	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (7/16)

(f) Alternate function of P5

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P50	TI02	Input	1	x	0	TIS00.4 = 1	-
	TO02	Output	0	0	0	TOS00.4 = 1	STSEL1.7,6 = 00/01
	SDA11	I/O	0	1	0	STSEL1.7,6 = 10	TOS00.4 = 0
	SEG49	Output	x	x	1	-	-
P51	TI04	Input	1	x	0	TIS01.0 = 1	-
	TO04	Output	0	0	0	TOS01.0 = 1	STSEL0.6 = 0
	SCK10	Output	0	1	0	STSEL0.6 = 1	TOS01.0 = 0
		Input	1	x			-
	SEG50	Output	x	x	1	-	-
P52	TI06	Input	1	x	0	TIS01.4 = 1 RTCSEL.0 = 0	-
	TO06	Output	0	0	0	TOS01.4 = 1	-
	SI10	Input	1	x	0	STSEL0.6 = 1	-
	SEG51	Output	x	x	1	-	-
P53	TI13	Input	1	x	0	TIS10.7,6 = 10	-
	TO13	Output	0	0	0	TOS10.7,6 = 10	STSEL0.6 = 0
	SO10	Output	0	1	0	STSEL0.6 = 1	TOS10.7,6 = 00/01/11
	SEG52	Output	x	x	1	-	-
P54	TI14	Input	1	x	0	TIS11.1,0 = 10 STSEL1.0 = 0	-
	TO14	Output	0	0	0	TOS11.1,0 = 10	
	SO01	Output	0	1	0	STSEL0.4 = 1	
	SEG2	Output	x	x	1	-	-
P55	TI15	Input	1	x	0	TIS11.3,2 = 10 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 10	
	SI01	Input	1	x	0	STSEL0.4 = 1	-
	SEG3	Output	x	x	1	-	-
P56	TI16	Input	1	x	0	TIS11.5,4 = 10 RTCSEL.2 = 0	-
	TO16	Output	0	0	0	TOS11.5,4 = 10	STSEL0.4 = 0
	SCK01	Output	0	1	0	STSEL0.4 = 1	TOS11.5,4 = 00/01
		Input	1	x			-
	SEG4	Output	x	x	1	-	-
P57	TI17	Input	1	x	0	TIS11.7,6 = 11 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 11	-
	SEG5	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (8/16)

(g) Alternate function of P6

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P60	INTP1	Input	1	x	N/A	-	-
	TI20	Input	1	x		TIS20.1,0 = 00 TISELSE.7 = 0	-
	TO20	Output	0	0		TOS20.1,0 = 00	STSEL1.7,6 = 01/10
	SCL11	Output	0	1		STSEL1.7,6 = 00	TOS20.1,0 = 01/10
P61	INTP3	Input	1	x	N/A	-	-
	TI21	Input	1	x		TIS20.3,2 = 00	-
	TO21	Output	0	0		TOS20.3,2 = 00	STSEL1.7,6 = 01/10
	SDA11	I/O	0	1		STSEL1.7,6 = 00	TOS20.3,2 = 01/10/11
P62	TI27	Input	1	x	N/A	TIS21.7,6 = 01	-
	TO27	Output	1	0		TOS21.7,6 = 10	STSEL1.3 = 1
	CTxD1	Output	0	1		STSEL1.3 = 0	TOS21.7,6 = 00/01
P63	TI26	Input	1	x	N/A	TIS21.5,4 = 00	-
	TO26	Output	0	0		TOS21.5,4 = 01	-
	CRxD1	Input	1	x		STSEL1.3 = 0	-
P64	TI11	Input	1	x	N/A	TIS10.3,2 = 11	-
	TO11	Output	0	0		TOS10.3,2 = 11	RTCSEL.7,6 = 01/10/11
	RTC1HZ	Output	0	0		RTCSEL.7,6 = 00	TOS10.3,2 = 00/01/10
P65	TI25	Input	1	x	N/A	TIS21.3,2 = 00	-
	TO25	Output	0	0		TOS21.3,2 = 00	-
P66	TI24	Input	1	x	N/A	TIS21.1,0 = 00	-
	TO24	Output	0	0		TOS21.1,0 = 00	SGSEL.3 = 0
	PCL	Output	0	0		SGSEL.3 = 1	TOS21.1,0 = 01/10

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (9/16)

(h) Alternate function of P7

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P70	INTPLR0	Input	1	x	N/A	STSEL0.0 = 0	-
	TI03	Input	1	x		TIS00.7,6 = 11	-
	TO03	Output	0	0		TOS00.7,6 = 11	-
	CRxD0	Input	1	x		STSEL1.2 = 0	-
	LRxD0	Input	1	x		STSEL0.0 = 0	-
P71	CTxD0	Output	0	1	N/A	STSEL1.2 = 0	STSEL0.0 = 1
	LTxD0	Output	0	1		STSEL0.0 = 0	STSEL1.3 = 1
P72	ADTRG	Input	1	x	0	-	-
	SGOA	Output	0	0	0	SGSEL.2-0 = 000	-
	SEG1	Output	x	x	1	-	-
P73	SGO/SGOF	Output	0	0	0	SGSEL.2-0 = 000	-
	SEG0	Output	x	x	1	-	-
P74	TI23	Input	1	x	0	TIS20.7,6 = 00	-
	TO23	Output	0	0	0	TOS20.7,6 = 00	STSEL0.4 = 1
	SCK01	Output	0	1	0	STSEL0.4 = 0	TOS20.7,6 = 01/10
		Input	1	x			-
	SEG26	Output	x	x	1	-	-
P75	TI22	Input	1	x	0	TIS20.5,4 = 00	-
	TO22	Output	0	0	0	TOS20.5,4 = 00	SGSEL.3 = 1
	PCL	Output	0	0	0	SGSEL.3 = 0	TOS20.5,4 = 01/10
	SI01	Input	1	x	0	STSEL0.4 = 0	-
	SEG27	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (10/16)

(i) Alternate function of P8

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P80	TI01	Input	1	x	0	TIS00.3,2 = 01	-
	TO01	Output	0	0	0	TOS00.3,2 = 01	SMPC.0 = 0
	SM11	Output	0	0	0	SMPC.0 = 1	TOS00.3,2 = 00/10
	SEG32	Output	x	x	1	-	-
P81	TI03	Input	1	x	0	TIS00.7,6 = 01	-
	TO03	Output	0	0	0	TOS00.7,6 = 01	SMPC.0 = 0
	SM12	Output	0	0	0	SMPC.0 = 1	TOS00.7,6 = 00/10/11
	SEG33	Output	x	x	1	-	-
P82	TI05	Input	1	x	0	TIS01.3,2 = 01 TISELSE.1,0 = 00	-
	TO05	Output	0	0	0	TOS01.3,2 = 01	SMPC.0 = 0
	SM13	Output	0	0	0	SMPC.0 = 1	TOS01.3,2 = 00/10
	SEG34	Output	x	x	1	-	-
P83	TI07	Input	1	x	0	TIS01.7,6 = 01 RTCSEL.1 = 0	-
	TO07	Output	0	0	0	TOS01.7,6 = 01	SMPC.0 = 0
	SM14	Output	0	0	0	SMPC.0 = 1	TOS01.7,6 = 00/10
	ZPD14	Input	x	x	0	ZPDS0.3 = 1	-
	SEG35	Output	x	x	1	-	-
P84	TI11	Input	1	x	0	TIS10.3,2 = 01	-
	TO11	Output	0	0	0	TOS10.3,2 = 01	SMPC.1 = 0
	SM21	Output	0	0	0	SMPC.1 = 1	TOS10.3,2 = 00/10/11
	SEG36	Output	x	x	1	-	-
P85	TI13	Input	1	x	0	TIS10.7,6 = 11	-
	TO13	Output	0	0	0	TOS10.7,6 = 11	SMPC.1 = 0
	SM22	Output	0	0	0	SMPC.1 = 1	TOS10.7,6 = 00/01/10
	SEG37	Output	x	x	1	-	-
P86	TI15	Input	1	x	0	TIS11.3,2 = 11 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 11	SMPC.1 = 0
	SM23	Output	0	0	0	SMPC.1 = 1	TOS11.3,2 = 00/01/10
	SEG38	Output	x	x	1	-	-
P87	TI17	Input	1	x	0	TIS11.7,6 = 10 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 10	SMPC.1 = 0
	SM24	Output	0	0	0	SMPC.1 = 1	TOS11.7,6 = 00/01/11
	ZPD24	Input	x	x	0	ZPDS0.7 = 1	-
	SEG39	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (11/16)

(j) Alternate function of P9(1/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P90	TI21	Input	1	x	0	TIS20.3,2 = 11	-
	TO21	Output	0	0	0	TOS20.3,2 = 11	SMPC.2 = 0
	SM31	Output	0	0	0	SMPC.2 = 1	TOS20.3,2 = 00/01/10
	SEG40	Output	x	x	1	-	-
P91	TI23	Input	1	x	0	TIS20.7,6 = 10	-
	TO23	Output	0	0	0	TOS20.7,6 = 10	SMPC.2 = 0
	SM32	Output	0	0	0	SMPC.2 = 1	TOS20.7,6 = 00/01
	SEG41	Output	x	x	1	-	-
P92	TI25	Input	1	x	0	TIS21.3,2 = 01	-
	TO25	Output	0	0	0	TOS21.3,2 = 01	SMPC.2 = 0 SGSEL.2-0 = 000/010/100 to 110
	SM33	Output	0	0	0	SMPC.2 = 1	TOS21.3,2 = 00/10 SGSEL.2-0 = 000/010/100 to 110
	SGOA	Output	0	0	0	SGSEL.2-0 = 001	SMPC.2 = 0 TOS21.3,2 = 00/10
	SEG42	Output	x	x	1	-	-
P93	TI27	Input	1	x	0	TIS21.7,6 = 00	-
	TO27	Output	0	0	0	TOS21.7,6 = 00	SMPC.2 = 0 SGSEL.1,0 = 00/10
	SM34	Output	0	0	0	SMPC.2 = 1	TOS21.7,6 = 01/10 SGSEL.1,0 = 00/10
	ZPD34	Input	x	x	0	ZPDS1.3 = 1	-
	SGO/SGOF	Output	0	0	0	SGSEL.1,0 = 01	TOS21.7,6 = 01/10 SMPC.2 = 0
	SEG43	Output	x	x	1	-	-
P94	TI01	Input	1	x	0	TIS00.3,2 = 10	-
	TO01	Output	0	0	0	TOS00.3,2 = 10	SMPC.3 = 0 RTCSEL.7,6 = 00/01
	RTC1HZ	Output	0	0	0	RTCSEL.7,6 = 10	TOS00.3,2 = 00/01 SMPC.3 = 0
	SM41	Output	0	0	0	SMPC.3 = 1	TOS00.3,2 = 00/01 RTCSEL.7,6 = 00/01
	SEG44	Output	x	x	1	-	-
P95	TI03	Input	1	x	0	TIS00.7,6 = 10	-
	TO03	Output	0	0	0	TOS00.7,6 = 10	SMPC.3 = 0
	SM42	Output	0	0	0	SMPC.3 = 1	TOS00.7,6 = 00/01/11
	SEG45	Output	x	x	1	-	-
P96	TI05	Input	1	x	0	TIS01.3,2 = 10 TISELSE.1,0 = 00	-
	TO05	Output	0	0	0	TOS01.3,2 = 10	SMPC.3 = 0
	SM43	Output	0	0	0	SMPC.3 = 1	TOS01.3,2 = 00/01
	SEG46	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (12/16)

(j) Alternate function of P9 (2/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P97	TI07	Input	1	x	0	TIS01.7,6 = 10 RTCSEL.1 = 0	-
	TO07	Output	0	0	0	TOS01.7,6 = 10	SMPC.3 = 0
	SM44	Output	0	0	0	SMPC.3 = 1	TOS01.7,6 = 00/01
	ZPD44	Input	x	x	0	ZPDS1.7 = 1	-
	SEG47	Output	x	x	1	-	-

<R> **(k) Alternate function of P10**

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P100	TI24	Input	1	x	0	TIS21.1,0 = 11	-
	TO24	Output	0	0	0	TOS21.1,0 = 11	-
	SEG36	Output	x	x	1	-	-
P101	TI25	Input	1	x	0	TIS21.3,2 = 11	-
	TO25	Output	0	0	0	TOS21.3,2 = 11	-
	SEG37	Output	x	x	1	-	-
P102	TI26	Input	1	x	0	TIS21.5,4 = 11	-
	TO26	Output	0	0	0	TOS21.5,4 = 11	-
	SEG38	Output	x	x	1	-	-
P103	TI27	Input	1	x	0	TIS21.7,6 = 11	-
	TO27	Output	0	0	0	TOS21.7,6 = 11	-
	SEG39	Output	x	x	1	-	-
P104	TI01	Input	1	x	0	TIS00.3,2 = 11	-
	TO01	Output	0	0	0	TOS00.3,2 = 11	-
	SEG44	Output	x	x	1	-	-
P105	TI02	Input	1	x	0	TIS00.5,4 = 11	-
	TO02	Output	0	0	0	TOS00.5,4 = 11	-
	SEG45	Output	x	x	1	-	-
P106	TI05	Input	1	x	0	TIS01.3,2 = 11	-
	TO05	Output	0	0	0	TIS01.3,2 = 11	-
	SEG46	Output	x	x	1	-	-
P107	TI06	Input	1	x	0	TIS01.5,4 = 10	-
	TO06	Output	0	0	0	TIS01.5,4 = 10	-
	SEG47	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (13/16)

<R> **(l) Alternate function of P11**

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P110	TI00	Input	1	x	0	TIS00.1,0 = 00	-
	TO00	Output	0	0	0	TOS00.1,0 = 00	STSEL0.3,2 = 00/01/10
	SCK00	Output	0	1	0	STSEL0.3,2 = 11	TOS00.1,0 = 01/10
		Input	1	x			-
	DBD0	I/O	1	0	0	-	-
	SEG35	Output	x	x	1	-	-
P111	TI02	Input	1	x	0	TIS00.5,4 = 11	-
	TO02	Output	0	0	0	TOS00.5,4 = 11	-
	SI00	Input	1	x	0	STSEL0.3,2 = 11	-
	RxD0	Input	1	x	0	STSEL0.3,2 = 11	-
	DBD1	I/O	1	0	0	-	-
	SEG34	Output	x	x	1	-	-
P112	TI04	Input	1	x	0	TIS01.1,0 = 10	-
	TO04	Output	0	0	0	TOS01.1,0 = 10	STSEL0.3,2 = 00/01/10
	SO00	Output	0	1	0	STSEL0.3,2 = 11	TOS01.1,0 = 00/01/11
	TxD0	Output	0	1	0	STSEL0.3,2 = 11	TOS01.1,0 = 00/01/11
	DBD2	I/O	1	0	0	-	-
	SEG33	Output	x	x	1	-	-
P113	TI06	Input	1	x	0	TIS01.5,4 = 11	-
	TO06	Output	0	0	0	TOS01.5,4 = 11	-
	DBD3	I/O	1	0	0	-	-
	SEG32	Output	x	x	1	-	-
P114	TI07	Input	1	x	0	TIS01.7,6 = 11	-
	TO07	Output	0	0	0	TOS01.7,6 = 11	-
	DBD4	I/O	1	0	0	-	-
	SEG31	Output	x	x	1	-	-
P115	TI10	Input	1	x	0	TIS10.1,0 = 01	-
	TO10	Output	0	0	0	TOS10.1,0 = 01	-
	DBD5	I/O	1	0	0	-	-
	SEG30	Output	x	x	1	-	-
P116	TI12	Input	1	x	0	TIS10.5,4 = 11	-
	TO12	Output	0	0	0	TOS10.5,4 = 11	-
	DBD6	I/O	1	0	0	-	-
	SEG29	Output	x	x	1	-	-
P117	TI20	Input	1	x	0	TIS20.1,0 = 11	-
	TO20	Output	0	0	0	TOS20.1,0 = 11	-
	DBD7	I/O	1	0	0	-	-
	SEG28	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (14/16)

(m) Alternate function of P12

port	Alternate function		CMC
	Function name	I/O	
P121	X1	-	CMC.7,6 = 01
P122	X2	-	
	EXCLK	Input	CMC.7,6 = 11
P123	XT1	-	CMC.4 = 1
P124	XT2	-	

<R> **(n) Alternate function of P12**

Port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P125	TI12	Input	1	x	0	TIS10.5,4 = 10	-
	TO12	Output	0	0	0	TOS10.5,4 = 10	-
	SEG25	Output	x	x	1	-	-
P126	TI14	Input	1	x	0	TIS11.1,0 = 11	-
	TO14	Output	0	0	0	TOS11.1,0 = 11	-
	SEG24	Output	x	x	1	-	-
P127	TI16	Input	1	x	0	TIS11.5,4 = 11	-
	TO16	Output	0	0	0	TOS11.5,4 = 11	-
	SEG23	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (15/16)

(o) Alternate function of P13

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P131	TI21	Input	1	x	N/A	TIS20.3,2 = 10	-
	TO21	Output	0	0		TOS20.3,2 = 10	STSEL0.6 = 1 STSEL0.1 = 0
	SO10	Output	0	1		STSEL0.6 = 0	TOS20.3,2 = 00/01/11 STSEL0.1 = 0
	LTxD1	Output	0	1		STSEL0.1 = 1	TOS20.3,2 = 00/01/11 STSEL0.6 = 1
P132	INTPLR1	Input	1	x	N/A	STSEL0.1 = 1	-
	TI20	Input	1	x		TIS20.1,0 = 10 TISELSE.7 = 0	-
	TO20	Output	0	0		TOS20.1,0 = 10	-
	SI10	Input	1	x		STSEL0.6 = 0	-
	LRxD1	Input	1	x		STSEL0.1 = 1	-
P133	TI22	Input	1	x	N/A	TIS20.5,4 = 10	-
	TO22	Output	0	0		TOS20.5,4 = 10	STSEL0.6 = 1
	SCK10	Output	0	1		STSEL0.6 = 0	TOS20.5,4 = 00/01
		Input	1	x			-
P134	TI24	Input	1	x	N/A	TIS21.1,0 = 10	-
	TO24	Output	0	0		TOS21.1,0 = 10	SGSEL.2-0 = 000/001/100 to 110 STSEL1.3 = 0
	SGOA	Output	0	0		SGSEL.2-0 = 010	TOS21.1,0 = 00/01 STSEL1.3 = 0
	CTxD1	Output	0	1		STSEL1.3 = 1	TOS21.1,0 = 00/01 SGSEL.2-0 = 000/001/100 to 110
P135	TI26	Input	1	x	N/A	TIS21.5,4 = 10	-
	TO26	Output	0	0		TOS21.5,4 = 10	SGSEL.1, 0 = 00/01
	SGO/SGOF	Output	0	0		SGSEL.1,0 = 10	TOS21.5,4 = 00/01
	CRxD1	Input	1	x		STSEL1.3 = 1	-
P136	TI00	Input	1	x	0	TIS00.0 = 1 TISELSE.6 = 0	-
	TO00	Output	0	0	0	TOS00.0 = 1	STSEL1.7,6 = 00/01
	SCL11	Output	0	1	0	STSEL1.7,6 = 10	TOS00.0 = 0
	SEG48	Output	x	x	1	-	-

Table 4-23. Settings of Register, and Output Latch When Using Alternate Function (16/16)

(p) Alternate function of P14

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P14	TI11	Input	1	x	N/A	TIS10.3,2 = 10	-
	TO11	Output	0	0		TOS10.3,2 = 10	-

(q) Alternate function of 15

port	Alternate function		PMxx	Pxx	ADPC (bit 3 to 0)
	Function name	I/O			
P150	ANI8	Input	x	x	0000
<R> P151	ANI9	Input	x	x	0000/1011
<R> P152	ANI10	Input	x	x	0000

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 PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78/D1A.

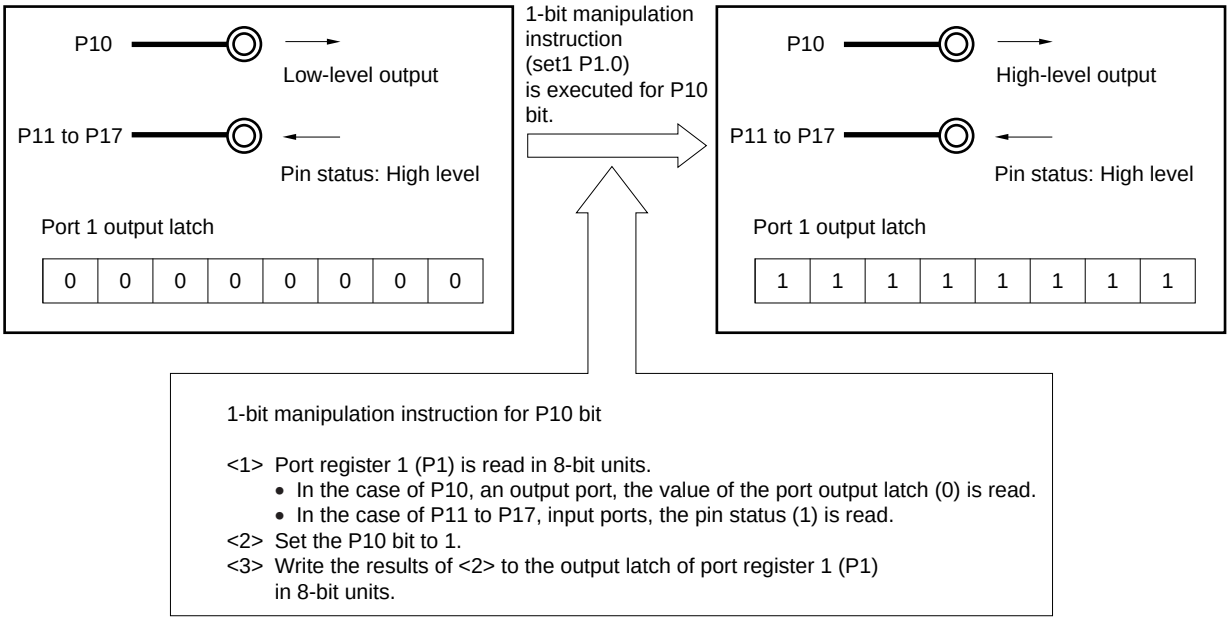
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-79. Bit Manipulation Instruction (P10)



CHAPTER 5 CLOCK GENERATOR

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

<R>

Output pin	128-pin	100-pin	80-pin	64-pin	48-pin
X1, X2 pins	√	√	√	√	√
EXCLK pin	√	√	√	√	√
XT1, XT2 pins	√	√	√	√	N/A

5.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 32, 24, 16, 8, or 4 MHz (typ.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this High-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting the HIOSTOP bit (bit 0 of the CSC register).

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

<R>

(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) to 0.

(3) Subsystem clock

- **XT1 clock oscillator**

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

(4) Low-speed on-chip oscillator clock

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

- Watchdog timer
- Real-time clock

- Interval timer
- LCD controller/driver

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

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

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

f_{IL} itself is controlled by the combination of WDT option bytes and OSMC register and CPU status as shown below. It is not controlled by CLKMB option byte. The other clocks are the same situation. Clock operation control is separated from clock enable settings for peripherals.

f_{IL} can operate continuously and independently of the CPU status and WDT operation. In order to use f_{IL} as continuous clock source for the peripheral hardware, WUTTMCK0 should be set to 1.

CPU status	WDTON (option byte)	WDTSTBYON (option byte)	WUTMMCK0 (OSMC register)	f_{IL} operation
RUN	0	0	0	Stopped
RUN	0	0	1	Operated
RUN	0	1	0	Stopped
RUN	0	1	1	Operated
RUN	1	0	0	Operated
RUN	1	0	1	Operated
RUN	1	1	0	Operated
RUN	1	1	1	Operated
HALT/STOP/SNOOZE	0	0	0	Stopped
HALT/STOP/SNOOZE	0	0	1	Operated
HALT/STOP/SNOOZE	0	1	0	Stopped
HALT/STOP/SNOOZE	0	1	1	Operated
HALT/STOP/SNOOZE	1	0	0	Stopped
HALT/STOP/SNOOZE	1	0	1	Operated
HALT/STOP/SNOOZE	1	1	0	Operated
HALT/STOP/SNOOZE	1	1	1	Operated

5.2 Configuration of Clock Generator

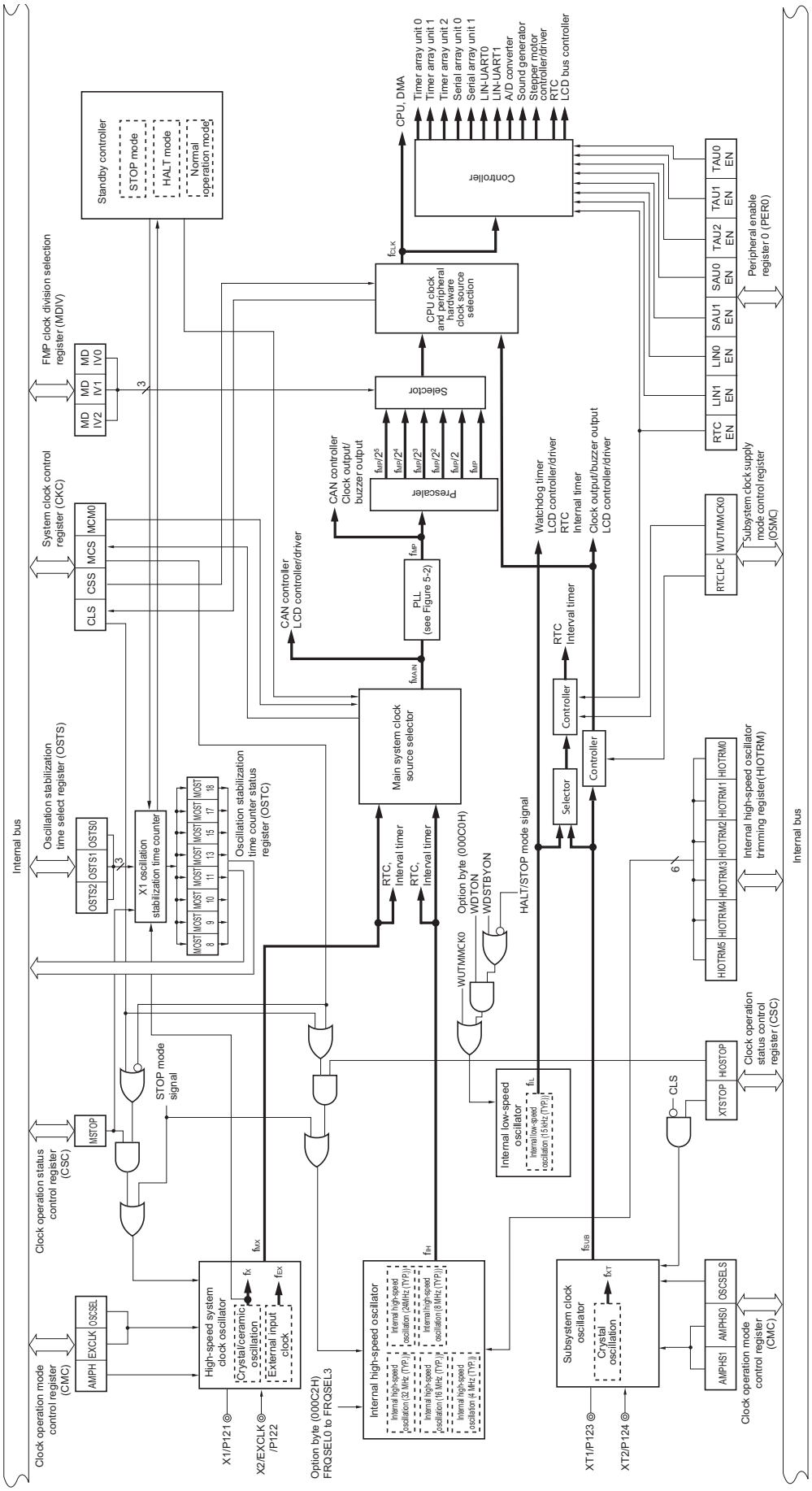
The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 1 (PER0, PER1) Peripheral clock select register(PCKSEL) Operation speed mode control register (OSMC) High-speed on-chip oscillator trimming register (HIOTRM) PLL control register (PLLCTL) PLL status register (PLLSTS) FMP clock division selection register (MDIV)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator

<R>

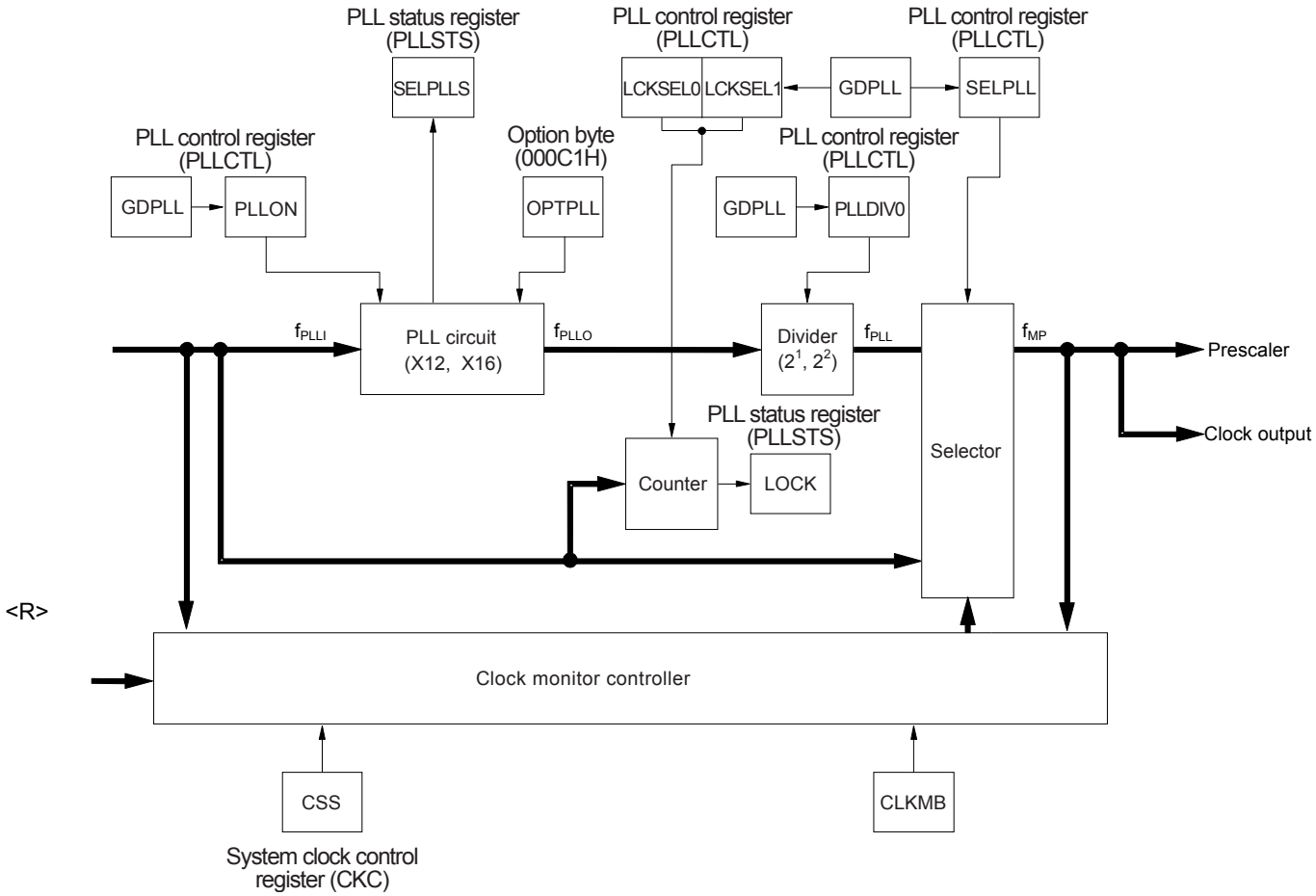
Figure 5-1. Block Diagram of Clock Generator



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

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 5-2. Block Diagram of PLL Circuit



Remark f_{MAIN}: Main system clock
f_{IL}: Low-speed on-chip oscillator clock
f_{PLLI}: PLL input clock
f_{PLLO}: PLL output clock
f_{PLL}: PLL clock
f_{MP}: PLL output for main system clock

5.3 Registers Controlling Clock Generator

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

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Operation speed mode control register (OSMC)
<R>
- High-speed on-chip oscillator trimming register (HIOTRM)
- PLL control register (PLLCTL)
- PLL status register (PLLSTS)
- Peripheral clock select register(PCKSEL)
- FMP clock division selection register (MDIV)

(1) Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/P124 pins, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by 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	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH

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

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

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

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

Cautions

1. **The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.**
2. **After reset release, set the CMC register before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).**
3. **Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.**
4. **Specify the settings for the AMPH, AMPHS1 and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).**
5. **Oscillation stabilization time of f_{XT}, counting on the software.**
6. **Although the maximum system clock frequency is 32 MHz, the maximum frequency of the X1 oscillator is 20 MHz.**

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

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

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

Remark f_X: X1 clock frequency

(2) System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.

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

Reset signal generation sets this register to 00H.

Figure 5-4. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 00H R/W[Note 1]

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

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

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

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

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

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

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

Remark f_{IH}: High-speed on-chip oscillator clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

Cautions 1. Be sure to set bits 3 to 0 of CKC to 0.

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

3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 32 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT) and CHAPTER 33 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT).

(3) Clock operation status control register (CSC)

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

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

Reset signal generation sets this register to C0H.

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

Address: FFFA1H After reset: C0H R/W

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

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

XTSTOP	Subsystem clock operation control	
	XT1 oscillation mode	Input port mode
0	XT1 oscillator operating	Input port
1	XT1 oscillator stopped	

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

Cautions

1. **After reset release, set the clock operation mode control register (CMC) before setting the CSC register.**
2. **Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.**
3. **To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).**
4. **When starting XT1 oscillation by setting the XSTOP bit to 0, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.**
5. **Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the CSC register.**
6. **The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5-2.**

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

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

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

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

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

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

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

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

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

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

Figure 5-6. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

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

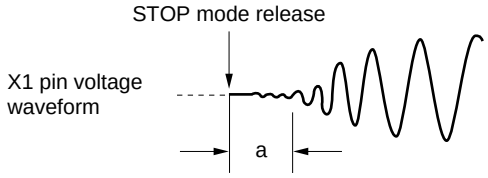
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs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

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

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

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

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



Remark fx: X1 clock oscillation frequency

(5) Oscillation stabilization time select register (OSTS)

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

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

When the high-speed on-chip oscillator clock is selected as the CPU clock, confirm with the oscillation stabilization time counter status register (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 the OSTC register.

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

Reset signal generation sets the OSTS register to 07H.

Figure 5-7. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

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

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

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

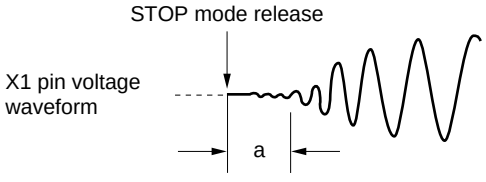
3. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

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

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

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



Remark f_X: X1 clock oscillation frequency

(6) Peripheral enable registers 0, 1 (PER0, PER1)

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

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

- Real-time clock, interval timer
- LIN-UART1
- LIN-UART0
- Serial array unit 1
- Serial array unit 0
- Timer array unit 2
- Timer array unit 1
- Timer array unit 0
- A/D converter
- Sound generator
- Stepper motor controller/driver

<R> - LCD bus controller (128-pin products only)

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 5-8. 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	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W: Bits 0 to 3 and 6 (Read Only)

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
<R> PER1	ADCEN	0	MTRCEN	SGEN	LBEN	0	0	0

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

LIN1EN	Control of serial interface LIN-UART1 input clock supply
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.

Figure 5-8. Format of Peripheral Enable Registers 0, 1 (PER0, PER1) (2/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W: Bits 0 to 3 and 6 (Read Only)

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
<R> PER1	ADCEN	0	MTRCEN	SGEN	LBEN	0	0	0

LIN0EN	Control of LIN-UART0 converter input clock supply
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.

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

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

TAU2EN	Control of serial array unit 2 input clock supply
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	Enables input clock supply. • SFR used by timer array unit 2 can be read and written.

TAU1EN	Control of timer array unit 1 input clock supply
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	Enables input clock supply. • SFR used by timer array unit 1 can be read and written.

Figure 5-8. Format of Peripheral Enable Registers 0, 1 (PER0, PER1) (3/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W (Note: Bits 0 to 3 and 6 are Read Only)

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
<R> PER1	ADCEN	0	MTRCEN	SGEN	LBEN	0	0	0

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

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

MTRCEN	Control of stepper motor controller/driver clock supply
0	Stops input clock supply. • SFR used by the stepper motor controller/driver cannot be written. • The stepper motor controller/driver is in the reset status.
1	Supplies input clock. • SFR used by stepper motor controller/driver can be read and written.

SGEN	Control of sound generator clock supply
0	Stops input clock supply. • SFR used by the sound generator cannot be written. • The sound generator is in the reset status.
1	Supplies input clock. • SFR used by sound generator can be read and written.

<R> LBEN	Control of LCD bus controller clock supply
0	Stops input clock supply. • SFR used by the LCD bus controller cannot be written. • The LCD bus controller is in the reset status.
1	Supplies input clock. • SFR used by LCD bus controller can be read and written.

(7) Operation speed mode control register (OSMC)

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

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock and interval timer, is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock. Set bit 7 (RTCEN) of peripheral enable registers 0 (PER0) to 1 before this setting.

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

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

Reset signal generation clears this register to 00H.

Figure 5-9. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

RTCLPC	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See Table 22-1 for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time clock and interval timer.

WUTMMCK0	Selection of operation clock for real-time clock and interval timer.
0	Other than f_{IL}
1	Low-speed on-chip oscillator clock (f_{IL})

<R>

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

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

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

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

Reset signal generation clears this register to default value (undefined).

Cautions 1. The frequency will vary if the temperature and VDD pin voltage change after accuracy adjustment. When the temperature and VDD voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

2. The optimized value is set by each chip, therefore keep this value unchanged.

Figure 5-10. Format of High-speed on-chip oscillator Trimming Register (HIOTRM)

Address: F00A0H After reset: undefined R/W

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

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	▲
0	0	0	0	1	0	|
0	0	0	0	1	1	|
0	0	0	1	0	0	|
•	•	•	•	•	•	|
•	•	•	•	•	•	|
•	•	•	•	•	•	|
1	1	1	1	1	0	▼
1	1	1	1	1	1	Maximum speed

(9) PLL control register (PLLCTL)

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

Address: F0129H After reset: 00H R/W (Note: Bits 1, 3 and 5 are Read Only)

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1	LCKSEL0	0	PLLDIV0	0	SELPLL	0	PLLON

LCKSEL1	LCKSEL0	Lockup wait counter setting value	Note
0	0	2^7/ f_{MAIN}	Should be selected 40 μs or more (PLL lock time target is 40 μs)
0	1	2^8/ f_{MAIN} (recommended selection of 4 MHz input)	
1	0	2^9/ f_{MAIN} (recommended selection of 8 MHz input)	
1	1	Setting prohibited	

PLLDIV0	PLL output clock (f_{PLLO}) division 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_{PLL})

PLLON	PLL operation control
0	Stop PLL
1	Operates PLL (A lockup wait time is required after the PLL starts operating, so that the frequency stabilizes.)

Note SELPLL setting is only possible when PLLON = 1 and LOCK = 1.
SELPLL is cleared when either PLLON or LOCK is “0”.
When PLLON = 1, changing of PLLDIV0 is prohibited.
When PLLON = 1, changing of f_{MAIN} is prohibited.

Table 5-3. PLL Input/Output Clock Control

Option byte	PLL control register (PLLCTL)	User input frequency	SELPLLS = 1 frequency selection	PLL	
OPTPLL	PLLDIV0			Multiplication ratio(nr/pr)	Generate frequency
0	0	4 MHz	32 MHz	8	32 MHz
1	0	4 MHz	24 MHz	6	24 MHz
0	1	8 MHz	32 MHz	4	32 MHz
1	1	8 MHz	24 MHz	3	24 MHz

Setting value of PLLDIV0 must be related with input frequency. See above table.

The PLL multiplication number (x12 or x16) is set by using bit 5 (OPTPLL) of the option byte (000C2H). See **CHAPTER 28 OPTION BYTE** for details.

(10) PLL status register (PLLSTS)

Figure 5-12. Format of PLL Status Register (PLLSTS)

Address: F0128H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
PLLSTS	LOCK	0	0	0	SELPLLS	0	0	0

LOCK[Note]	PLL lock state
0	Unlocked state
1	Locked state

Note This is set (1) when the lockup wait counter overflows.

SELPLLS	State of the clock mode
0	Clock through mode (f_{MAIN})
1	PLL clock select mode (f_{PLL})

(11) FMP clock selection division register (MDIV)

Figure 5-13. Format of FMP Clock Selection Division Register (MDIV)

Address: F00F8H After reset: 00H R/W[Note1]

Symbol	7	6	5	4	3	2	1	0
MDIV	0	0	0	0	0	MDIV2	MDIV1	MDIV0

MDIV2	MDIV1	MDIV0	Division of PLL clock (f_{MP})
0	0	0	f_{MP} (default)
0	0	1	$f_{MP}/2$
0	1	0	$f_{MP}/2^2$
0	1	1	$f_{MP}/2^3$
1	0	0	$f_{MP}/2^4$
1	0	1	$f_{MP}/2^5$ Note 2
Other than the above			Setting prohibited

Notes 1. Bits 7 to 3 must be set to 0.

2. Setting prohibited if f_{PLL} < 4 MHz.

(12) Peripheral Clock select register (PCKSEL)

Figure 5-14. Format of Peripheral Clock select register (PCKSEL)

Address: F00F2H After reset: 00H @R/W (Note: Bits 1,2 and 7 are Read Only)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PCKSEL	0	CAN MCKE1	CAN MCK1	CAN MCKE0	CAN MCK0	0	0	SGCLK SEL

CANMCKE1	Supply/stop control of clock (bus & operation) of aFCAN unit1
0	Stops supplying clock (bus & operation) of aFCAN unit1
1	Supplies clock (bus & operation) of aFCAN unit1

CANMCK1	Input clock (operation) supply selection of aFCAN unit1
0	f_{MAIN} is supplied
1	f_{MP} is supplied

CANMCKE0	Supplies/stops control of clock (bus & operation) of aFCAN unit0
0	Stops supplying clock (bus & operation) of aFCAN unit0
1	Supplies clock (bus & operation) of aFCAN unit0

CANMCK0	Input clock (operation) supply selection of aFCAN unit0
0	f_{MAIN} is supplied
1	f_{MP} is supplied

SGCLKSEL	Clock (operation) source supply selection of Sound Generator
0	f_{CLK} is supplied
1	$f_{CLK}/2$ is supplied

5.4 Clock monitor (CLM)

The clock monitor uses the low-speed on-chip 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 (RESFCLM) is generated. If the PLL clock stops, an interrupt request signal (INTCLM) is generated. Up to 4 clocks of f_{IL} is necessary to detect stop of Main OSC/PLL. After detection, reset/interrupt request will immediately occurs.

When CLM macro monitors PLL clock (f_{PLL}) and PLL clock stops, clock through is selected (original clock to PLL input), but the FF/flag of SELPLL/SELPLLS itself is not cleared, so it is necessary to reset chip before select PLL clock again.

Table 5-4. Clock Monitor Operation Conditions

Condition			Optionbyte	Clock monitor operation
$f_{CLK}=f_{SUB}$			-	Stop
$f_{CLK}=f_{MP}/2^N$	f_{IL} Stop		-	Stop
	f_{IL} Operation	STOP mode	-	Stop
		During oscillation stabilization after MCM0 setting	-	Stop
		Other than the above	CLKMB=1	Stop
			CLKMB=0	Operation

As described in above table, f_{IL} must be operated to activate CLM.

f_{IL} operation is controled by the combination of below factor.

- WDSTBYON option byte
- WDTON option byte
- WUTMMCK0 bit
- Chip status (RUN/HALT/STOP/SNOOZE)

Please refer to the description of “Clock tree” for detail.

5.5 System Clock Oscillator

5.5.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

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

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

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

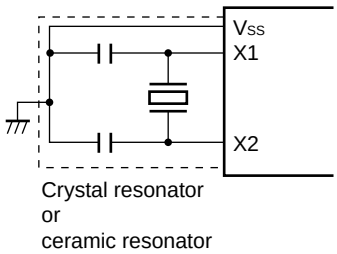
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Chapter 2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**.

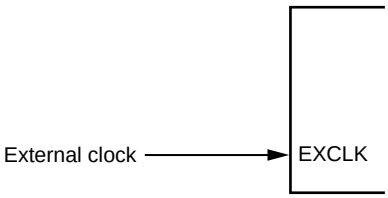
Figure 5-16 shows an example of the external circuit of the X1 oscillator.

Figure 5-15. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

5.5.2 XT1 oscillator

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

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

- Crystal or ceramic oscillation: OSCSELS = 1

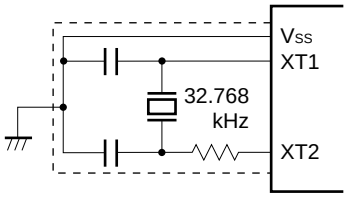
When the XT1 oscillator is not used, set the input port mode (OSCSELS = 0).

When the pins are not used as input port pins, either, see **Chapter 2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**.

Figure 5-17 shows an example of the external circuit of the XT1 oscillator.

Figure 5-16. Example of External Circuit of XT1 Oscillator

(a) Crystal or ceramic oscillation



Caution 1. When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-16 and 5-17 to avoid an adverse effect from wiring capacitance.

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

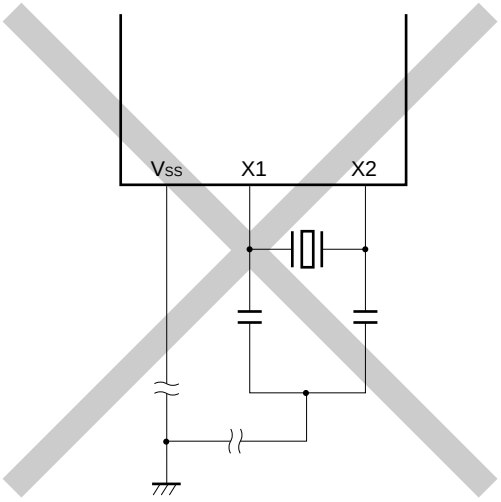
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

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

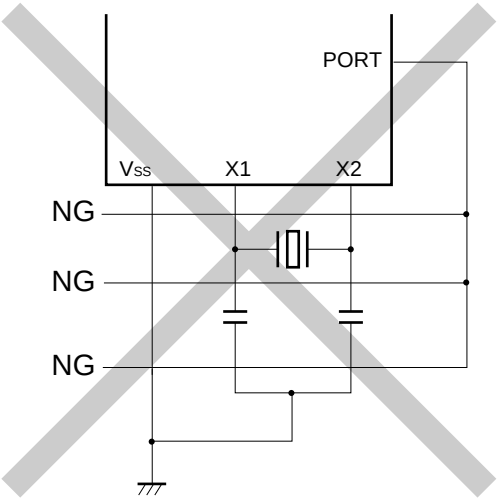
Figure 5-17 shows examples of incorrect resonator connection.

Figure 5-17. Examples of Incorrect Resonator Connection (1/2)

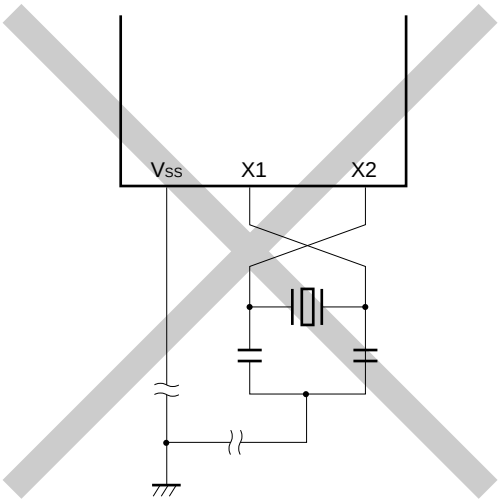
(a) Too long wiring



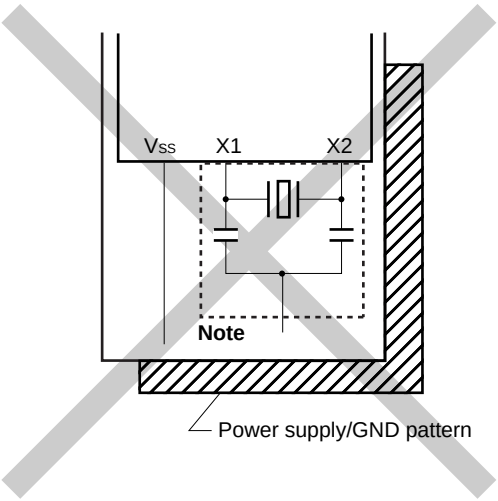
(b) Crossed signal line



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



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

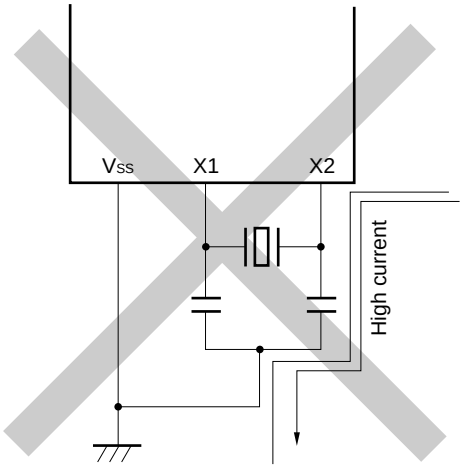


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

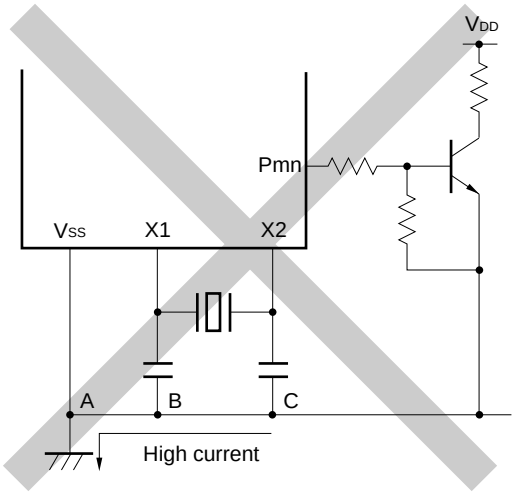
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-17. Examples of Incorrect Resonator Connection (2/2)

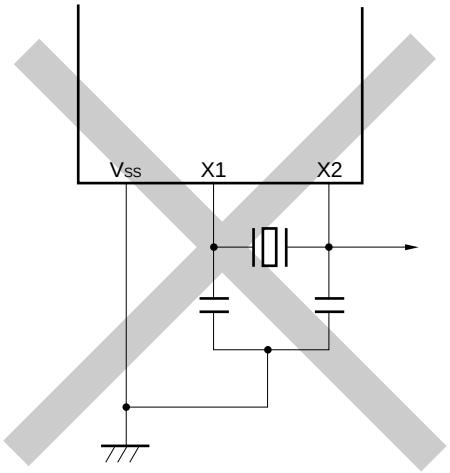
(e) Wiring near high alternating current



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



(g) Signals are fetched



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

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

5.5.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/D1A. The frequency can be selected from among 32, 24, 16, 12, 8, 4, or 1 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC). The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.5.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/D1A.

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

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

Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip oscillator continues. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

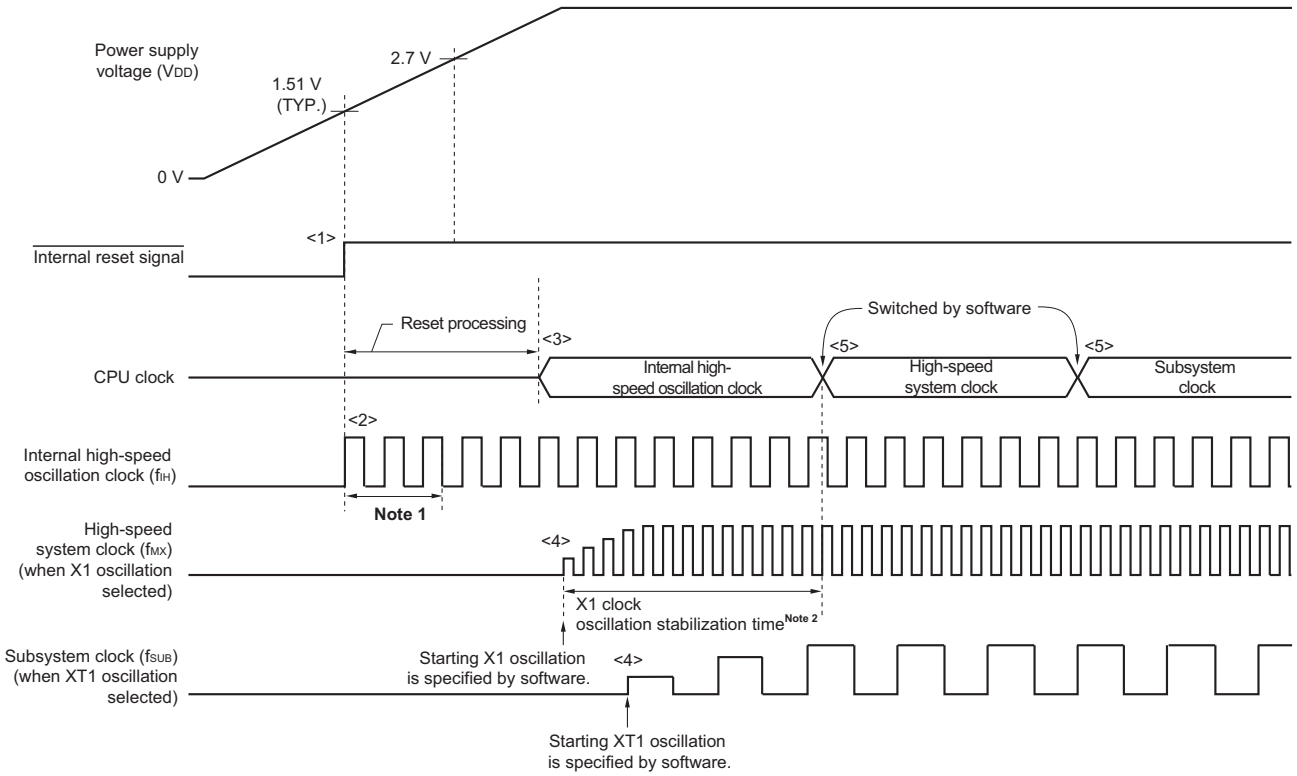
5.6 Clock Generator Operation

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}
  - High-speed on-chip oscillator clock f_{IH}
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/D1A. When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-19.

Figure 5-18. Clock Generator Operation When Power Supply Voltage Is Turned On



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

<2> When the power supply voltage exceeds 1.51 V (TYP.), the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

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

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **5.7.2 Example of setting X1 oscillation clock** and **5.7.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **5.7.2 Example of setting X1 oscillation clock** and **5.7.3 Example of setting XT1 oscillation clock**).

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

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

3. Reset processing time: 497 to 720 μs (When LVD is used)

<R>

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

5.7 Controlling Clock

5.7.1 Example of controlling high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 32, 24, 16, 8, and 4 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)

7	6	5	4	3	2	1	0
CMODE1	CMODE0			FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0
1	1	1	0	0/1	0/1	0/1	0/1

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
1	0	1	0	8 MHz
1	0	1	1	4 MHz
Other than the above				Setting prohibited

<R>

5.7.2 Example of controlling X1 oscillation clock

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

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

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

CMC

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

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

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.

Example: Setting values when a wait of at least 102.4 μs is set based on a 10 MHz resonator.

OSTS

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

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

CSC

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

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

Example: Wait until the bits reach the following values when a wait of at least 102.4 μs is set based on a 10 MHz resonator.

OSTC

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

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

CKC

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

5.7.3 Example of controlling XT1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the XT1 oscillation clock, set the oscillator and start oscillation by using the operation speed mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

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

<1> To run only the real-time clock and interval timer on the subsystem clock (ultra-low current consumption) when in the STOP mode or sub-HALT mode, set the RTCLPC bit to 1.

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

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

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

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

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

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

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

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

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

5.7.4 Example of controlling Peripheral clock

In this product, the unnecessary macro clock is stopped in the root for low power consumption and noise attenuation. The special control registers are configured for the purpose.

Moreover, PCKSEL controls the selection and supply of the operation clock for the asynchronous macro CAN, but the clock selection bit of SG macro is also in this register, bit0 (SGCLKSEL) in order to save address resources.

Peripheral enable register0 (PER0)

Symbol	7	6	5	4	3	2	1	0
PER0	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN
Reset init value	0	0	0	0	0	0	0	0
R/W (hardware)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W (user)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Peripheral enable register1 (PER1)

<R>

Symbol	7	6	5	4	3	2	1	0
PER1	ADCEN	0	MTRCEN	SGEN	LBEN	0	0	0
Reset init value	0	0	0	0	0	0	0	0
R/W (hardware)	R/W	R	R/W	R/W	R/W	R/W	R	R
R/W (user)	R/W	R	R/W	R/W	R/W	R/W	R	R

Peripheral clock select register (PCKSEL)

Symbol	7	6	5	4	3	2	1	0
PCKSEL	0	CAN MCKE1	CAN MCK1	CAN MCKE0	CAN MCK0	0	0	SGCLK SEL
Reset init value	0	0	0	0	0	0	0	0
R/W (hardware)	R	R/W	R/W	R/W	R/W	R	R	R/W
R/W (user)	R	R/W	R/W	R/W	R/W	R	R	R/W

Control contents of PER0, 1

Bit value	Control contents
0	Stops the input clock supply to peripheral macro. SFR of peripheral macro can't be written. (read possible) Peripheral macro is in reset status.
1	Supplies the input clock to peripheral macro. SFR of peripheral macro can be written.

The LCD macro connects directly with f_{IL}, f_{SUB}, and f_{MAIN}, and becomes an asynchronization macro like CAN macro. Inside LCD macro, SCOC bit is used to control LCD sub clock, the low power consumption has been taken into account to LCD source clock division, so the chip peripheral clock control bit PER/PCKSEL is not configured for LCD macro.

Controlled by PER0, 1 registers

Bit name	Control object
TAU0EN	Input clock (bus & operation) supply of TAU unit0 (TM00-07)
TAU1EN	Input clock (bus & operation) supply of TAU unit1 (TM10-17)
TAU2EN	Input clock (bus & operation) supply of TAU unit2 (TM20-27)
SAU0EN	Input clock (bus & operation) supply of SAU unit0 (CSI00,CSI01)
SAU1EN	Input clock (bus & operation) supply of SAU unit1 (CSI10)
LIN0EN	Input clock (bus & operation) supply of LIN-UART0 (UARTF0)
LIN1EN	Input clock (bus & operation) supply of LIN-UART1 (UARTF1)
RTCEN	Input clock (bus) supply of RTC
ADCEN	Input clock (bus & operation) supply of AD converter
SGEN	Input clock (bus & operation) supply of SG
MTRCEN	Input clock (bus & operation) supply of MTRC
<R> LBEN	Input clock (bus & operation) supply of LCD bus controller

Operation clock controlled by PCKSEL register

Bit name	Controlled object
CANMCK0	Input clock (operation) supply selection of aFCAN unit0
CANMCKE0	Supplies/stops control of clock (bus & operation) of aFCAN unit0
CANMCK1	Input clock (operation) supply selection of aFCAN unit1
CANMCKE1	Supply/stop control of clock (bus & operation) of aFCAN unit1
SGCLKSEL	Clock (operation) source supply selection of Sound Generator

Selection and supply control of aFCAN0, 1 operation clock

CANMCKE0/1	CANMCK0/1	Selection and supply of operation clock	Bus clock supply (used for SFR access)
0	x	Clock supply stopped	Clock supply stopped (SFR write is impossible)
1	0	f_{MAIN} is supplied	f_{CLK} is supplied (SFR R/W is possible)
1	1	f_{MP} is supplied Note	f_{CLK} is supplied (SFR R/W is possible)

Note Wake up interrupt can be generated during CAN sleep mode even if CANMCKEn=0.

SG clock source selection

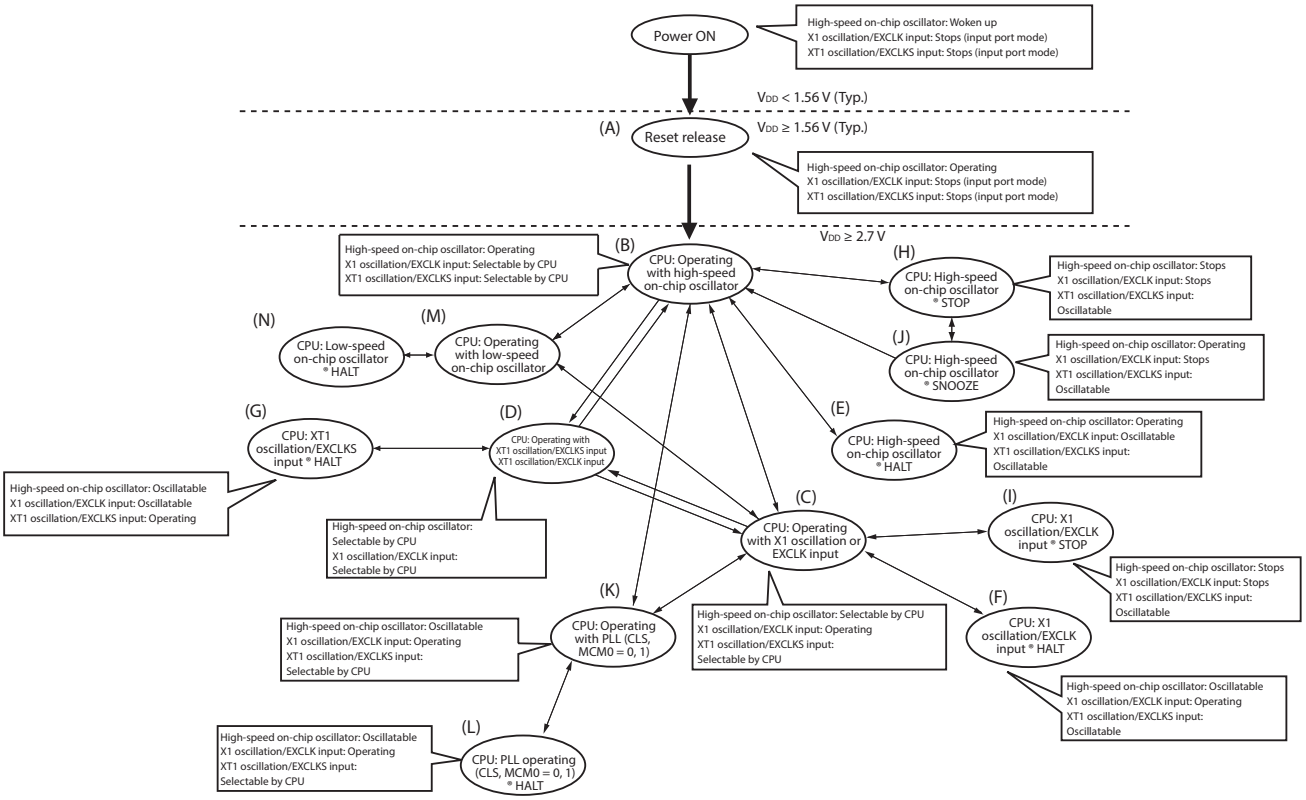
SGEN	SGCLKSEL	Selection of operation clock	Bus clock supply
0	x	Clock supply stopped	Clock supply stopped (SFR write is impossible)
1	0	f_{CLK} is supplied	f_{CLK} is supplied (SFR R/W is possible)
1	1	$f_{CLK}/2$ is supplied	f_{CLK} is supplied (SFR R/W is possible)

5.7.5 CPU clock status transition diagram

Figure 5-19 shows the CPU clock status transition diagram of this product.

<R>

Figure 5-19. CPU Clock Status Transition Diagram



Caution Transitions in the order of (B) → (D) → (C) or (C) → (D) → (B) are prohibited.

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 (1/6)

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

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

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (2/6)

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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	Note 2	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	Note 2	0	Must not be checked	1

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

2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT) and CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT).

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

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

(Setting sequence of SFR registers) →

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

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

Remarks 1. ×: don't care

2. (A) to (J) in Table 5-5 correspond to (A) to (J) in Figure 5-20.

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (3/6)

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

(Setting sequence of SFR registers) →

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

Unnecessary if these registers are already set

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

Notes 1. The clock operation mode control register (CMC) can be changed only once after reset release. This setting is not necessary if it has already been set.

2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT) and CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT).

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note]	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	XTSTOP		CSS
(B) → (D) (XT1 clock)	1	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

Remark (A) to (J) in Table 5-5 correspond to (A) to (J) in Figure 5-20.

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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	30 μs	0

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

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

(Setting sequence of SFR registers) →

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

Unnecessary if the CPU is operating with the subsystem clock

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	CKC Register	
	HIOSTOP	CSS	MCM0
(D) → (B)	0	0	0

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

Unnecessary if this register is already set

Remark (A) to (J) in Table 5-5 correspond to (A) to (J) in Figure 5-20.

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

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

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	OSTS Register	CSC Register	OSTC Register	CKC Register	
		MSTOP		MCM0	CSS
(D) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	**Note**	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	**Note**	0	Must be checked	1	0
(D) → (C) (external main clock)	**Note**	0	Must not be checked	1	0

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

Unnecessary if these registers are already set

Note Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT) and CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT).

(10)

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

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

Remark (A) to (J) in Table 5-5 correspond to (A) to (J) in Figure 5-20.

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

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

(Setting sequence) ⟶

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

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

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

Remark (A) to (J) in Table 5-5 correspond to (A) to (J) in Figure 5-20.

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

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

Table 5-6. Changing CPU Clock (1/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	
X1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
External main system clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).

Table 5-6. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	

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

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

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see Table 5-7 to Table 5-9).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

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

Table 5-7. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	←→	f_{MX}	See Table 5-8
f_{MAIN}	←→	f_{SUB}	See Table 5-9

Table 5-8. Maximum Number of Clocks Required for f_{IH} ↔ f_{MX}

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

Table 5-9. Maximum Number of Clocks Required for f_{MAIN} ↔ f_{SUB}

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

Remarks 1. The number of clocks listed in Table 5-8 to Table 5-9 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-8 to Table 5-9 by removing the decimal portion.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with fIH = 8 MHz, fMX = 10 MHz)

$2\ f_{MX}/f_{IH} = 2\ (10/8) = 2.5 \rightarrow 3$ clocks

5.7.8 Conditions before clock oscillation is stopped

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

Table 5-10. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

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

CHAPTER 6 TIMER ARRAY UNIT

<R>

Product / TAU	RL78/D1A
16-bit timer/unit	8
Timer array unit	3

RL78/D1A has three timer array units, and each unit has eight 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.

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

Timer		Additional function				
Array	Channel	LIN-bus reception	Clock for SM	Division clock for SAU0	Trigger for A/D conversion	Trigger for DMA
Unit0	0					DMA0, DMA1
	1					DMA0, DMA1
	2				Yes	
	3	LIN-UART0				DMA0, DMA1
	4				Yes	
	5					DMA0, DMA1
	6					
	7					DMA0, DMA1
Unit1	0		Yes			DMA0, DMA1
	1	LIN-UART1				DMA0, DMA1
	2					DMA0, DMA1
	3					DMA2, DMA3
	4	LIN-UART0				DMA2, DMA3
	5					DMA2, DMA3
	6					
	7					DMA2, DMA3
Unit2	0	LIN-UART1				DMA2, DMA3
	1					DMA2, DMA3
	2					
	3			Yes		DMA2, DMA3
	4					
	5					DMA2, DMA3
	6					
	7					DMA2, DMA3

6.1 Functions of Timer Array Unit

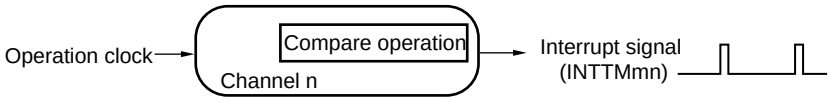
The timer array unit has the following functions.

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, refer to **6.6.1 Overview of single-operation function and combination operation-function**).

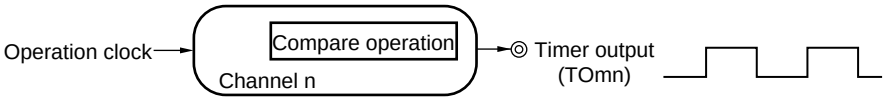
(1) Interval timer

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



(2) Square wave output

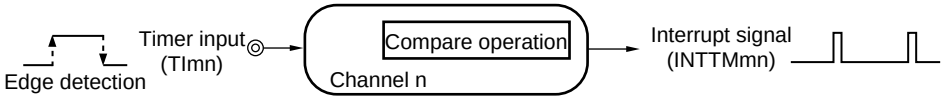
A toggle operation is performed each time INTTMmn is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

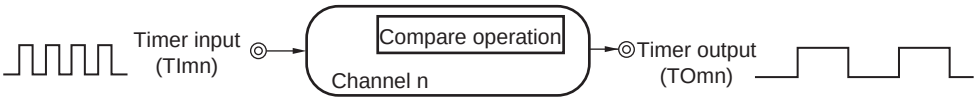
(3) External event counter

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.



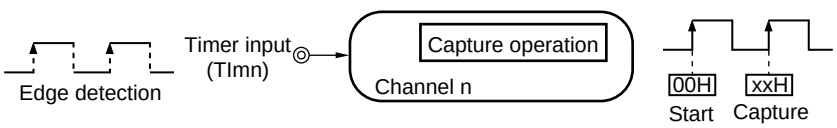
(4) Divider function

A clock input from a timer input pin (TImn) is divided and output from an output pin (TOmn).



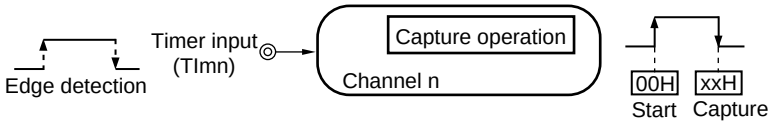
(5) Input pulse interval measurement

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.



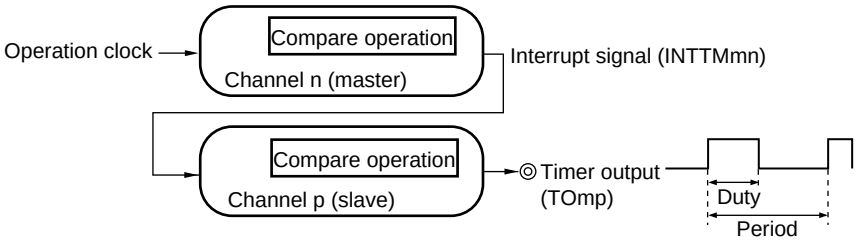
Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

6.1.2 Functions of each channel when it operates with another channel

Combination-operation functions are those functions that are attained by using the master channel (mostly the reference timer that controls cycles) and the slave channels (timers that operate following the master channel) in combination (for details, refer to **6.6.1 Overview of single-operation function and combination-operation function**).

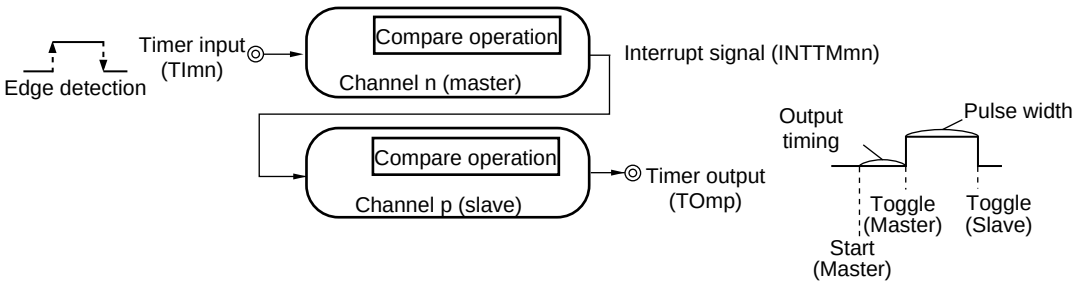
(1) PWM (Pulse Width Modulator) output

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



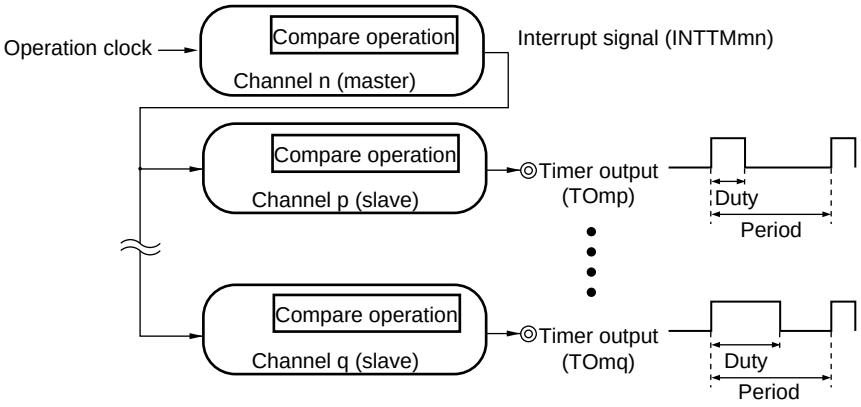
(2) One-shot pulse output

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



(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 (Channel 3 of the timer array unit 0, channels 1 and 4 of the timer array unit 1, and channel 0 of the timer array unit 2 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) 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.

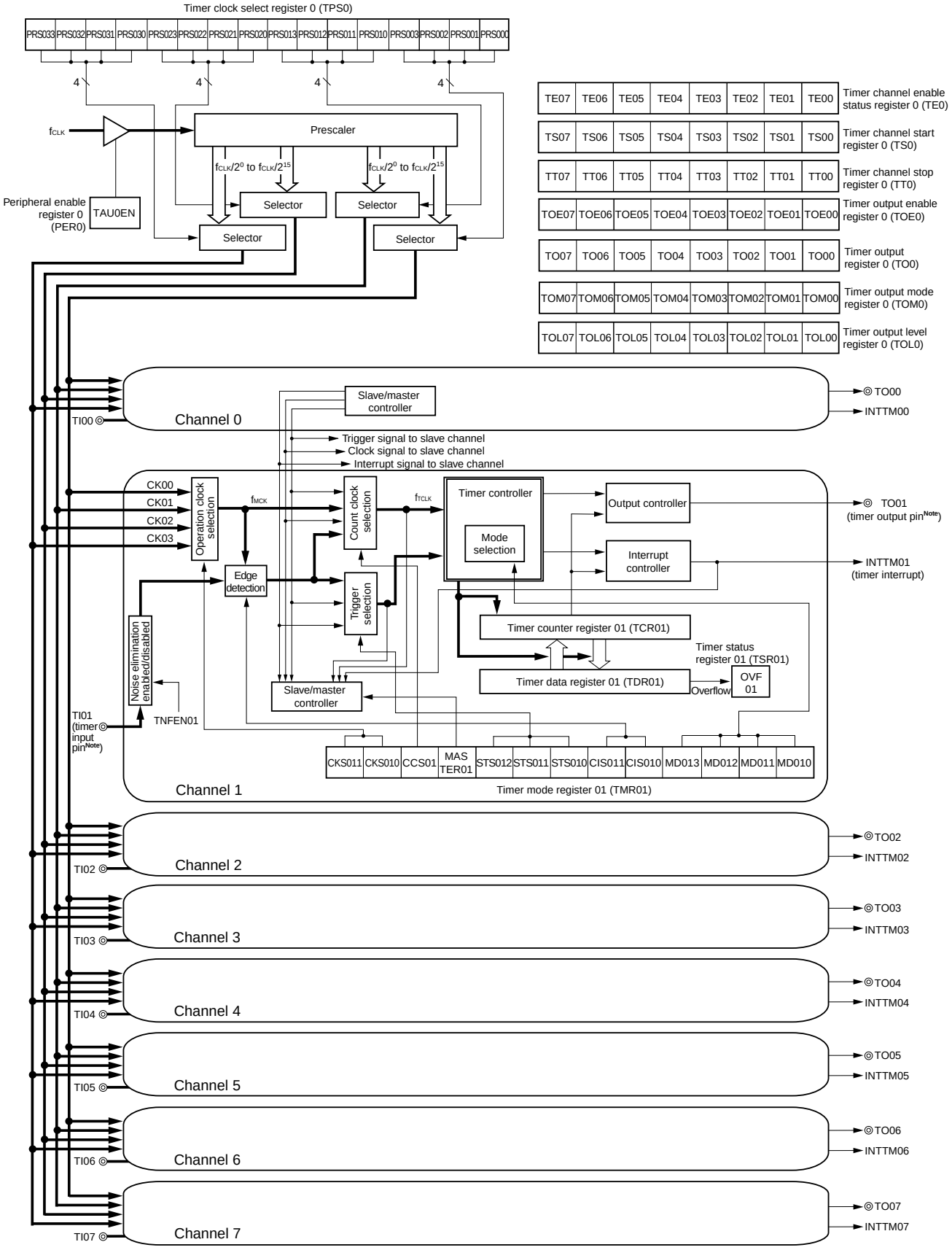
Table 6-1. Configuration of Timer Array Unit

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 0 to 2 (TNFEN0 to TNFEN2) • Sampling clock select register (TNFSMP0 to TNFSMP2) • Noise filter clock select register (TNFCS0 to TNFCS2) • Timer input select registers 00, 01, 10, 11 (TIS00, TIS01, TIS10, TIS11) • Timer output select registers 00, 01, 10, 11 (TOS00, TOS01, TOS10, TOS11) • Serial communication pin select register 1 (STSEL1) • Timer input select else register (TISELSE) • RTC1Hz pin select register (RTCSEL) • Port mode registers 0, 1, 3, 5 to 9, 13, 14 (PM0, PM1, PM3, PM5 to PM9, PM13, PM14) • Port registers 0, 1, 3, 5 to 9, 13, 14 (P0, P1, P3, P5 to P9, P13, P14)

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

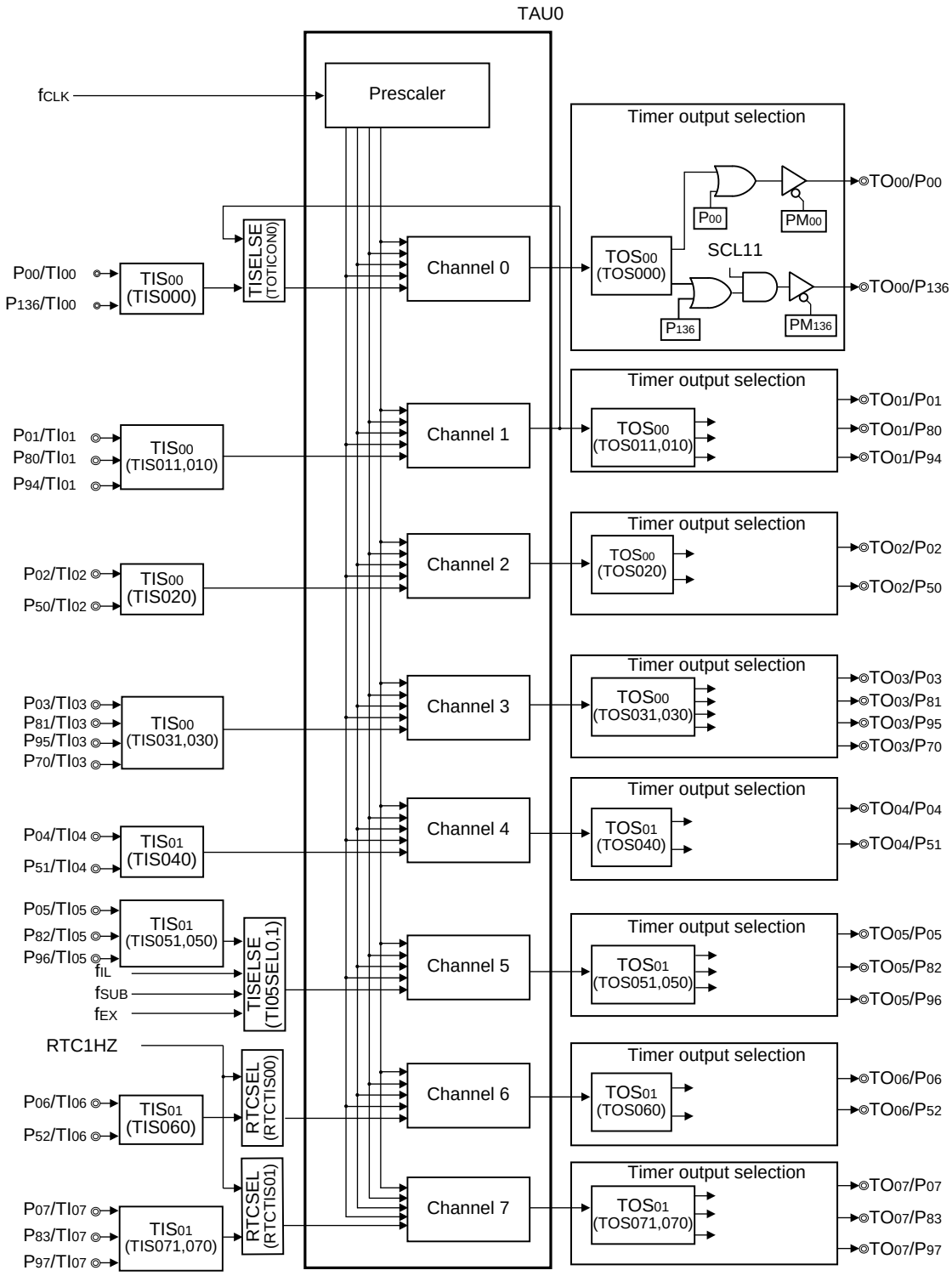
Figure 6-1, Figure 6-3, and Figure 6-5 show the block diagrams.

Figure 6-1. Block Diagram of Timer Array Unit 0



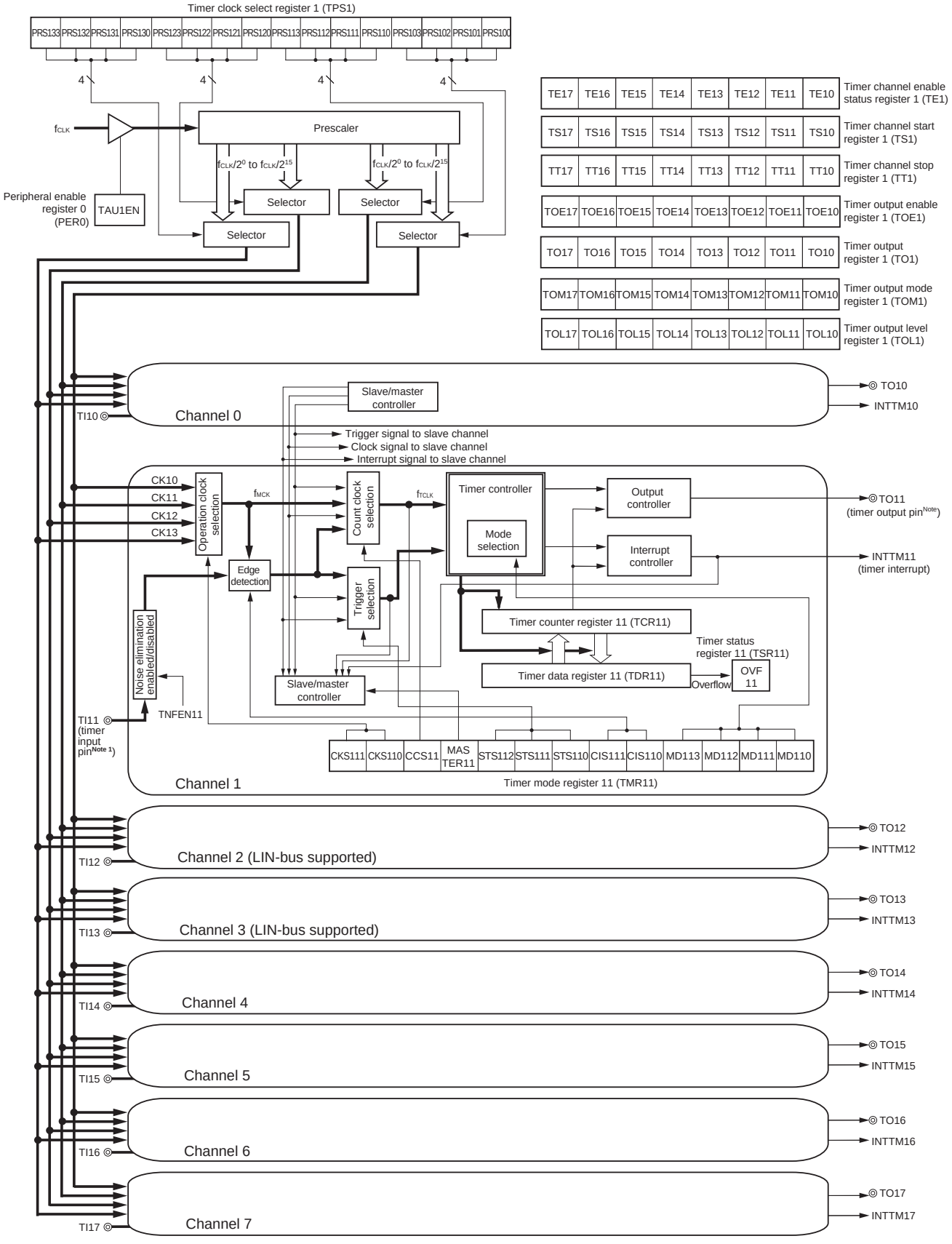
Note See **Figure 6-2** for timer input pin selection and timer output pin selection.

Figure 6-2. Port Configuration Diagram of Timer Array Unit 0



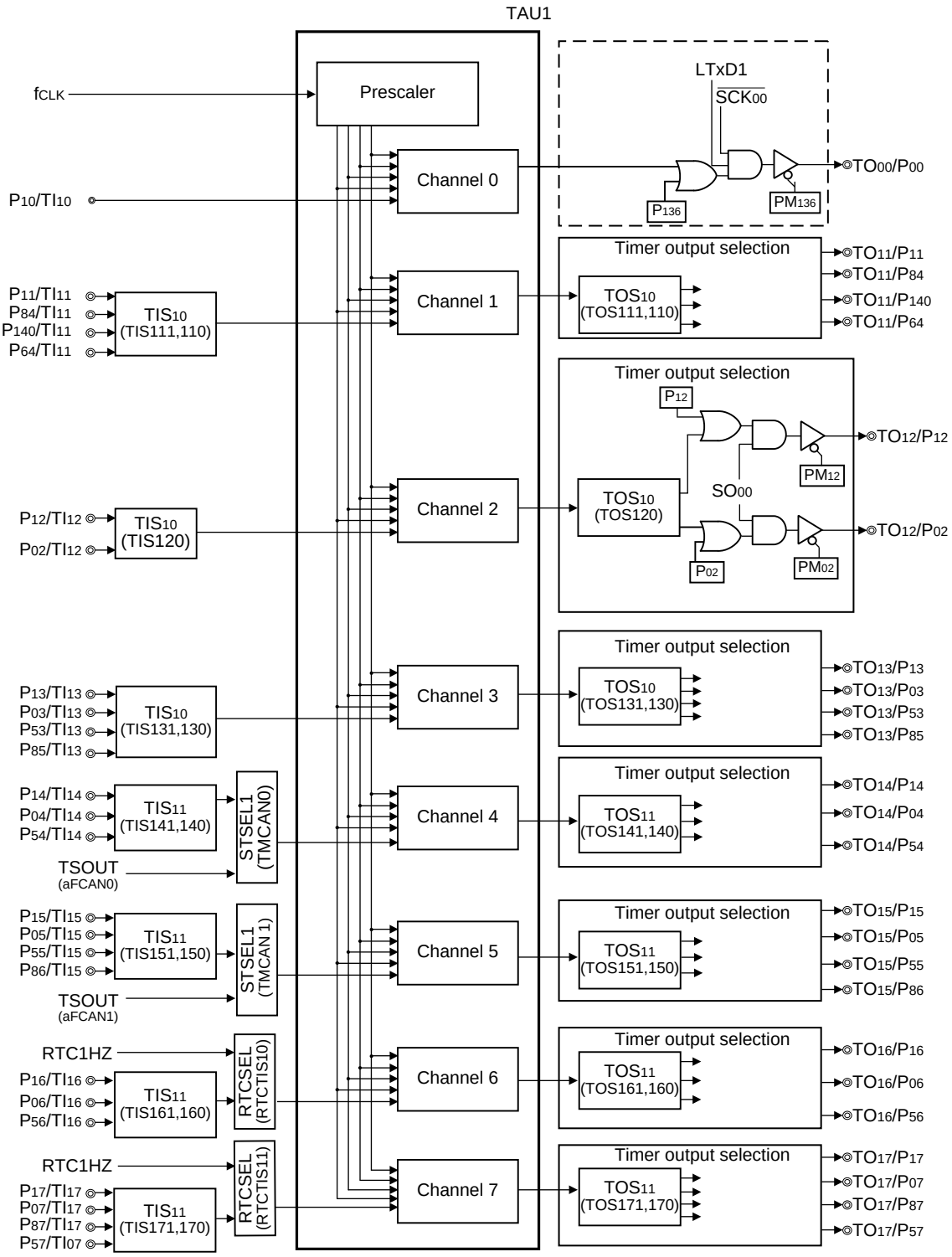
Remark The pins mounted differ depending on the product. See **2.1 Pin Function List, 2.1.5 Pins for each product (pins other than port pins)**

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



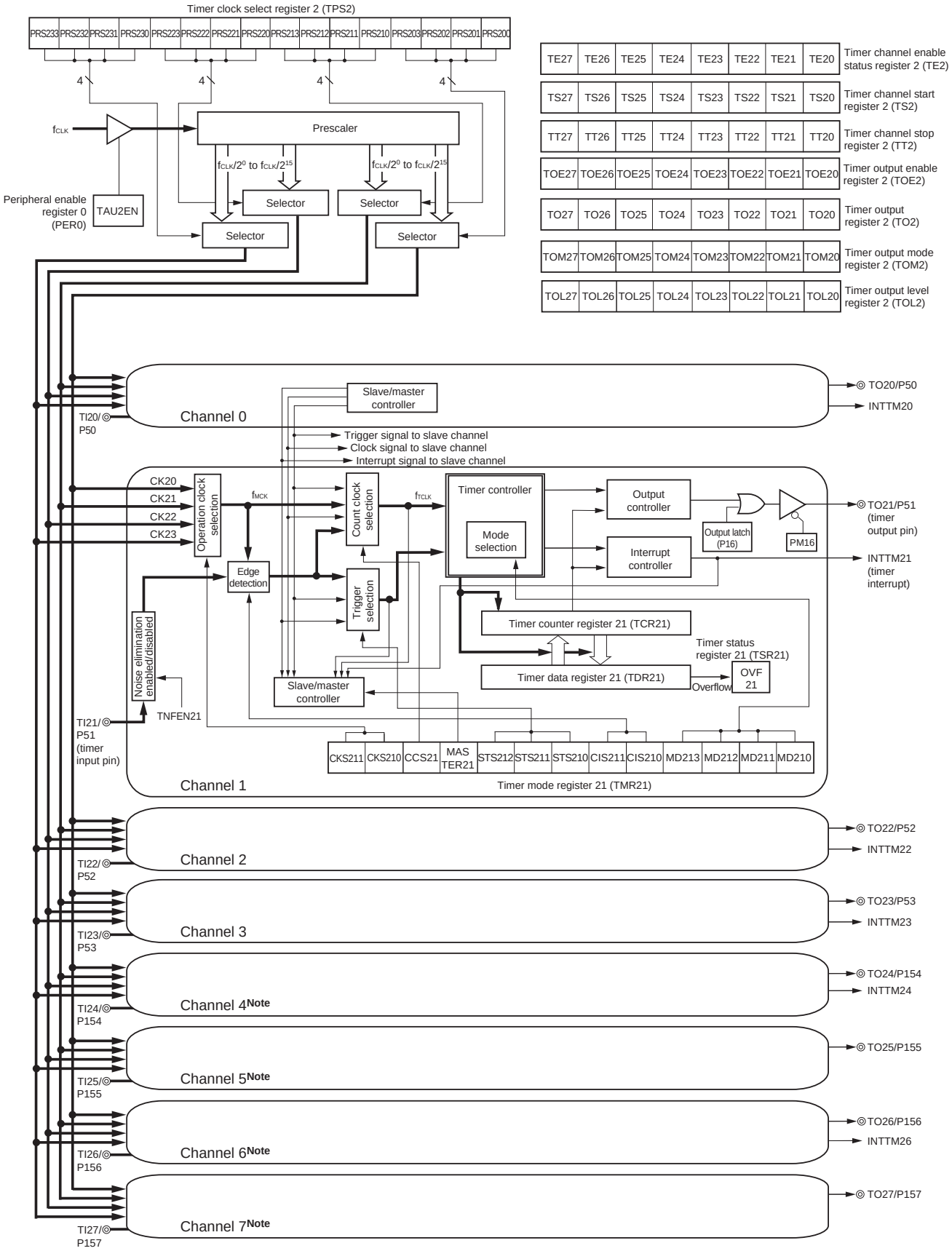
Note See **Figure 6-4** for timer input pin selection and timer output pin selection.

Figure 6-4. Port Configuration Diagram of Timer Array Unit 1



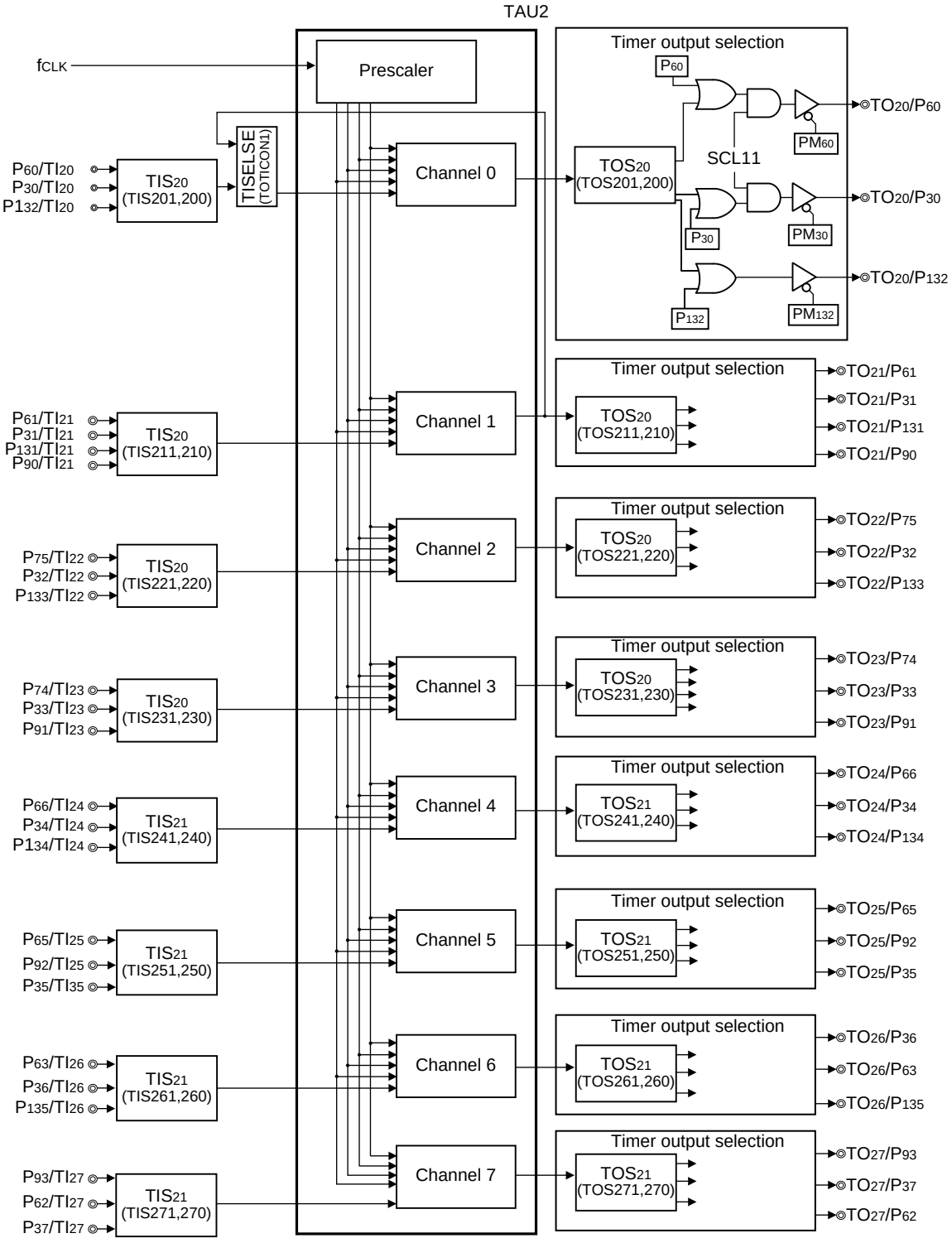
Remark The pins mounted differ depending on the product. See **2.1 Pin Function List, 2.1.5 Pins for each product (pins other than port pins)**

Figure 6-5. Block Diagram of Timer Array Unit 2



Note See **Figure 6-6** for timer input pin selection and timer output pin selection.

Figure 6-6. Port Configuration Diagram of Timer Array Unit 2



Remark The pins mounted differ depending on the product. See **2.1 Pin Function List, 2.1.5 Pins for each product (pins other than port pins)**

(1) Timer counter register mn (TCRmn)

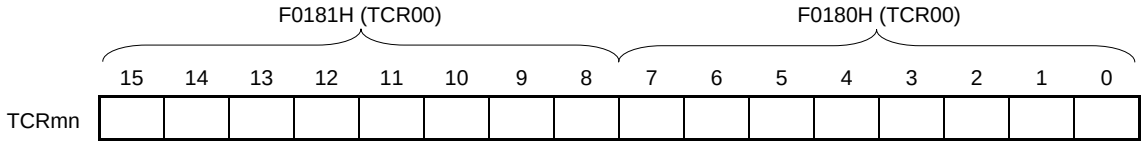
TCRmn is a 16-bit read-only register and is used to count clocks.

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 6-7. 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 to 2)
n: Channel number (n = 0 to 7)

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

Table 6-2. TCRmn Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCRmn Register Read Value[Note]			
		Operation mode change after reset	Operation mode change after count operation paused (TTmn = 1)	Operation restart after count operation paused (TTmn = 1)	During start trigger wait status after one count
Interval timer mode	Count down	FFFFH	Undefined	Stop value	–
Capture mode	Count up	0000H	Undefined	Stop value	–
Event counter mode	Count down	FFFFH	Undefined	Stop value	–
One-count mode	Count down	FFFFH	Undefined	Stop value	FFFFH
Capture & one-count mode	Count up	0000H	Undefined	Stop value	Capture value of TDRmn register + 1

Note The read values of the TCRmn register when TSmn has been set to “1” while TEmn = 0 are shown. The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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.

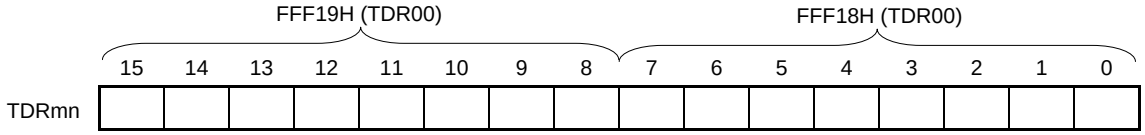
This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Data Register mn (TDRmn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07), FFF70H, FFF71H (TDR10) to FFF7EH, FFF7FH (TDR17), FFF90H, FFF91H (TDR20) to FFF9EH, FFF9FH (TDR27)

After reset: 0000H R/W



(i) When TDRmn is used as compare register

Counting down is started from the value set to TDRmn. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. TDRmn holds its value until it is rewritten.

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

(ii) When 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 to 2)
n: Channel number (n = 0 to 7)

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 0 to 2 (TNFEN0 to TNFEN2)
- Sampling clock select register (TNFSMP0 to TNFSMP2)
- Noise filter clock select register (TNFCS0 to TNFCS2)
- Timer input select registers 00, 01, 10, 11, 20, 21 (TIS00, 01, 10, 11, 20, 21)
- Timer output select registers 00, 01, 10, 11, 20, 21 (TOS00, 01, 10, 11, 20, 21)
- Serial communication pin select register 1 (STSEL1)
- Timer input select else register (TISELSE)
- RTC1Hz pin select register (RTCSEL)
- Port mode registers 0, 1, 3, 5 to 9, 13, 14
- Port registers 0, 1, 3, 5 to 9, 13, 14

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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 6-9. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	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 When setting the timer array unit, be sure to set TAUmEN to 2 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 mn (TISmn), timer output select register mn (TOSmn), noise filter enable registers 0 to 2 (TNFEN0 to TNFEN2), serial communication pin select register 1 (STSEL1), port mode registers 0, 1, 3, 5 to 9, 13, 14).

(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 (TEmn = 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 (TEmn = 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 (TEmn = 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 (TEmn = 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 to 2)
n: Channel number (n = 0 to 7)

Figure 6-10. 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} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	8 MHz	16 MHz	24 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	4 MHz	8 MHz	12 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	2 MHz	4 MHz	6 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	1 MHz	2 MHz	3 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	0.5 MHz	1 MHz	1.5 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	250 kHz	0.5 MHz	750 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	125 kHz	250 kHz	375 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.63 kHz	62.5 kHz	125 kHz	187.5 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	31.25 kHz	62.5 kHz	93.75 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	15.63 kHz	31.25 kHz	46.87 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	7.81 kHz	15.63 kHz	23.43 kHz	31.25 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	3.91 kHz	7.81 kHz	11.71 kHz	15.63 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.95 kHz	3.91 kHz	5.85 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	976 Hz	1.95 kHz	2.92 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	488 Hz	976 Hz	1.46 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	244 Hz	488 Hz	732.42 Hz	976 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 to 2)
n: Channel number (n = 0 to 7)
k = 0 to 3

(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 6-11. 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 to 2)
n: Channel number (n = 0 to 7)

Figure 6-11. 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	MASTERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

MASTER 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	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of 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 the 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 to 2)
n: Channel number (n = 0 to 7)

Figure 6-11. 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	Corresponding function	Count operation of TCR
0	0	0	1/0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Counting down
0	1	0	1/0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	0	Event counter mode	External event counter	Counting down
1	0	0	1/0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than the 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)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation[Note]. At that time, interrupt is also generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
Other than the above		Setting prohibited

Note If the start trigger (TSmn = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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 6-12. 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 to 2)
n: Channel number (n = 0 to 7)

Table 6-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer Operation Mode	OVF	Set/Clear Conditions
• Capture mode	clear	When no overflow has occurred upon capturing
• Capture & one-count mode	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear set	– (Use prohibited, not set/cleared)

Remark The 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.

Figure 6-13. Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H (TE0), F01F0H, F01F1H (TE1), F0230H, F0231H (TE2) — After reset: 0000H — R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	0	0	0	0	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Caution Be sure to clear bits 15 to 8 of TEm to 0.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(6) Timer channel start register m (TSm)

TSm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TSmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is set to 1. TSmn is a trigger bit and cleared immediately when TEmn = 1.

TSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TSm can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6-14. Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H (TS0), F01F2H, F01F3H (TS1), F0232H, F0233H (TS2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	0	0	0	0	TSm7	TSm6	TSm5	TSm4	TSm3	TSm2	TSm1	TSm0

TS mn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEmn is set to 1 and the count operation becomes enabled. The TCRmn count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-4).

Caution Be sure to clear bits 15 to 8 of TSm to 0.

Remarks 1. When the TSm register is read, 0 is always read.

2. m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Table 6-4. Operations from Count Operation Enabled State to TCRmn Count Start

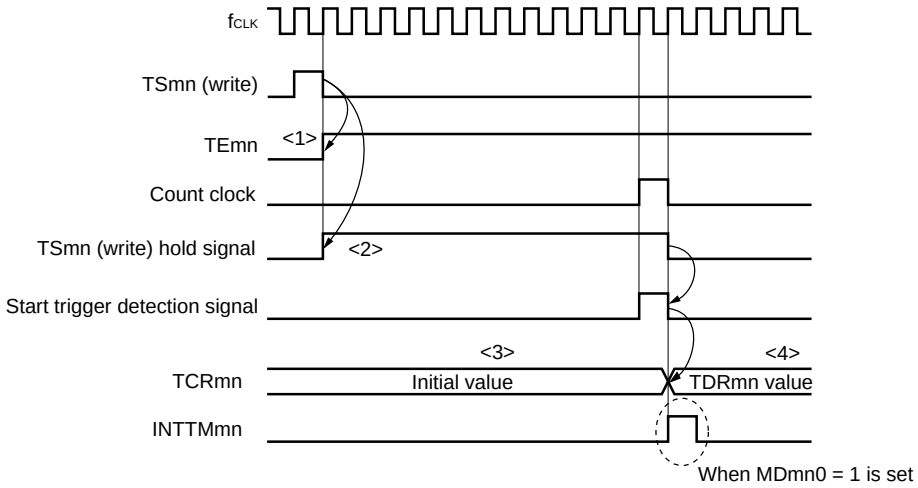
Timer Operation Mode	Operation When TSmn = 1 Is Set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 to TSmn bit loads the value of TDRmn to TCRmn. The subsequent count clock performs count down operation. The external trigger detection selected by STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **6.3 (6) (b) Start timing in event counter mode**).
• Capture mode	No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see **6.3 (6) (c) Start timing in capture mode**).
• One-count mode	When TSmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	When TSmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see **6.3 (6) (e) Start timing in capture & one-count mode**).

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(a) Start timing in interval timer mode

<1> Writing 1 to TSmn sets TEmn = 1.
<2> The write data to TSmn is held until count clock generation.
<3> TCRmn holds the initial value until count clock generation.
<4> On generation of count clock, the “TDRmn value” is loaded to TCRmn and count starts.

Figure 6-15. Start Timing (In Interval Timer Mode)



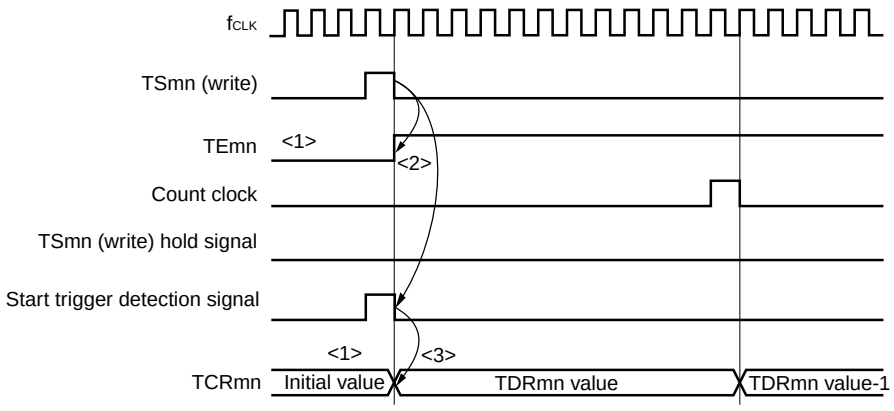
Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(b) Start timing in event counter mode

<1> While TEmn is set to 0, TCRmn holds the initial value.
<2> Writing 1 to TSmn sets 1 to TEmn.
<3> As soon as 1 has been written to TSmn and 1 has been set to TEmn, 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 6-16. Start Timing (In Event Counter Mode)

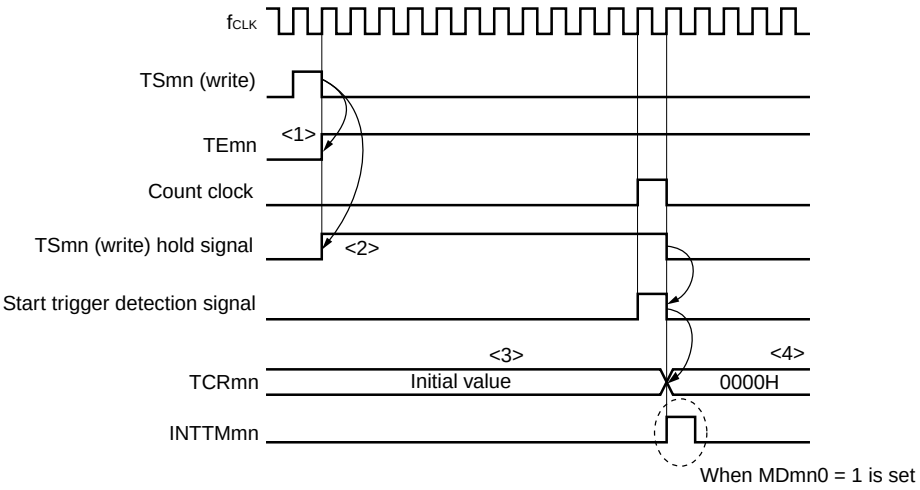


Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(c) Start timing in capture mode

<1> Writing 1 to TSmn sets TEmn = 1.
<2> The write data to TSmn is held until count clock generation.
<3> TCRmn holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCRmn and count starts.

Figure 6-17. Start Timing (In Capture Mode)



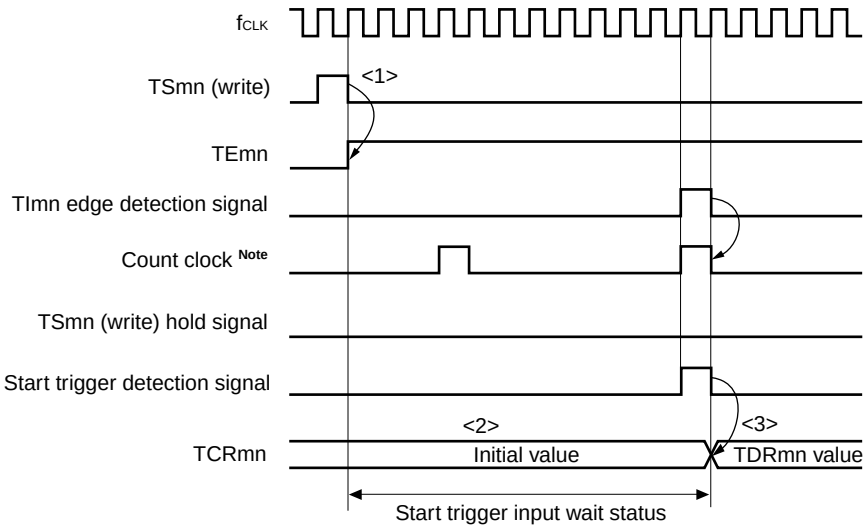
Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(d) Start timing in one-count mode

<1> Writing 1 to TSmn sets TEmn = 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.

Figure 6-18. Start Timing (In One-count Mode)



Note When the one-count mode is set, the operation clock (f_{MCK}) is selected as count clock (CCSmn = 0).

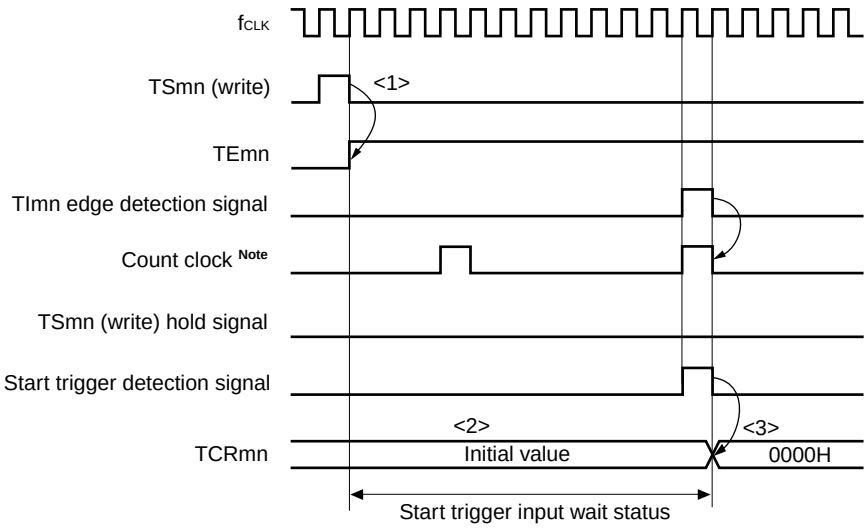
Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TImn is used).

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(e) Start timing in capture & one-count mode

<1> Writing 1 to TSmn sets TEmn = 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.

Figure 6-19. Start Timing (In Capture & One-count Mode)



Note When the capture & one-count mode is set, the operation clock (f_{MCK}) is selected as count clock (CCSmn = 0).

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 to 2)
n: Channel number (n = 0 to 7)

(7) Timer channel stop register m (TTm)

TTm is a trigger register that is used to clear a timer counter (TCRmn) and stop the counting operation of each channel.

When a bit (TTmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is cleared to 0. TTmn is a trigger bit and cleared to 0 immediately when TEmn = 0.

TTm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TTm can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 6-20. Format of Timer Channel Stop Register m (TTm)

Address: F01B4H, F01B5H (TT0), F01F4H, F01F5H (TT1) F0234H, F0235H (TT2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	0	0	0	0	TTm7	TTm6	TTm5	TTm4	TTm3	TTm2	TTm1	TTm0

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated).

Caution Be sure to clear bits 15 to 8 of TTm to 0.

Remarks 1. When the TTm register is read, 0 is always read.

2. m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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.

Figure 6-21. Format of Timer Output Enable Register m (TOEm)

Address: F01BAH, F01BBH (TOE0), F01FAH, F01FBH (TOE1), F023AH, F023BH (TOE2), After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Timer output enable/disable of channel n
0	The TOmn operation stopped by count operation (timer channel output bit). Writing to the TOmn bit is enabled. The TOmn pin functions as data output, and it outputs the level set to the TOmn bit. The output level of the TOmn pin can be manipulated by software.
1	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 TOEmn is set or reset depending on the timer operation. The TOmn pin outputs the square-wave or PWM depending on the timer operation.

Caution Be sure to clear bits 15 to 8 of TOEm to 0.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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 (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the pins sharing timer output as port function pins, set the corresponding TOmn bit to “0”.

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 6-22. Format of Timer Output Register m (TOm)

Address: F01B8H, F01B9H (TO0), F01F8H, F01F9H (TO1), F0238H, F0239H (TO2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	TOm7	TOm6	TOm5	TOm4	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 of TOm to 0.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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 (TOEmn = 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 6-23. Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH (TOL0), F01FCH, F01FDH (TOL1), F023CH, F023DH (TOL2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	TOLm0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Inverted output (active-low)

Caution Be sure to clear bits 15 to 8 of TOLm 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 to 2)
n: Channel number (n = 0 to 7)

(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 (TOEmn = 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 6-24. Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH (TOM0), F01FEH, F01FFH (TOM1), F023EH, F023FH (TOM2), After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	TOMm7	TOMm6	TOMm5	TOMm4	TOMm3	TOMm2	TOMm1	TOMm0

TOMmn	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 Be sure to clear bits 15 to 8 of TOMm to 0.

Remark m: Unit number (m = 0 to 2), n: Channel number, p: Slave channel number

(12) Noise filter enable registers 0 to 2 (TNFEN0 to TNFEN2)

TNFEN0 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.

TNFEN1 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.

TNFEN2 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 CPU/peripheral hardware clock (f_{MCK}), and synchronizes them. When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{MCK}).

TNFEN0 to TNFEN2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 6-25. Format of Noise Filter Enable Register 0 (TNFEN0)

Address: F0060H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFEN0	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 6-26. Format of Noise Filter Enable Register 1 (TNFEN1)

Address: F0064H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFEN1	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 6-27. Format of Noise Filter Enable Register 2 (TNFEN2)

Address: F0068H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFEN2	TNFEN27	TNFEN26	TNFEN25	TNFEN24	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

(13) Sampling clock select register (TNFSMP0 to TNFSMP2)

Figure 6-28. Format of sampling clock select register (TNFSMP0 to TNFSMP2)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFSMP0	NFSMP013	NFSMP012	NFSMP011	NFSMP010	NFSMP003	NFSMP002	NFSMP001	NFSMP000

Address: F0065H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFSMP1	NFSMP113	NFSMP112	NFSMP111	NFSMP110	NFSMP103	NFSMP102	NFSMP101	NFSMP100

Address: F0069H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFSMP2	NFSMP213	NFSMP212	NFSMP211	NFSMP210	NFSMP203	NFSMP202	NFSMP201	NFSMP200

n = 0, 1

NFSMPmn3	NFSMPmn2	NFSMPmn1	NFSMPmn0	Clock select
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	f_{MAIN}
1	0	1	1	$f_{MAIN}/2$
1	1	0	0	$f_{MAIN}/2^2$
1	1	0	1	$f_{MAIN}/2^3$
1	1	1	0	$f_{MAIN}/2^4$
1	1	1	1	f_{IL}

Note In fact, NF clock is gated by TAUxEN. If not use TAU, NE is also disabled.

(14) Noise filter clock select register (TNFCS0 to TNFCS2)

Figure 6-29. Format of noise filter clock select register (TNFCS0 to TNFCS2)

Address: F0062H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFCS0	TNFCS07	TNFCS06	TNFCS05	TNFCS04	TNFCS03	TNFCS02	TNFCS01	TNFCS00

Address: F0066H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFCS1	TNFCS17	TNFCS16	TNFCS15	TNFCS14	TNFCS13	TNFCS12	TNFCS11	TNFCS10

Address: F006AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TNFCS2	TNFCS27	TNFCS26	TNFCS25	TNFCS24	TNFCS23	TNFCS22	TNFCS21	TNFCS20

TNFCSmn	Noise filter clock for TImn (m:TAU unit Number (0 to 2), n: channel Number (0 to 7))
0	NFSMPm0 selected by TNFSMPm0[3:0]
1	NFSMPm1 selected by TNFSMPm1[3:0]

(15) Timer input select register (TIS00, TIS01, TIS10, TIS11, TIS20, TIS21)

These registers are used for alternate switch of TAU input pins. TIS00 and TIS01 is for TAU unit0, TIS10 and TIS11 for TAU unit1, TIS20 and TIS21 for TAU unit2.

Figure 6-30. Format of Timer Input Select Register 0 (TIS00, TIS01) (1/2)

Address: F0070H After reset: 00H R/W (Note: Bits 1 and 5 are read only bit)

Symbol	7	6	5	4	3	2	1	0
TIS00	TIS031	TIS030	0	TIS020	TIS011	TIS010	0	TIS000

Address: F0071H After reset: 00H R/W (Note: Bits 1 and 5 are read only bit)

Symbol	7	6	5	4	3	2	1	0
TIS01	TIS071	TIS070	0	TIS060	TIS051	TIS050	0	TIS040

TIS000	TI00 (TAU unit0 CH0) alternate pin selection
0	P00
1	P136

TIS011	TIS010	TI01 (TAU unit0 CH1) alternate pin selection
0	0	P01
0	1	P80
1	0	P94
Other than the above		Setting prohibited (same as “00” setting)

TIS020	TI02 (TAU unit0 CH2) alternate pin selection
0	P02
1	P50

TIS031	TIS030	TI03 (TAU unit0 CH3) alternate pin selection
0	0	P03
0	1	P81
1	0	P95
1	1	P70

TIS040	TI04 (TAU unit0 CH4) alternate pin selection
0	P04
1	P51

TIS051	TIS050	TI05 (TAU unit0 CH5) alternate pin selection
0	0	P05
0	1	P82
1	0	P96
Other than the above		Setting prohibited (same as “00” setting)

TIS060	TI06 (TAU unit0 CH6) alternate pin selection
0	P06
1	P52

Figure 6-30. Format of Timer Input Select Register 0 (TIS00, TIS01) (2/2)

TIS071	TIS070	TI07 (TAU unit0 CH7) alternate pin selection
0	0	P07
0	1	P83
1	0	P97
Other than the above		Setting prohibited (same as “00” setting)

Figure 6-31. Format of TIS10 and TIS11 Registers (1/2)

Address: F0072H After reset: 00H R/W: (Note: Bits 0, 1 and 5 are read only bit.)

Symbol	7	6	5	4	3	2	1	0
TIS10	TIS131	TIS130	0	TIS120	TIS111	TIS110	0	0

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS11	TIS171	TIS170	TIS161	TIS160	TIS151	TIS150	TIS141	TIS140

TIS111	TIS110	TI11 (TAU unit1 CH1) alternate pin selection
0	0	P11
0	1	P84
1	0	P140
1	1	P64

TIS120	TI12 (TAU unit1 CH2) alternate pin selection
0	P12
1	P02

TIS131	TIS130	TI13 (TAU unit1 CH3) alternate pin selection
0	0	P13
0	1	P03
1	0	P53
1	1	P85

TIS141	TIS140	TI14 (TAU unit1 CH4) alternate pin selection
0	0	P14
0	1	P04
1	0	P54
Other than the above		Setting prohibited (same as “00” setting)

TIS151	TIS150	TI15 (TAU unit1 CH5) alternate pin selection
0	0	P15
0	1	P05
1	0	P55
1	1	P86

TIS161	TIS160	TI16 (TAU unit1 CH6) alternate pin selection
0	0	P16
0	1	P06
1	0	P56
Other than the above		Setting prohibited (same as “00” setting)

Figure 6-31. Format of TIS10 and TIS11 Registers (2/2)

TIS171	TIS170	TI17 (TAU unit1 CH7) alternate pin selection
0	0	P17
0	1	P07
1	0	P87
1	1	P57

Figure 6-32. Format of TIS20 and TIS21 Registers (1/2)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS20	TIS231	TIS230	TIS221	TIS220	TIS211	TIS210	TIS201	TIS200

Address: F0075H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS21	TIS271	TIS270	TIS261	TIS260	TIS251	TIS250	TIS241	TIS240

TIS201	TIS200	TI20 (TAU unit2 CH0) alternate pin selection
0	0	P60
0	1	P30
1	0	P132
Other than the above		Setting prohibited (same as “00” setting)

TIS211	TIS210	TI21 (TAU unit2 CH1) alternate pin selection
0	0	P61
0	1	P31
1	0	P131
1	1	P90

TIS221	TIS220	TI22 (TAU unit2 CH2) alternate pin selection
0	0	P75
0	1	P32
1	0	P133
Other than the above		Setting prohibited (same as “00” setting)

TIS231	TIS230	TI23 (TAU unit2 CH3) alternate pin selection
0	0	P74
0	1	P33
1	0	P91
Other than the above		Setting prohibited (same as “00” setting)

TIS241	TIS240	TI24 (TAU unit2 CH4) alternate pin selection
0	0	P66
0	1	P34
1	0	P134
Other than the above		Setting prohibited (same as “00” setting)

TIS251	TIS250	TI25 (TAU unit2 CH5) alternate pin selection
0	0	P65
0	1	P92
1	0	P35
Other than the above		Setting prohibited (same as “00” setting)

Figure 6-32. Format of TIS20 and TIS21 Registers (2/2)

TIS261	TIS260	TI26 (TAU unit2 CH6) alternate pin selection
0	0	P63
0	1	P36
1	0	P135
Other than the above		Setting prohibited (same as “00” setting)

TIS271	TIS270	TI27 (TAU unit2 CH7) alternate pin selection
0	0	P93
0	1	P62
1	0	P37
Other than the above		Setting prohibited (same as “00” setting)

(16) Timer output select register (TOS00, TOS01, TOS10, TOS11, TOS20, TOS21)

These registers are used for alternate switch of TAU output pins. TOS00 to TOS01 is for TAU unit0, TOS10 to TOS11 for TAU unit1, TOS20 to TOS21 for TAU unit2.

Figure 6-33. Format of Timer Output Select Register 0 (TOS0) (1/2)

Address: F0076H After reset: 00H R/W: (Note: Bits 1 and 5 are read only bit.)

Symbol	7	6	5	4	3	2	1	0
TOS00	TOS031	TOS030	0	TOS020	TOS011	TOS010	0	TOS000

Address: F0077H After reset: 00H R/W: (Note: Bits 1 and 5 are read only bit.)

Symbol	7	6	5	4	3	2	1	0
TOS01	TOS071	TOS070	0	TOS060	TOS051	TOS050	0	TOS040

TOS000	TO00 (TAU unit0 CH0) alternate pin selection
0	P00
1	P136

TOS011	TIS010	TO01 (TAU unit0 CH1) alternate pin selection
0	0	P01
0	1	P80
1	0	P94
Other than the above		Setting prohibited (same as “00” setting)

TOS020	TO02 (TAU unit0 CH2) alternate pin selection
0	P02
1	P50

TOS031	TOS030	TO03 (TAU unit0 CH3) alternate pin selection
0	0	P03
0	1	P81
1	0	P95
1	1	P70

TOS040	TO04 (TAU unit0 CH4) alternate pin selection
0	P04
1	P51

TOS051	TOS050	TO05 (TAU unit0 CH5) alternate pin selection
0	0	P05
0	1	P82
1	0	P96
Other than the above		Setting prohibited (same as “00” setting)

TOS060	TO06 (TAU unit0 CH6) alternate pin selection
0	P06
1	P52

Figure 6-33. Format of Timer Output Select Register 0 (TOS0) (2/2)

TOS071	TOS070	TO07 (TAU unit0 CH7) alternate pin selection
0	0	P07
0	1	P83
1	0	P97
Other than the above		Setting prohibited (same as “00” setting)

Figure 6-34. Format of TOS10 and TOS11 Registers (1/2)

Address: F0079H After reset: 00H R/W: (Note: Bits 0, 1 and 5 are read only bit.)

Symbol	7	6	5	4	3	2	1	0
TOS10	TOS131	TOS130	0	TOS120	TOS111	TOS110	0	0

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOS11	TOS171	TOS170	TOS161	TOS160	TOS151	TOS150	TOS141	TOS140

TOS111	TOS110	TO11 (TAU unit1 CH1) alternate pin selection
0	0	P11
0	1	P84
1	0	P140
1	1	P64

TOS120	TO12 (TAU unit1 CH2) alternate pin selection
0	P12
1	P02

TOS131	TOS130	TO13 (TAU unit1 CH3) alternate pin selection
0	0	P13
0	1	P03
1	0	P53
1	1	P85

TOS141	TOS140	TO14 (TAU unit1 CH4) alternate pin selection
0	0	P14
0	1	P04
1	0	P54
Other than the above		Setting prohibited (same as “00” setting)

TOS151	TOS150	TO15 (TAU unit1 CH5) alternate pin selection
0	0	P15
0	1	P05
1	0	P55
1	1	P86

TOS161	TOS160	TO16 (TAU unit1 CH6) alternate pin selection
0	0	P16
0	1	P06
1	0	P56
Other than the above		Setting prohibited (same as “00” setting)

Figure 6-34. Format of TOS10 and TOS11 Registers (2/2)

TOS171	TOS170	TO17 (TAU unit1 CH7) alternate pin selection
0	0	P17
0	1	P07
1	0	P87
1	1	P57

Figure 6-35. Format of TOS20 and TOS21 Registers (1/2)

Address: F007BH After reset: 00H R/W: (Note: Bits 0, 1 and 5 are read only bit.)

Symbol	7	6	5	4	3	2	1	0
TOS20	TOS231	TOS230	TOS221	TOS220	TOS211	TOS210	TOS201	TOS200

Address: F007CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOS21	TOS271	TOS270	TOS261	TOS260	TOS251	TOS250	TOS241	TOS240

TOS201	TOS200	TO20 (TAU unit2 CH0) alternate pin selection
0	0	P60
0	1	P30
1	0	P132
Other than the above		Setting prohibited (same as “00” setting)

TOS211	TOS210	TO21 (TAU unit2 CH1) alternate pin selection
0	0	P61
0	1	P31
1	0	P131
1	1	P90

TOS221	TOS220	TO22 (TAU unit2 CH2) alternate pin selection
0	0	P75
0	1	P32
1	0	P133
Other than the above		Setting prohibited (same as “00” setting)

TOS231	TOS230	TO23 (TAU unit2 CH3) alternate pin selection
0	0	P74
0	1	P33
1	0	P91
Other than the above		Setting prohibited (same as “00” setting)

TOS241	TOS240	TO24 (TAU unit2 CH4) alternate pin selection
0	0	P66
0	1	P34
1	0	P134
Other than the above		Setting prohibited (same as “00” setting)

TOS251	TOS250	TO25 (TAU unit2 CH5) alternate pin selection
0	0	P65
0	1	P92
1	0	P35
Other than the above		Setting prohibited (same as “00” setting)

Figure 6-35. Format of TOS20 and TOS21 Registers (2/2)

TOS261	TOS260	TO26 (TAU unit2 CH6) alternate pin selection
0	0	P36
0	1	P63
1	0	P135
Other than the above		Setting prohibited (same as “00” setting)

TIS271	TIS270	TI27 (TAU unit2 CH7) alternate pin selection
0	0	P93
0	1	P37
1	0	P62
Other than the above		Setting prohibited (same as “00” setting)

Considering direct LED driving, or other large current application, 16-bit resolution PWM outputs are also alternated to the SM pins. When configure pin function, the policy is that odd TO (ch1,3,5,7) of TAU should be output with higher priority. In addition, 4 kinds of output with different periods using different master CH’s can be achieved if by this means.

(17) Timer input select else register (TISELSE)

This register provides below selection function.

(a) TAU unit 0 channel5 input selection

The input source can be timer input signal (TI05) from port or internal/external clock.

(b) Timer conjunction function of timer output to timer input just like 78K0/Dx2.

TAU unit0 CH1 output can be connected to TAU unit0 CH0. This function is controlled by bit6.

TAU unit2 CH1 output can be connected to TAU unit2 CH0. This function is controlled by bit7.

This function is used for measuring speed or taco pulse. If only use timer caupture function to measure, there will be too many interrupts and increase the loading of software when input is higher (about 8kHz, 125us interrupt interval). So division of interrupt is necessary. At this usage, one timer is used as capture mode, its output is internally connected to another timer (operated as external event mode) to generate divided interrupt.

(Refer to TMP2 and TMP3 conjunction function of 78K0/Dx2)

Figure 6-36. Format of TISELSE Registers

Address: FFF3E After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TISELSE	TOTICON1	TOTICON0	0	0	0	0	TI05SEL1	TI05SEL0

TI05SEL1	TI05SEL0	TIS051	TIS050	TAU unit0 CH5 input alternate selection
0	0	0	0	P05
0	0	0	1	P82
0	0	1	0	P96
0	1	x	x	Low-speed on-chip clock (f_{IL})
1	0	x	x	Sub system clock (f_{SUB})
1	1	x	x	Main external clock (f_{EX})
Other than the above				Setting prohibited (same as “0000” setting)

Considering the below purposes, every peripheral clock is connected to TI05 of TAU0

- Low-speed on-chip clock: For Frequency Detection of Safefy Function.
- Sub system clock: For the ultra accuracy trimming of high-speed on-chip oscillator Note
- External main clock: For the ultra accuracy trimming of high-speed on-chip oscillator without sub system clock Note

Note: Count present operation frequency by timer. It is possible to change trimming code by access HIOTRM register.

TOTICON0	Timer conjunction function control
0	Cut off the connection of TAU unit0 CH1 output to CH0 input
1	Connect TAU unit0 CH1 output to TAU unit0 CH0 input

TOTICON1	Timer conjunction function control
0	Cut off the connection of TAU unit2 CH1 output to CH0 input
1	Connect TAU unit2 CH1 output to TAU unit2 CH0 input

The connection with TOTICONn is used to count external event (pulse) input to TI01/TI21 in long term such as 16-bit counter is overflowed. Timer array unit 0 channel 1/timer array unit 2 channel 1 is worked as divider of input pulse and generates slower pulse to TO01/TO21. Timer array unit 0 channel 0/timer array unit 2 channel 0 is worked as external event counter. Its expected value should be made the calculated value according to timer array unit 0 channel 1/timer array unit 2 channel 1 divider setting.

(18) Serial communication pin select register (STSEL0, STSEL1)

These registers are used for alternate switch of serial input/output pins.

Figure 6-37. Format of STSEL0 Register

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL0	0	SCSI100	0	SCSI010	SCSI001	SCSI000	SUARTF1	SUARTF0

SUARTF0	Communication pin selection of UARTF0	
	LTxD0	LRxD0
0	P71	P70
1	P15	P14

SUARTF1	Communication pin selection of UARTF1	
	LTxD1	LRxD1
0	P10	P11
1	P131	P132

SCSI001	SCSI000	CSI00 communication pin selection		
		$\overline{SCK00}$	SI00	SO00
0	0	P10	P11	P12
0	1	P04	P03	P02
1	0	P34	P33	P32
Other than the above		Setting prohibited (same as “00” setting)		

SCSI010	CSI01 communication pin selection		
	$\overline{SCK01}$	SI01	SO01
0	P74	P75	P13
1	P56	P55	P54

SCSI100	CSI10 communication pin selection		
	$\overline{SCK10}$	SI10	SO10
0	P133	P132	P131
1	P51	P52	P53

Figure 6-38. Format of STSEL1 Register

Address: FFF3DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL1	SIIC1	SIIC0	0	0	SCAN1	SCAN0	TMCAN1	TMCAN0

TMCAN0	Input source switch of TAU unit1 CH4
0	Input from TI14 (after selected by TIS141 to 0 bits)
1	TSOUT of aFCAN0 (CAN0 time stamp function)

TMCAN1	Input source switch of TAU unit1 CH5
0	Input from TI15 (after selected by TIS151 to 0 bits)
1	TSOUT of aFCAN1 (CAN1 time stamp function)

SCAN0	Communication pin selection of aFCAN0	
	CTxD0	CRxD0
0	P71	P70
1	P00	P01

SCAN1	Communication pin selection of aFCAN1	
	CTxD1	CRxD1
0	P62	P63
1	P134	P135

SIIC1	SIIC0	Communication pin selection of IIC11	
		SCL11	SDA11
0	0	P60	P61
0	1	P30	P31
1	0	P136	P50
Other than the above		Setting prohibited	

(19) Port mode registers 0, 1, 3, 5 to 9, 13, 14 (PM0, PM1, PM3, PM5 to PM9, PM13, PM14)

These registers set input/output of ports 0, 1, 3, 5 to 9, 13, 14 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, PM5 to PM9, PM13, PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (1/13)

(a) Alternate function of P0 (1/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P00	TI00	Input	1	x	0	TIS00.0 = 0 TISELSE.6 = 0	-
	TO00	Output	0	0	0	TOS00.0 = 0	STSEL1.2 = 0
	CTxD0	Output	0	1	0	STSEL1.2 = 1	-
	SEG14	Output	x	x	1	-	-
P01	TI01	Input	1	x	0	TIS00.3,2 = 00	-
	TO01	Output	0	0	0	TOS00.3,2 = 00	-
	CRxD0	Input	1	x	0	STSEL1.2 = 1	-
	SEG15	Output	x	x	1	-	-
P02	TI02	Input	1	x	0	TIS00.4 = 0	-
	TO02	Output	0	0	0	TOS00.4 = 0	STSEL0.3,2 = 00/10 TOS10.4 = 0
	TI12	Input	1	x	0	TIS10.4 = 1	-
	TO12	Output	0	0	0	TOS10.4 = 1	STSEL0.3,2 = 00/10 TOS00.4 = 0
	SO00	Output	0	1	0	STSEL0.3,2 = 01	-
	SEG16	Output	x	x	1	-	-
P03	TI03	Input	1	x	0	TIS00.7,6 = 00	-
	TO03	Output	0	0	0	TOS00.7,6 = 00	TOS10.7,6 = 00/10/11
	TI13	Input	1	x	0	TIS10.7,6 = 01	-
	TO13	Output	0	0	0	TOS10.7,6 = 01	TOS00.7,6 = 01/10/11
	SI00	Input	1	x	0	STSEL0.3,2 = 01	-
	SEG17	Output	x	x	1	-	-
P04	TI04	Input	1	x	0	TIS01.0 = 0	
	TO04	Output	0	0	0	TOS01.0 = 0	STSEL0.3,2 = 00/10 TOS11.1,0 = 00/10
	TI14	Input	1	x	0	TIS11.1,0 = 01 STSEL1.0 = 0	-
	TO14	Output	0	0	0	TOS11.1,0 = 01	STSEL0.3,2 = 00/10 TOS01.0 = 1
	$\overline{SCK00}$	Output	0	1	0	STSEL0.3,2 = 01	-
		Input	1	x			
	SEG18	Output	x	x	1	-	-
P05	TI05	Input	1	x	0	TIS01.3,2 = 00 TISELSE.1,0 = 00	-
	TO05	Output	0	0	0	TOS01.3,2 = 00	TOS11.3,2 = 00/10/11
	TI15	Input	1	x	0	TIS11.3,2 = 01 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 01	TOS01.3,2 = 01/10
	SEG19	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (2/13)

(a) Alternate function of P0 (2/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P06	TI06	Input	1	x	0	TIS01.4 = 0 RTCSEL.0 = 0	
	TO06	Output	0	0	0	TOS01.4 = 0	TOS11.5,4 = 00/10
	TI16	Input	1	x	0	TIS11.5,4 = 01 RTCSEL.2 = 0	
	TO16	Output	0	0	0	TOS11.5,4 = 01	TOS01.4 = 1
	SEG20	Output	x	x	1	-	-
P07	TI07	Input	1	x	0	TIS01.7,6 = 00 RTCSEL.1 = 0	-
	TO07	Output	0	0	0	TOS01.7,6 = 00	TOS11.7,6 = 00/10/11
	TI17	Input	1	x	0	TIS11.7,6 = 01 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 01	TOS01.7,6 = 01/10
	SEG21	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (3/13)

(b) Alternate function of P1 (1/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P10	INTP4	Input	1	x	0	-	-
	TI10	Input	1	x	0	-	-
	TO10	Output	0	0	0	-	STSEL0.1 = 1 STSEL0.3,2 = 01/10
	LTxD1	Output	0	1	0	STSEL0.1 = 0	STSEL0.3,2 = 01/10
	$\overline{SCK00}$	Output	0	1	0	STSEL0.3,2 = 00	STSEL0.1 = 1
		Input	1	x			-
	SEG31	Output	x	x	1	-	-
P11	INTPLR1	Input	1	x	0	STSEL0.1 = 0	-
	TI11	Input	1	x	0	TIS10.3,2 = 00	-
	TO11	Output	0	0	0	TOS10.3,2 = 00	-
	LRxD1	Input	1	x	0	STSEL0.1 = 0	-
	SI00	Input	1	x	0	STSEL0.3,2 = 00	-
	SEG30	Output	x	x	1	-	-
P12	INTP2	Input	1	x	0	-	-
	TI12	Input	1	x	0	TIS10.4 = 0	-
	TO12	Output	0	0	0	TOS10.4 = 0	STSEL0.3,2 = 01/10
	SO00	Output	0	1	0	STSEL0.3,2 = 00	TOS10.4 = 1
	SEG29	Output	x	x	1	-	-
P13	TI13	Input	1	x	0	TIS10.7,6 = 00	-
	TO13	Output	0	0	0	TOS10.7,6 = 00	STSEL0.4 = 1
	SO01	Output	0	1	0	STSEL0.4 = 0	TOS10.7,6 = 01/10/11
	SEG25	Output	x	x	1	-	-
P14	INTPLR0	Input	1	x	0	STSEL0.0 = 1	-
	TI14	Input	1	x	0	TIS11.1,0 = 00 STSEL1.0 = 0	-
	TO14	Output	0	0	0	TOS11.1,0 = 00	-
	LRxD0	Input	1	x	0	STSEL0.0 = 1	-
	SEG24	Output	x	x	1	-	-
P15	TI15	Input	1	x	0	TIS11.3,2 = 00 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 00	RTCSEL.7,6 = 00/10/11 STSEL0.0 = 0
	RTC1HZ	Output	0	0	0	RTCSEL.7,6 = 01	TOS11.3,2 = 01/10/11 STSEL0.0 = 0
	LTxD0	Output	0	1	0	STSEL0.0 = 1	TOS11.3,2 = 01/10/11 RTCSEL.7,6 = 00/10/11
	SEG23	Output	x	x	1	-	-
P16	TI16	Input	1	x	0	TIS11.5,4 = 00 RTCSEL.2 = 0	-
	TO16	Output	0	0	0	TOS11.5,4 = 00	-
	SEG22	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (4/13)

(b) Alternate function of P1 (2/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P17	INTP0	Input	1	x	0	-	-
	TI17	Input	1	x	0	TIS11.7,6 = 00 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 00	STSEL0.3,2 = 01/10
	SEG28	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (5/13)

(c) Alternate function of P3

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P30	TI20	Input	1	x	0	TIS20.1,0 = 01 TISELSE.7 = 0	-
	TO20	Output	0	0	0	TOS20.1,0 = 01	STSEL1.7,6 = 00/10
	SCL11	Output	0	1	0	STSEL1.7,6 = 01	TOS20.1,0 = 00/10
	SEG6	Output	x	x	1	-	-
P31	TI21	Input	1	x	0	TIS20.3,2 = 01	-
	TO21	Output	0	0	0	TOS20.3,2 = 01	STSEL1.7,6 = 00/10
	SDA11	I/O	0	1	0	STSEL1.7,6 = 01	TOS20.3,2 = 00/10/11
	SEG7	Output	x	x	1	-	-
P32	TI22	Input	1	-	0	TIS20.5,4 = 01	-
	TO22	Output	0	0	0	TOS20.5,4 = 01	STSEL0.3,2 = 00/01
	SO00	Output	0	1	0	STSEL0.3,2 = 10	TOS20.5,4 = 00/10
	SEG8	Output	x	x	1	-	-
P33	TI23	Input	1	x	0	TIS20.7,6 = 01	-
	TO23	Output	0	0	0	TOS20.7,6 = 01	STSEL0.3,2 = 00/01
	SI00	Input	1	x	0	STSEL0.3,2 = 10	TOS20.7,6 = 00/10
	SEG9	Output	x	x	1	-	-
P34	TI24	Input	1	x	0	TIS21.1,0 = 01	-
	TO24	Output	0	0	0	TOS21.1,0 = 01	STSEL0.3,2 = 00/01
	$\overline{\text{SCK00}}$	Output	0	1	0	STSEL0.3,2 = 10	TOS21.1,0 = 00/10
		Input	1	x			-
	SEG10	Output	x	x	1	-	-
P35	TI25	Input	1	x	0	TIS21.3,2 = 10	-
	TO25	Output	0	0	0	TOS21.3,2 = 10	STSEL0.3,2 = 00/01
	SEG11	Output	x	x	1	-	-
P36	TI26	Input	1	x	0	TIS21.5,4 = 01	-
	TO26	Output	0	0	0	TOS21.5,4 = 00	-
	SEG12	Output	x	x	1	-	-
P37	TI27	Input	1	x	0	TIS21.7,6 = 10	-
	TO27	Output	0	0	0	TOS21.7,6 = 01	-
	SEG13	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (6/13)

(d) Alternate function of P5

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P50	TI02	Input	1	x	0	TIS00.4 = 1	-
	TO02	Output	0	0	0	TOS00.4 = 1	STSEL1.7,6 = 00/01
	SDA11	I/O	0	1	0	STSEL1.7,6 = 10	TOS00.4 = 0
	SEG49	Output	x	x	1	-	-
P51	TI04	Input	1	x	0	TIS01.0 = 1	-
	TO04	Output	0	0	0	TOS01.0 = 1	STSEL0.6 = 0
	$\overline{SCK10}$	Output	0	1	0	STSEL0.6 = 1	TOS01.0 = 0
		Input	1	x			-
	SEG50	Output	x	x	1	-	-
P52	TI06	Input	1	x	0	TIS01.4 = 1 RTCSEL.0 = 0	-
	TO06	Output	0	0	0	TOS01.4 = 1	-
	SI10	Input	1	x	0	STSEL0.6 = 1	-
	SEG51	Output	x	x	1	-	-
P53	TI13	Input	1	x	0	TIS10.7,6 = 10	-
	TO13	Output	0	0	0	TOS10.7,6 = 10	STSEL0.6 = 0
	SO10	Output	0	1	0	STSEL0.6 = 1	TOS10.7,6 = 00/01/11
	SEG52	Output	x	x	1	-	-
P54	TI14	Input	1	x	0	TIS11.1,0 = 10 STSEL1.0 = 0	-
	TO14	Output	0	0	0	TOS11.1,0 = 10	
	SO01	Output	0	1	0	STSEL0.4 = 1	
	SEG2	Output	x	x	1	-	-
P55	TI15	Input	1	x	0	TIS11.3,2 = 10 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 10	
	SI01	Input	1	x	0	STSEL0.4 = 1	-
	SEG3	Output	x	x	1	-	-
P56	TI16	Input	1	x	0	TIS11.5,4 = 10 RTCSEL.2 = 0	-
	TO16	Output	0	0	0	TOS11.5,4 = 10	STSEL0.4 = 0
	$\overline{SCK01}$	Output	0	1	0	STSEL0.4 = 1	TOS11.5,4 = 00/01
		Input	1	x			-
	SEG4	Output	x	x	1	-	-
P57	TI17	Input	1	x	0	TIS11.7,6 = 11 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 11	-
	SEG5	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (7/13)

(e) Alternate function of P6

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P60	INTP1	Input	1	x	N/A	-	-
	TI20	Input	1	x		TIS20.1,0 = 00 TISELSE.7 = 0	-
	TO20	Output	0	0		TOS20.1,0 = 00	STSEL1.7,6 = 01/10
	SCL11	Output	0	1		STSEL1.7,6 = 00	TOS20.1,0 = 01/10
P61	INTP3	Input	1	x	N/A	-	-
	TI21	Input	1	x		TIS20.3,2 = 00	-
	TO21	Output	0	0		TOS20.3,2 = 00	STSEL1.7,6 = 01/10
	SDA11	I/O	0	1		STSEL1.7,6 = 00	TOS20.3,2 = 01/10/11
P62	TI27	Input	1	x	N/A	TIS21.7,6 = 01	-
	TO27	Output	1	0		TOS21.7,6 = 10	STSEL1.3 = 1
	CTxD1	Output	0	1		STSEL1.3 = 0	TOS21.7,6 = 00/01
P63	TI26	Input	1	x	N/A	TIS21.5,4 = 00	-
	TO26	Output	0	0		TOS21.5,4 = 01	-
	CRxD1	Input	1	x		STSEL1.3 = 0	-
P64	TI11	Input	1	x	N/A	TIS10.3,2 = 11	-
	TO11	Output	0	0		TOS10.3,2 = 11	RTCSEL.7,6 = 01/10/11
	RTC1HZ	Output	0	0		RTCSEL.7,6 = 00	TOS10.3,2 = 00/01/10
P65	TI25	Input	1	x	N/A	TIS21.3,2 = 00	-
	TO25	Output	0	0		TOS21.3,2 = 00	-
P66	TI24	Input	1	x	N/A	TIS21.1,0 = 00	-
	TO24	Output	0	0		TOS21.1,0 = 00	SGSEL.3 = 0
	PCL	Output	0	0		SGSEL.3 = 1	TOS21.1,0 = 01/10

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (8/13)

(f) Alternate function of P7

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P70	INTPLR0	Input	1	x	N/A	STSEL0.0 = 0	-
	TI03	Input	1	x		TIS00.7,6 = 11	-
	TO03	Output	0	0		TOS00.7,6 = 11	-
	CRxD0	Input	1	x		STSEL1.2 = 0	-
	LRxD0	Input	1	x		STSEL0.0 = 0	-
P71	CTxD0	Output	0	1	N/A	STSEL1.2 = 0	STSEL0.0 = 1
	LTxD0	Output	0	1		STSEL0.0 = 0	STSEL1.3 = 1
P72	ADTRG	Input	1	x	0	-	-
	SGOA	Output	0	0	0	SGSEL.2-0 = 000	-
	SEG1	Output	x	x	1	-	-
P73	SGO/SGOF	Output	0	0	0	SGSEL.2-0 = 000	-
	SEG0	Output	x	x	1	-	-
P74	TI23	Input	1	x	0	TIS20.7,6 = 00	-
	TO23	Output	0	0	0	TOS20.7,6 = 00	STSEL0.4 = 1
	$\overline{\text{SCK01}}$	Output	0	1	0	STSEL0.4 = 0	TOS20.7,6 = 01/10
		Input	1	x			-
	SEG26	Output	x	x	1	-	-
P75	TI22	Input	1	x	0	TIS20.5,4 = 00	-
	TO22	Output	0	0	0	TOS20.5,4 = 00	SGSEL.3 = 1
	PCL	Output	0	0	0	SGSEL.3 = 0	TOS20.5,4 = 01/10
	SI01	Input	1	x	0	STSEL0.4 = 0	-
	SEG27	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (9/13)

(g) Alternate function of P8

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P80	TI01	Input	1	x	0	TIS00.3,2 = 01	-
	TO01	Output	0	0	0	TOS00.3,2 = 01	SMPC.0 = 0
	SM11	Output	0	0	0	SMPC.0 = 1	TOS00.3,2 = 00/10
	SEG32	Output	x	x	1	-	-
P81	TI03	Input	1	x	0	TIS00.7,6 = 01	-
	TO03	Output	0	0	0	TOS00.7,6 = 01	SMPC.0 = 0
	SM12	Output	0	0	0	SMPC.0 = 1	TOS00.7,6 = 00/10/11
	SEG33	Output	x	x	1	-	-
P82	TI05	Input	1	x	0	TIS01.3,2 = 01 TISELSE.1,0 = 00	-
	TO05	Output	0	0	0	TOS01.3,2 = 01	SMPC.0 = 0
	SM13	Output	0	0	0	SMPC.0 = 1	TOS01.3,2 = 00/10
	SEG34	Output	x	x	1	-	-
P83	TI07	Input	1	x	0	TIS01.7,6 = 01 RTCSEL.1 = 0	-
	TO07	Output	0	0	0	TOS01.7,6 = 01	SMPC.0 = 0
	SM14	Output	0	0	0	SMPC.0 = 1	TOS01.7,6 = 00/10
	ZPD14	Input	x	x	0	ZPDS0.3 = 1	-
	SEG35	Output	x	x	1	-	-
P84	TI11	Input	1	x	0	TIS10.3,2 = 01	-
	TO11	Output	0	0	0	TOS10.3,2 = 01	SMPC.1 = 0
	SM21	Output	0	0	0	SMPC.1 = 1	TOS10.3,2 = 00/10/11
	SEG36	Output	x	x	1	-	-
P85	TI13	Input	1	x	0	TIS10.7,6 = 11	-
	TO13	Output	0	0	0	TOS10.7,6 = 11	SMPC.1 = 0
	SM22	Output	0	0	0	SMPC.1 = 1	TOS10.7,6 = 00/01/10
	SEG37	Output	x	x	1	-	-
P86	TI15	Input	1	x	0	TIS11.3,2 = 11 STSEL1.1 = 0	-
	TO15	Output	0	0	0	TOS11.3,2 = 11	SMPC.1 = 0
	SM23	Output	0	0	0	SMPC.1 = 1	TOS11.3,2 = 00/01/10
	SEG38	Output	x	x	1	-	-
P87	TI17	Input	1	x	0	TIS11.7,6 = 10 RTCSEL.3 = 0	-
	TO17	Output	0	0	0	TOS11.7,6 = 10	SMPC.1 = 0
	SM24	Output	0	0	0	SMPC.1 = 1	TOS11.7,6 = 00/01/11
	ZPD24	Input	x	x	0	ZPDS0.7 = 1	-
	SEG39	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (10/13)

(h) Alternate function of P9(1/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P90	TI21	Input	1	x	0	TIS20.3,2 = 11	-
	TO21	Output	0	0	0	TOS20.3,2 = 11	SMPC.2 = 0
	SM31	Output	0	0	0	SMPC.2 = 1	TOS20.3,2 = 00/01/10
	SEG40	Output	x	x	1	-	-
P91	TI23	Input	1	x	0	TIS20.7,6 = 10	-
	TO23	Output	0	0	0	TOS20.7,6 = 10	SMPC.2 = 0
	SM32	Output	0	0	0	SMPC.2 = 1	TOS20.7,6 = 00/01
	SEG41	Output	x	x	1	-	-
P92	TI25	Input	1	x	0	TIS21.3,2 = 01	-
	TO25	Output	0	0	0	TOS21.3,2 = 01	SMPC.2 = 0 SGSEL.2-0 = 000/010/100 to 110
	SM33	Output	0	0	0	SMPC.2 = 1	TOS21.3,2 = 00/10 SGSEL.2-0 = 000/010/100 to 110
	SGOA	Output	0	0	0	SGSEL.2-0 = 001	SMPC.2 = 0 TOS21.3,2 = 00/10
	SEG42	Output	x	x	1	-	-
P93	TI27	Input	1	x	0	TIS21.7,6 = 00	-
	TO27	Output	0	0	0	TOS21.7,6 = 00	SMPC.2 = 0 SGSEL.1,0 = 00/10
	SM34	Output	0	0	0	SMPC.2 = 1	TOS21.7,6 = 01/10 SGSEL.1,0 = 00/10
	ZPD34	Input	x	x	0	ZPDS1.3 = 1	-
	SGO/SGOF	Output	0	0	0	SGSEL.1,0 = 01	TOS21.7,6 = 01/10 SMPC.2 = 0
	SEG43	Output	x	x	1	-	-
P94	TI01	Input	1	x	0	TIS00.3,2 = 10	-
	TO01	Output	0	0	0	TOS00.3,2 = 10	SMPC.3 = 0 RTCSEL.7,6 = 00/01
	RTC1HZ	Output	0	0	0	RTCSEL.7,6 = 10	TOS00.3,2 = 00/01 SMPC.3 = 0
	SM41	Output	0	0	0	SMPC.3 = 1	TOS00.3,2 = 00/01 RTCSEL.7,6 = 00/01
	SEG44	Output	x	x	1	-	-
P95	TI03	Input	1	x	0	TIS00.7,6 = 10	-
	TO03	Output	0	0	0	TOS00.7,6 = 10	SMPC.3 = 0
	SM42	Output	0	0	0	SMPC.3 = 1	TOS00.7,6 = 00/01/11
	SEG45	Output	x	x	1	-	-
P96	TI05	Input	1	x	0	TIS01.3,2 = 10 TISELSE.1,0 = 00	-
	TO05	Output	0	0	0	TOS01.3,2 = 10	SMPC.3 = 0
	SM43	Output	0	0	0	SMPC.3 = 1	TOS01.3,2 = 00/01
	SEG46	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (11/13)

(h) Alternate function of P9 (2/2)

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P97	TI07	Input	1	x	0	TIS01.7,6 = 10 RTCSEL.1 = 0	-
	TO07	Output	0	0	0	TOS01.7,6 = 10	SMPC.3 = 0
	SM44	Output	0	0	0	SMPC.3 = 1	TOS01.7,6 = 00/01
	ZPD44	Input	x	x	0	ZPDS1.7 = 1	-
	SEG47	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (12/13)

(i) Alternate function of P13

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P131	TI21	Input	1	x	N/A	TIS20.3,2 = 10	-
	TO21	Output	0	0		TOS20.3,2 = 10	STSEL0.6 = 1 STSEL0.1 = 0
	SO10	Output	0	1		STSEL0.6 = 0	TOS20.3,2 = 00/01/11 STSEL0.1 = 0
	LTxD1	Output	0	1		STSEL0.1 = 1	TOS20.3,2 = 00/01/11 STSEL0.6 = 1
P132	INTPLR1	Input	1	x	N/A	STSEL0.1 = 1	-
	TI20	Input	1	x		TIS20.1,0 = 10 TISELSE.7 = 0	-
	TO20	Output	0	0		TOS20.1,0 = 10	-
	SI10	Input	1	x		STSEL0.6 = 0	-
	LRxD1	Input	1	x		STSEL0.1 = 1	-
P133	TI22	Input	1	x	N/A	TIS20.5,4 = 10	-
	TO22	Output	0	0		TOS20.5,4 = 10	STSEL0.6 = 1
	$\overline{\text{SCK10}}$	Output	0	1		STSEL0.6 = 0	TOS20.5,4 = 00/01
		Input	1	x			-
P134	TI24	Input	1	x	N/A	TIS21.1,0 = 10	-
	TO24	Output	0	0		TOS21.1,0 = 10	SGSEL.2-0 = 000/001/100 to 110 STSEL1.3 = 0
	SGOA	Output	0	0		SGSEL.2-0 = 010	TOS21.1,0 = 00/01 STSEL1.3 = 0
	CTxD1	Output	0	1		STSEL1.3 = 1	TOS21.1,0 = 00/01 SGSEL.2-0 = 000/001/100 to 110
P135	TI26	Input	1	x	N/A	TIS21.5,4 = 10	-
	TO26	Output	0	0		TOS21.5,4 = 10	SGSEL.1, 0 = 00/01
	SGO/SGOF	Output	0	0		SGSEL.1,0 = 10	TOS21.5,4 = 00/01
	CRxD1	Input	1	x		STSEL1.3 = 1	-
P136	TI00	Input	1	x	0	TIS00.0 = 1 TISELSE.6 = 0	-
	TO00	Output	0	0	0	TOS00.0 = 1	STSEL1.7,6 = 00/01
	SCL11	Output	0	1	0	STSEL1.7,6 = 10	TOS00.0 = 0
	SEG48	Output	x	x	1	-	-

Table 6-5. Settings of Register, and Output Latch When Using Alternate Function (13/13)

(j) Alternate function of P14

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P14	TI11	Input	1	x	N/A	TIS10.3,2 = 10	-
	TO11	Output	0	0		TOS10.3,2 = 10	-

Figure 6-39. Format of Port Mode Registers (1/7)
(48-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	1	1	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	1	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	1	1	1	1

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	1	1

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	1	1	1	PM94	PM93	PM92	PM91	PM90

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5 to 9; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 2 to 7 of the PM0 register, bits 5 to 7 of the PM1 register, bit 2 and bits 4 to 7 of the PM3 register, bits 0 to 3 of the PM5 register, bits 2 to 7 of the PM6 register, bits 0 to 1 and 6 to 7 of the PM7 register, bits 4 to 7 of the PM8 register, bits 5 to 7 of the PM9 register and bits 0 to 7 of the PM13 register to "1".

Figure 6-39. Format of Port Mode Registers (2/7)
(64-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	1	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	1	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	1	1	1	1

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

Address: FFF27H After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF28H After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

Address: FFF29H After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	1	1	1	PM94	PM93	PM92	PM91	PM90

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5 to 9; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Cautions 1. Be sure to set bit 6 of the PM0 and PM1 registers, bits 4 to 6 of the PM2 register, bits 4 to 7 of the PM3 register, bits 1 to 7 of the PM4 register, bits 0 to 3 of the PM5 register, bits 2 to 7 of the PM6 register, bits 6 and 7 of the PM7 register, and bits 5 to 7 of the PM9 register to "1".

2. If port is set to analog input by ADPC register, PM setting is invalid.

Figure 6-39. Format of Port Mode Registers (3/7)
(80-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	1	1	1	1

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	PM66	PM65	1	1	1	PM61	PM60

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5 to 9; n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Cautions 1. Be sure to set bits 1 to 7 of the PM4 register, bits 0 to 3 of the PM5, bits 2 to 4 and 7 of the PM6 register, and bits 6 and 7 of the PM7 register to "1".

2. If port is set to analog input by ADPC register, PM setting is invalid.

Figure 6-39. Format of Port Mode Registers (4/7)
(100-pin products) (1/2)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

Address: FFF2DH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	PM136	PM135	PM134	PM133	PM132	PM131	0

Figure 6-39. Format of Port Mode Registers (5/7)
(100-pin products) (2/2)

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM140

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5 to 9, 13, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bit 7 of the PM6 register, bits 6 and 7 of the PM7 register, bit 7 of the PM13 register, and bits 1 to 7 of the PM14 register to “1” and bit 0 of the PM13 register to “0”.

<R>

Figure 6-39. Format of Port Mode Registers (6/7)
(128-pin products) (1/2)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

Address: FFF2AH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100

Figure 6-39. Format of Port Mode Registers (7/7)
(128-pin products) (2/2)

Address: FFF2BH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	PM127	PM126	PM125	1	1	1	1	1

Address: FFF2DH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	PM136	PM135	PM134	PM133	PM132	PM131	0

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM140

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	1	1	1	1	1	PM152	PM151	PM150

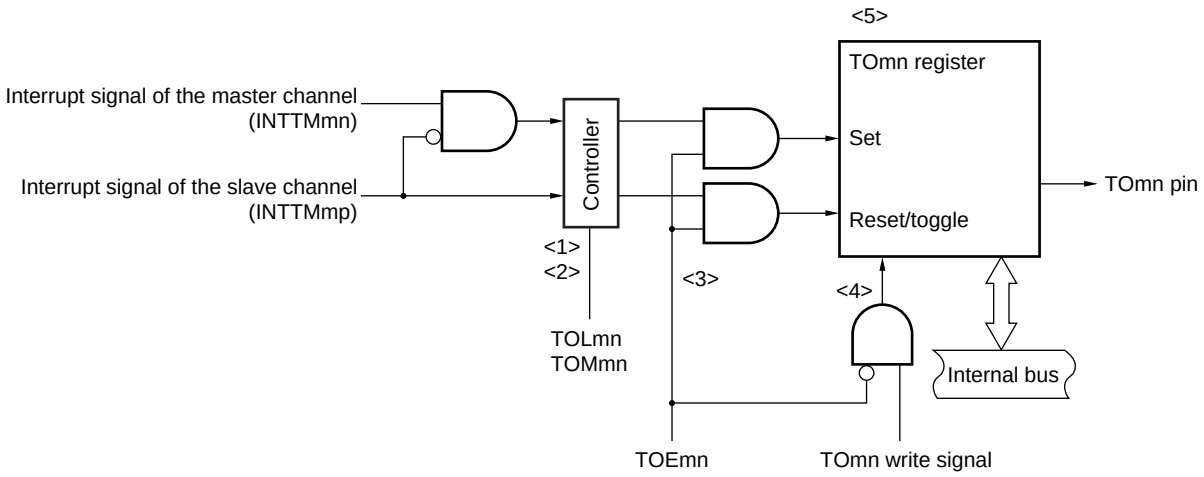
PMmn	Pmn pin I/O mode selection (m = 0 to 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 1 to 7 of the PM4 register, bits 7 of the PM6 register, bits 6 and 7 of the PM7 register, bits 0 to 4 of the PM12 register, bits 7 of the PM13 register, bits 1 to 7 of the PM14, and bits 3 to 7 of the PM15 registers to "1" and bit 0 of the PM13 register to "0".

6.4 Channel Output (TOmn Pin) Control

6.4.1 TOmn pin output circuit configuration

Figure 6-40. 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 (m = 0 to 2), n: Channel number, p: Slave channel number

<3> When TOEmn = 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 TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set TOEmn = 0 and to write a value to TOmn.

<4> When TOEmn = 0, writing to TOmn bit to the target channel (TOmn write signal) becomes valid. When TOEmn = 0 neither INTTMmn (master channel timer interrupt) nor INTTMmp (slave channel timer interrupt) is transmitted to TOmn register.

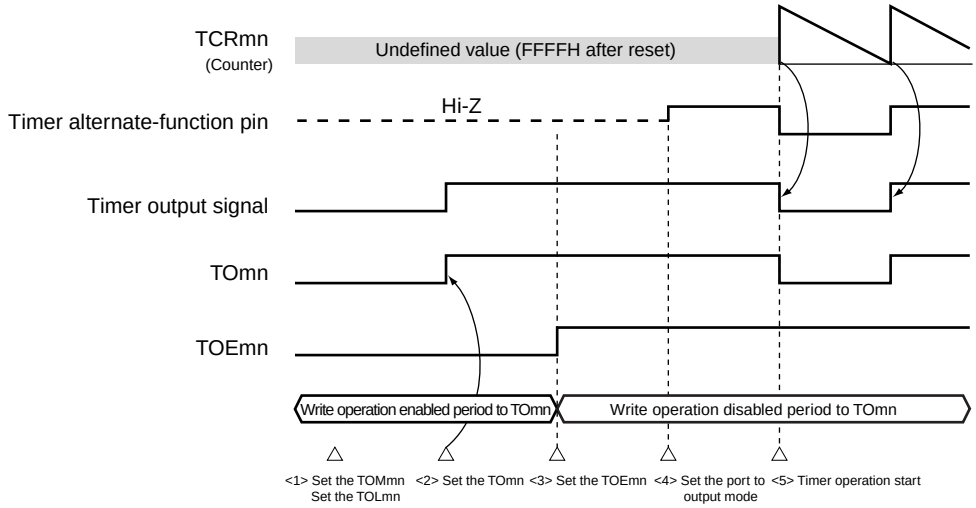
<5> The TOmn register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0 to 2), n: Channel number, p: Slave channel number

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 6-41. Status Transition from Timer Output Setting to 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 TOEmn (writing to TOmn is disabled).
<4> The port I/O setting is set to output (see **6.3 (19) Port mode registers 0 to 9, 13 to 15**).
<5> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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 to 2)
n: Channel number (n = 0 to 7)

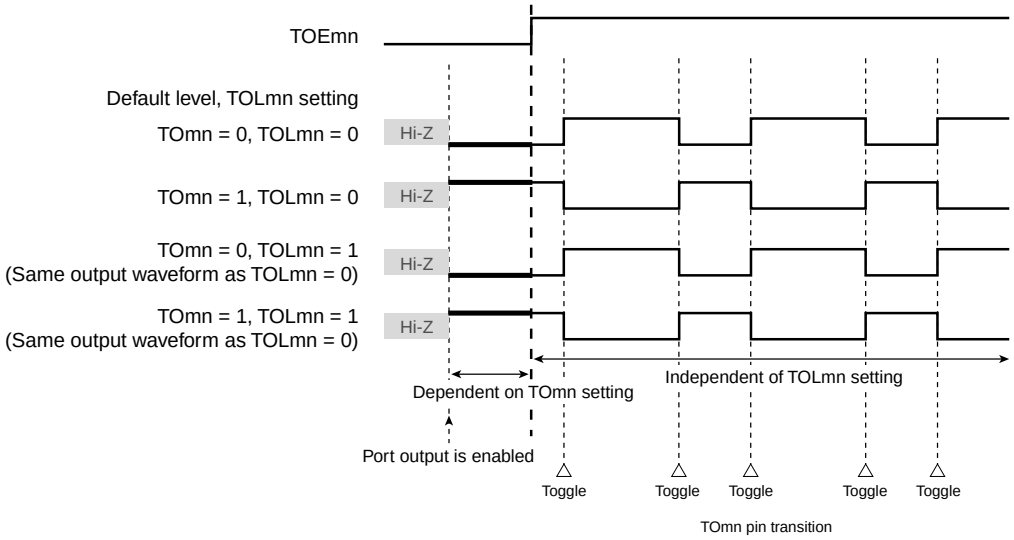
(2) Default level of TOmn pin and output level after timer operation start

The following figure shows the TOmn pin output level transition when writing has been done in the state of TOEmn = 0 before port output is enabled and TOEmn = 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 6-42. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



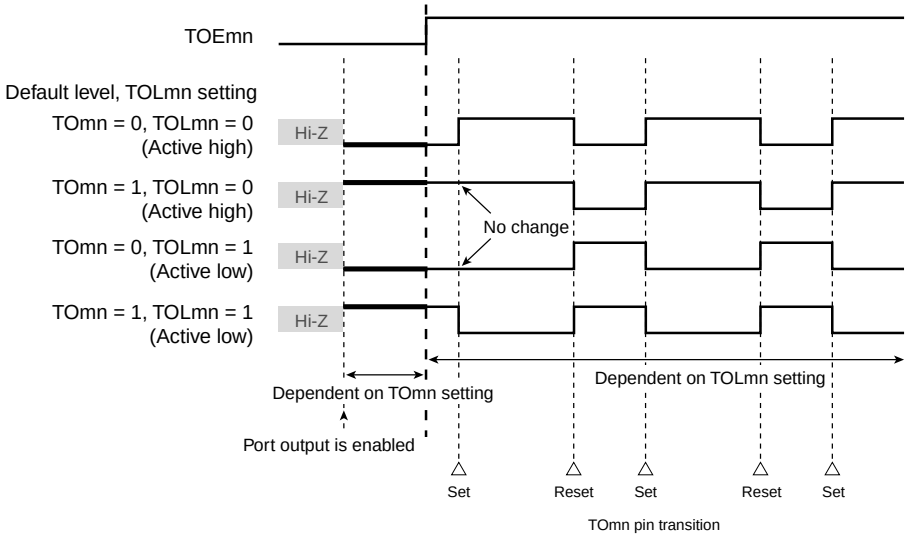
Remarks 1. Toggle: Reverse TOmn pin output status

2. m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(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 6-43. 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 to 2)

n: Channel number (n = 0 to 7)

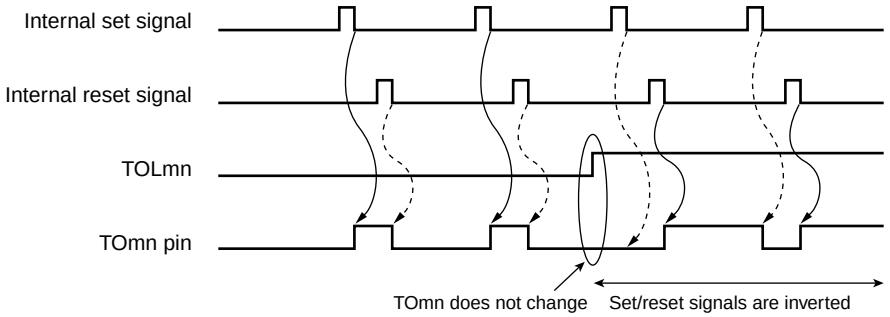
(3) Operation of TOmn pin in combination operation mode (TOMmn = 1)

(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 6-44. 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 to 2)
n: Channel number (n = 0 to 7)

(b) Set/reset timing

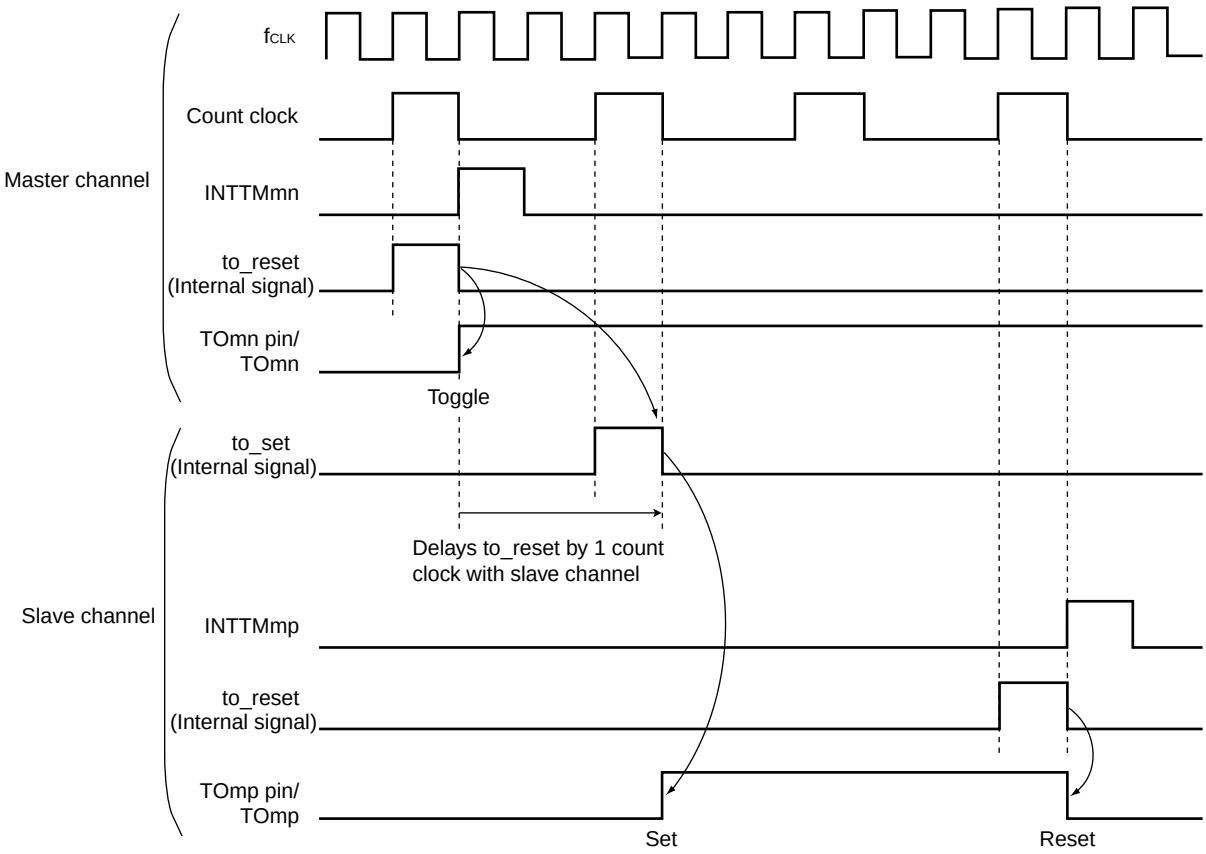
To realize 0%/100% output at PWM output, the TOmn pin/TOmn set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-40 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6-45. Set/Reset Timing Operating Statuses



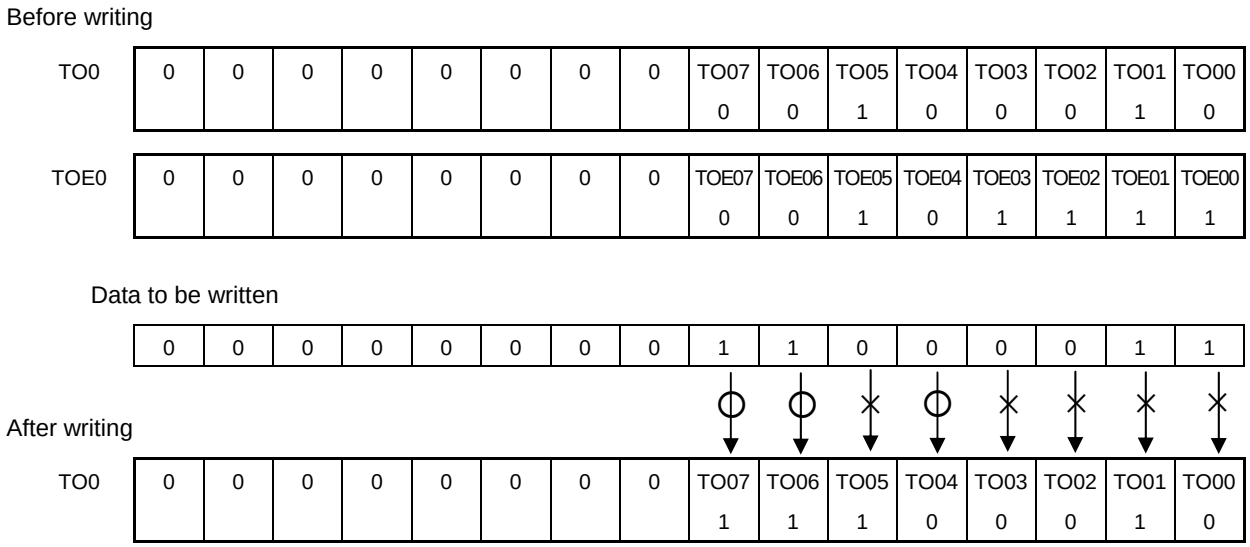
Remarks 1. to_reset: TOmn pin reset/toggle signal
to_set: TOmn pin set signal

2. m: Unit number (m = 0 to 2), n: Channel number, p: Slave channel number

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 TOEmn = 0 to a target TOmn (channel output).

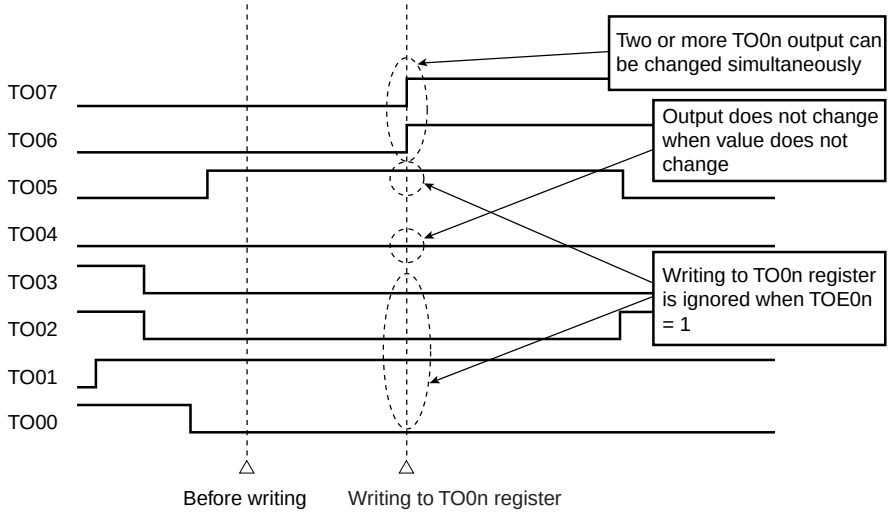
Figure 6-46. Example of TO0n Bits Collective Manipulation



Writing is done only to TOmn bits with TOEmn = 0, and writing to TOmn bits with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to TOmn, it is ignored and the output change by timer operation is normally done.

Figure 6-47. TOmn Pin Statuses by Collective Manipulation of TO0n Bits



Caution When TOEmn = 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 to 2)
n: Channel number (n = 0 to 7)

6.4.5 Timer interrupt and TOmn pin output at count operation start

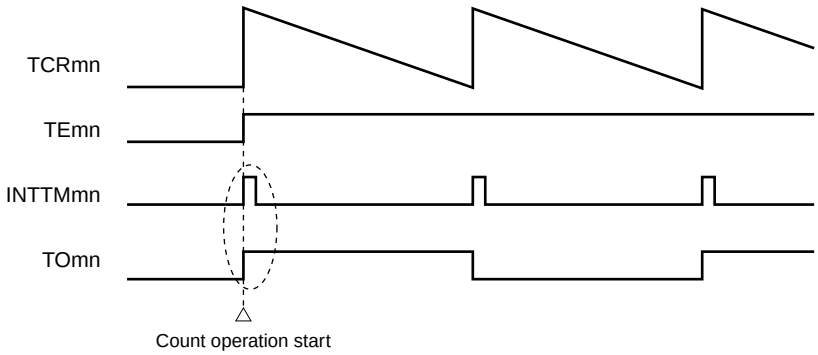
In the interval timer mode or capture mode, the MDmn0 bit in the TMRmn register sets whether or not to generate a timer interrupt at count start.

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.

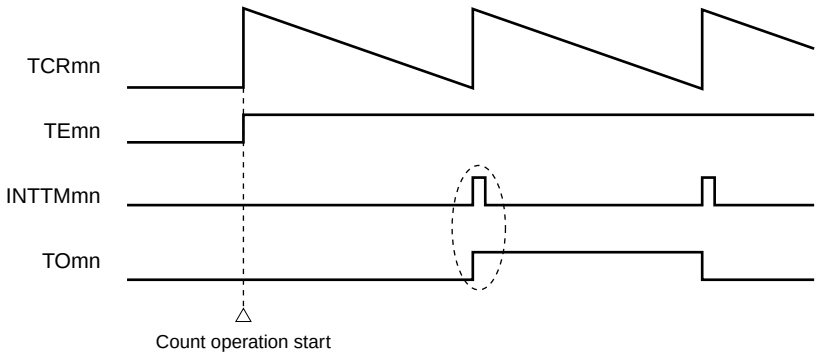
Figures 6-48 and 6-49 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-48. 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 6-49. 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 to 2)
n: Channel number (n = 0 to 7)

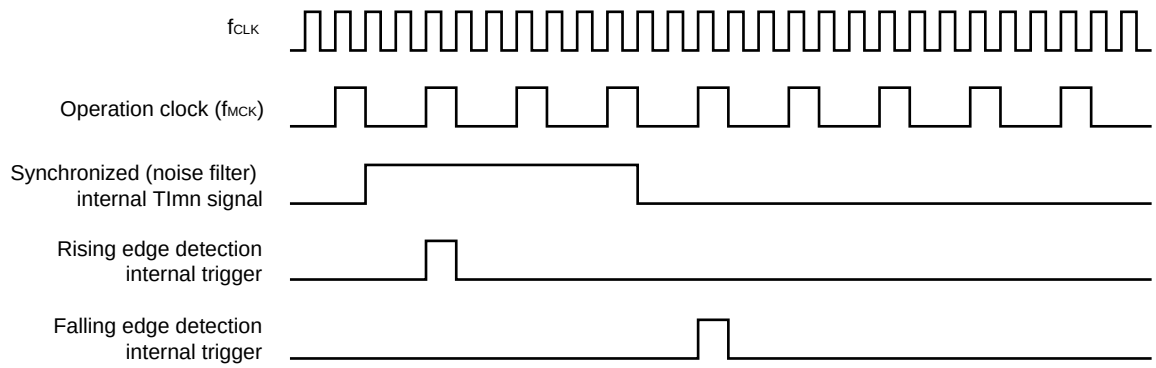
6.5 Channel Input (TImn Pin) Control

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}).

Figure 6-50. Edge Detection Basic Operation Timing



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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.

(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 software trigger, and count clock to the lower channels.
(8) A slave channel can use the INTTMmn (interrupt), start software trigger, and count clock of the master channel, but it cannot transmit its own INTTMmn (interrupt), start software trigger, and count clock to the lower channel.
(9) A master channel cannot use the INTTMmn (interrupt), start software trigger, and count clock from the other master channel.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

(10) To simultaneously start channels that operate in combination, the TSmn bit of the channels in combination must be set at the same time.
(11) During a counting operation, the TSmn bit of all channels that operate in combination or only the master channel can be set. TSmn of only a slave channel cannot be set.
(12) To stop the channels in combination simultaneously, the TTmn bit of the channels in combination must be set at the same time.

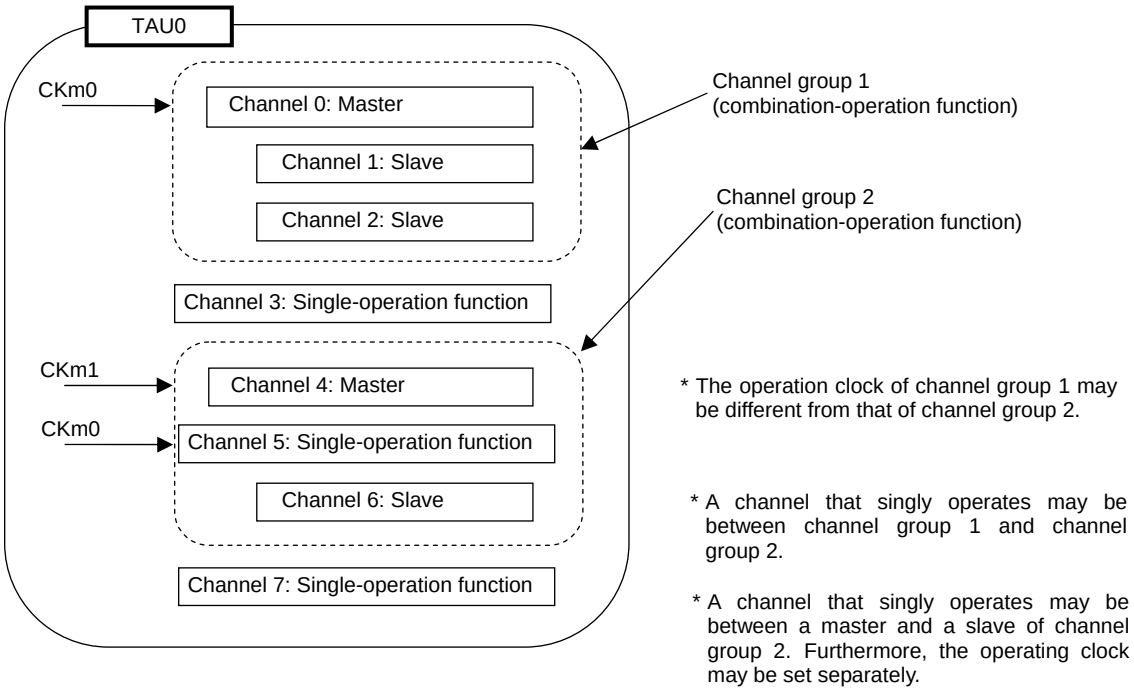
Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

6.6.3 Applicable range of basic rules of combination operation function

The rules of the combination operation function are applied in a channel group (a master channel and slave channels forming one combination operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the combination operation function in **6.6.2 Basic rules of combination operation function** do not apply to the channel groups.

Example



6.7 Operation of Timer Array Unit as Independent Channel

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

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.

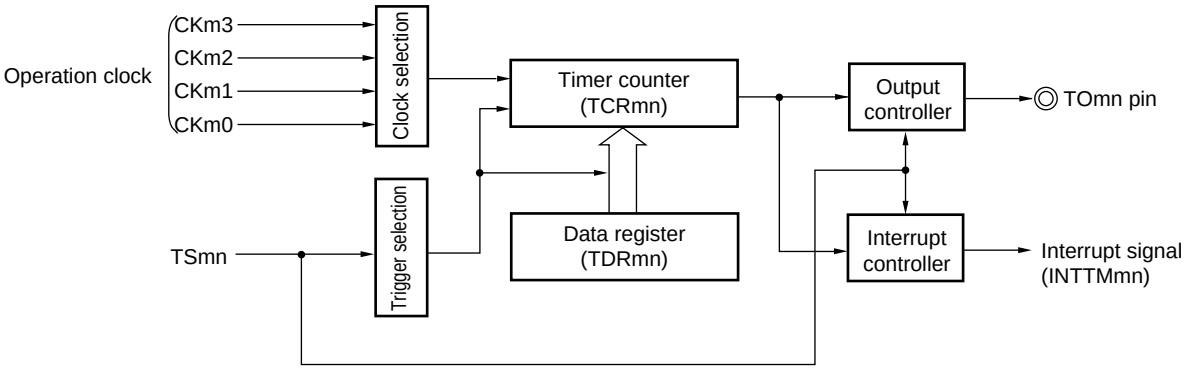
TCRmn loads the value of TDRmn at the first count clock after the channel start trigger bit (TSmn) is set to 1. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOmn is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOmn is toggled.

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.

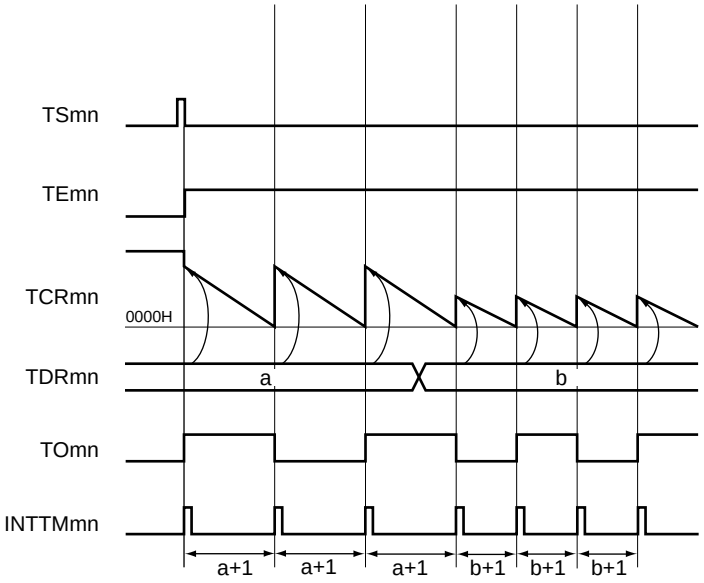
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid from the next period.

Figure 6-51. Block Diagram of Operation as Interval Timer/Square Wave Output



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

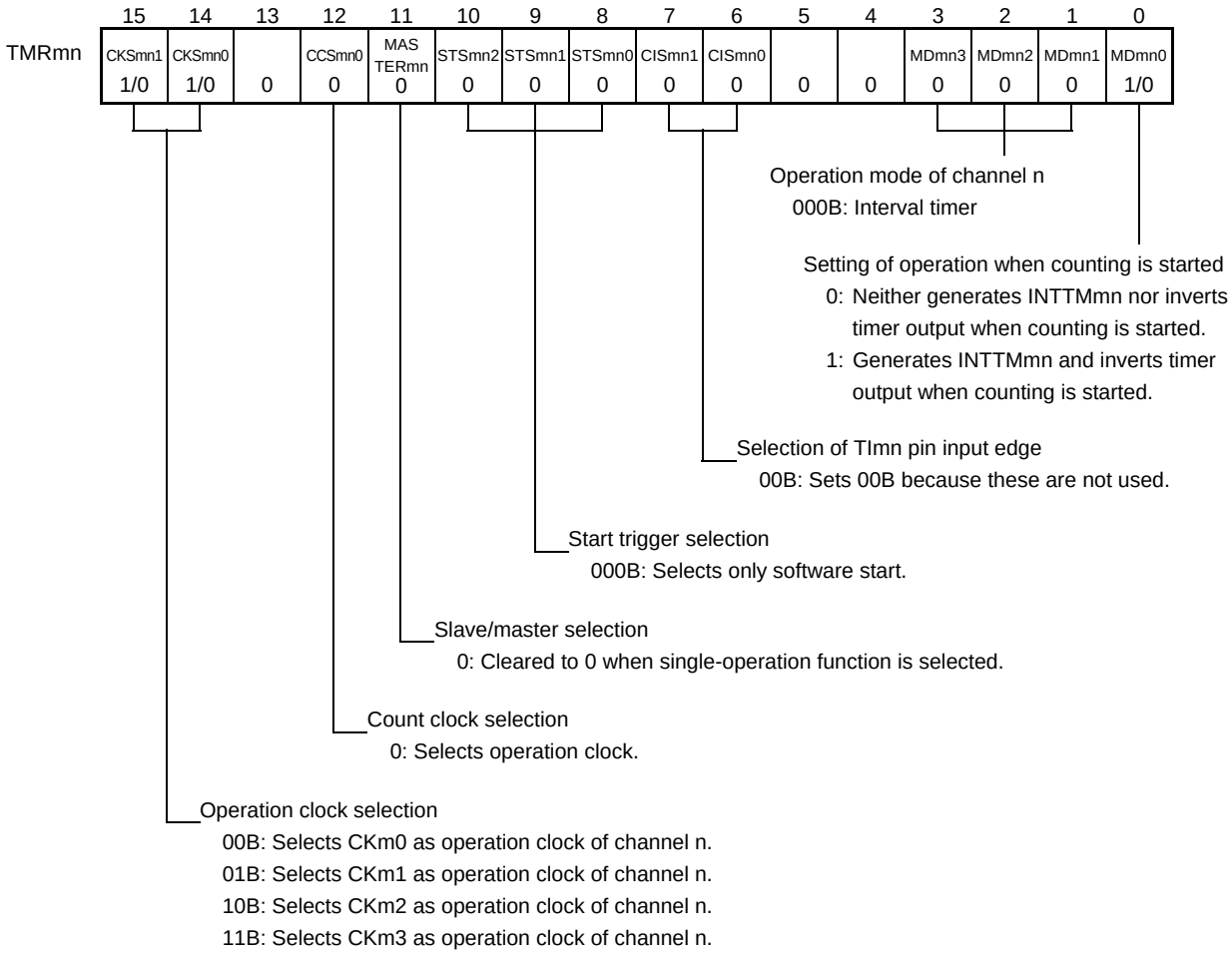
Figure 6-52. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-53. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 1/0	0: Outputs 0 from TOmn. 1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 1/0	0: Stops the TOmn output operation by counting operation. 1: Enables the TOmn output operation by counting operation.

(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 to 2)
n: Channel number (n = 0 to 7)

Figure 6-54. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 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 TOEmn 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 TOEmn to 1 (only when operation is resumed). Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and 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 TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The TOmn output is not initialized but holds current status.
	TOEmn 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 to 2)
n: Channel number (n = 0 to 7)

Figure 6-54. 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 TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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 (TSmn) is set to 1, TCRmn loads the value of TDRmn.

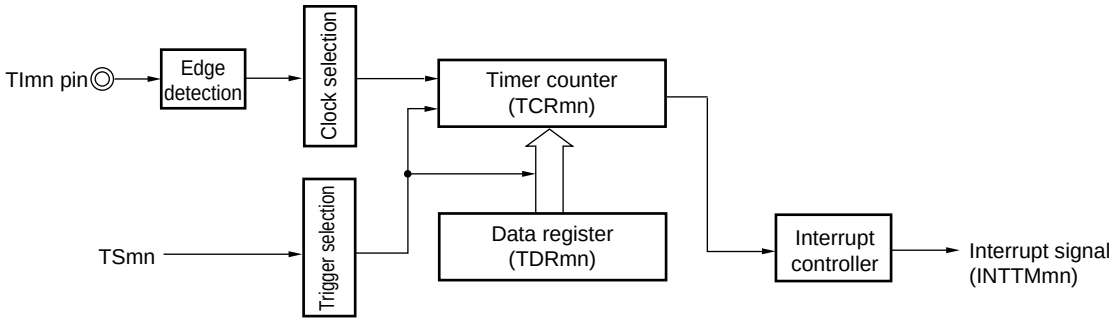
TCRmn counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, TCRmn loads the value of TDRmn again, and outputs INTTMmn.

After that, the above operation is repeated.

TOmn must not be used because its waveform depends on the external event and irregular.

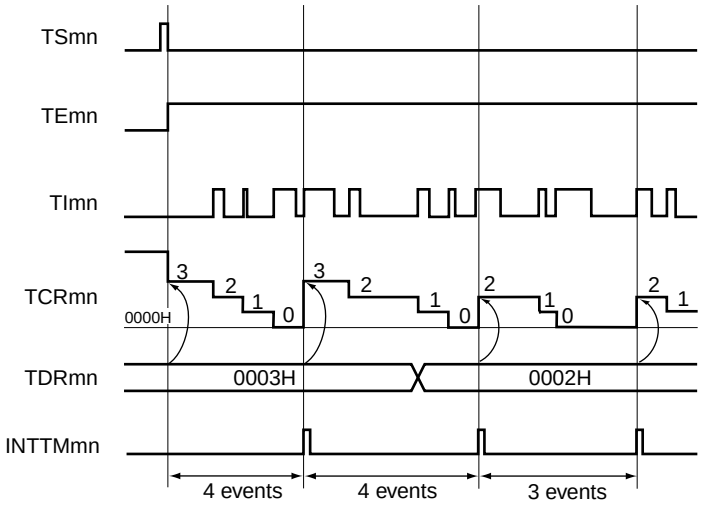
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid during the next count period.

Figure 6-55. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

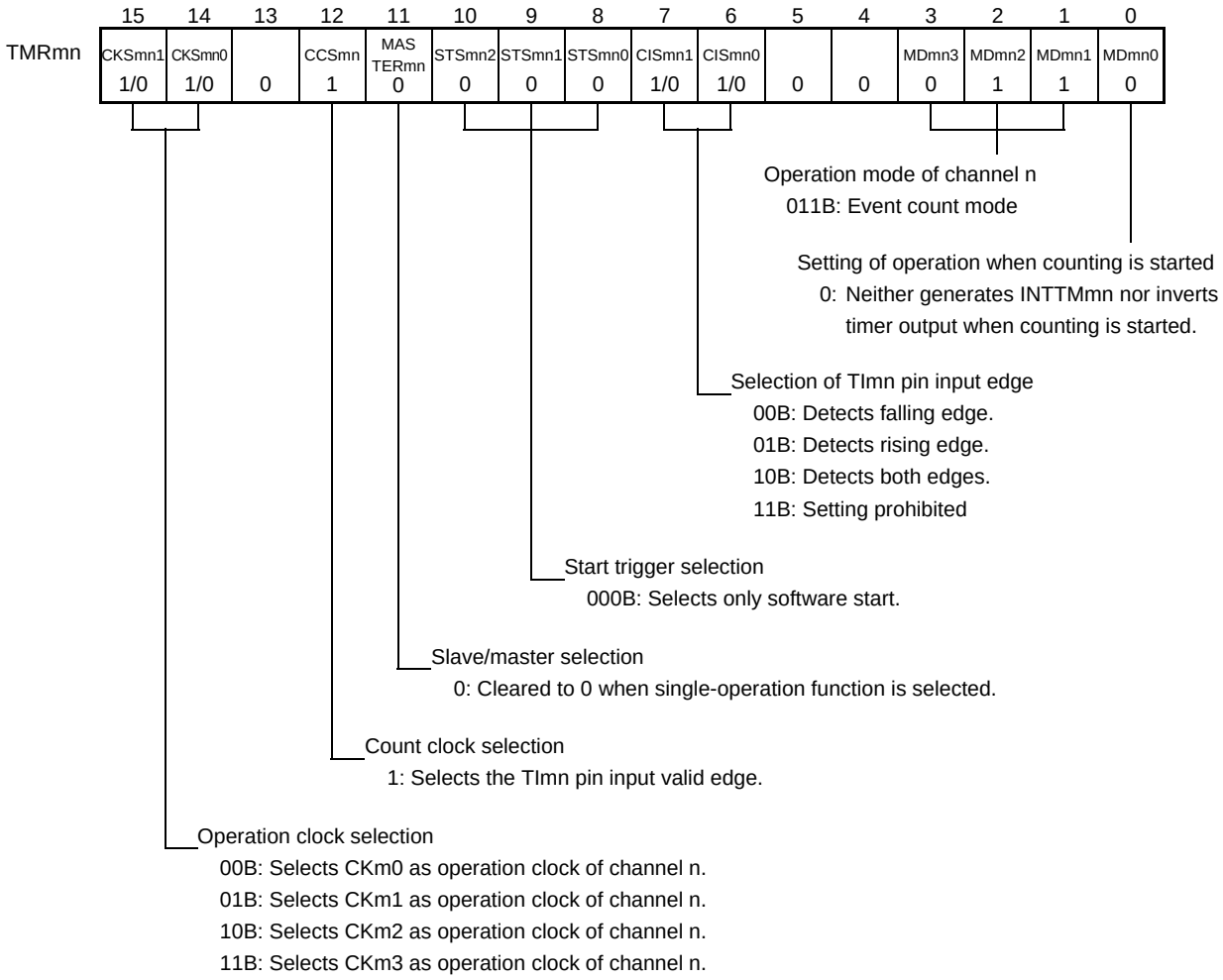
Figure 6-56. Example of Basic Timing of Operation as External Event Counter



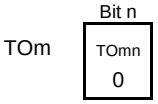
Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-57. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)

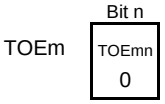


(b) Timer output register m (TOm)



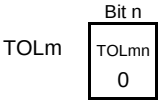
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (toggle mode).

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-57. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-58. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Sets number of counts to the TDRmn register. Clears the TOEmn bit of the TOEm register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. →	TEmn = 1, and count operation starts. Value of 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 TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of 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 TTmn bit is set to 1. →	TEmn = 0, and count operation stops. TCRmn holds count value and stops.
TAU stop	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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 (TSmn) 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.

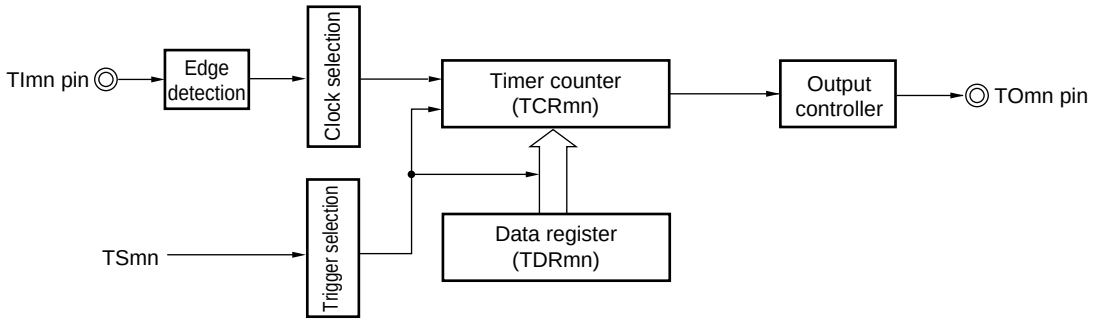
If detection of both the edges of TImn is selected, the duty factor error of the input clock affects the divided clock period of the TOmn output.

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)

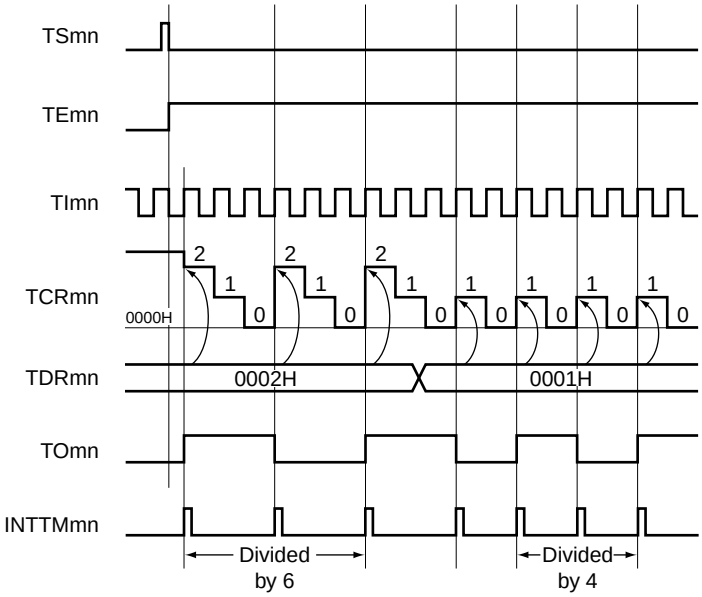
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid during the next count period.

Figure 6-59. Block Diagram of Operation as Frequency Divider



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

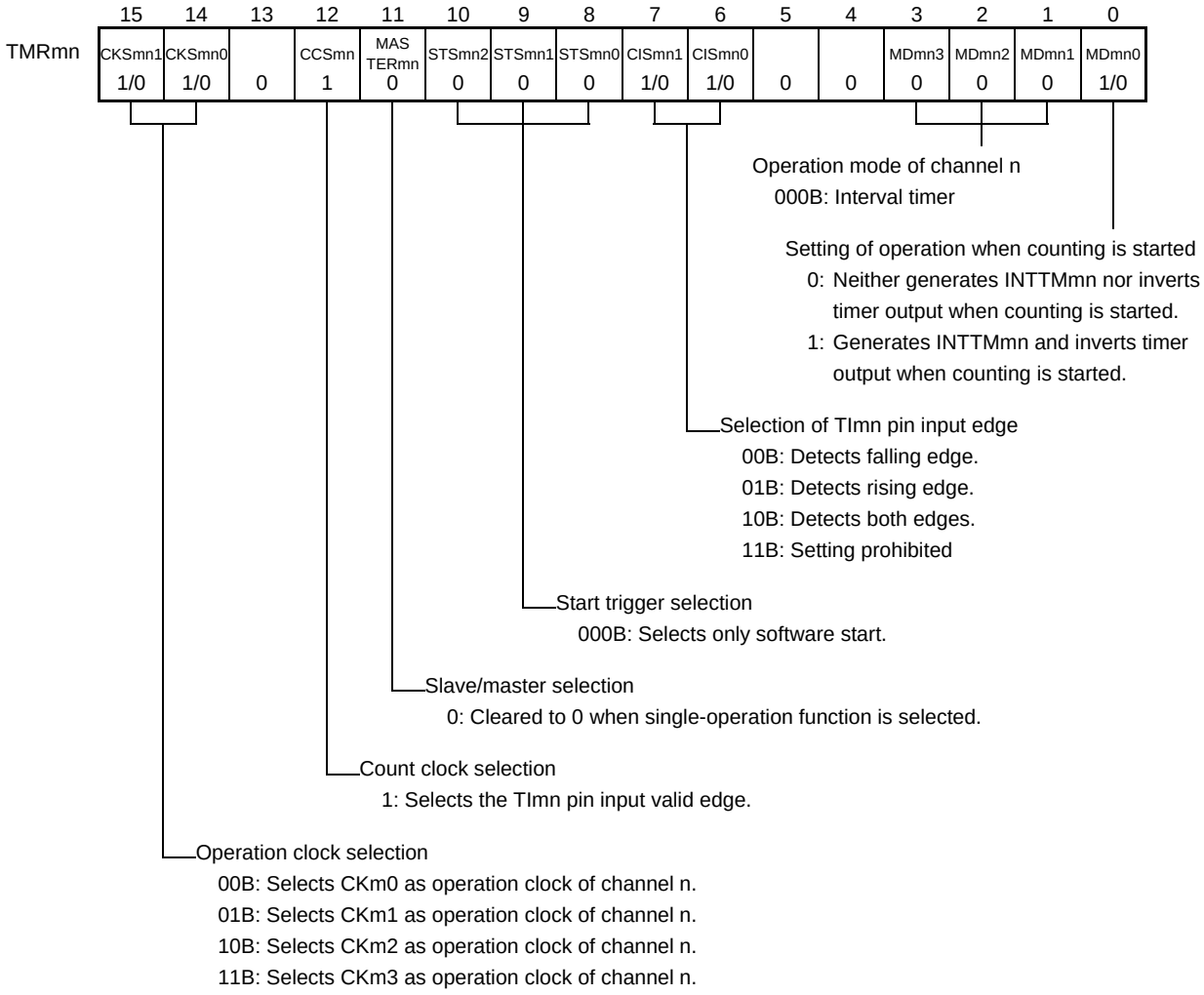
Figure 6-60. Example of Basic Timing of Operation as Frequency Divider (MDmn0 = 1)



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-61. Example of Set Contents of Registers When Frequency Divider Is Used (1/2)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 1/0	0: Outputs 0 from TOmn. 1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 1/0	0: Stops the TOmn output operation by counting operation. 1: Enables the TOmn output operation by counting operation.

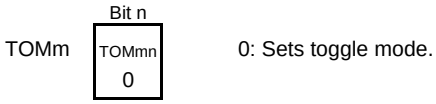
(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-61. Example of Set Contents of Registers When Frequency Divider Is Used (2/2)

(e) Timer output mode register m (TOMm)



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-62. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 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. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmn to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	Sets the TOEmn to 1 (only when operation is resumed). Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and 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 TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The TOmn output is not initialized but holds current status.
	TOEmn 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. →	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 TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode).

Operation is resumed.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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 (TSmn) 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.

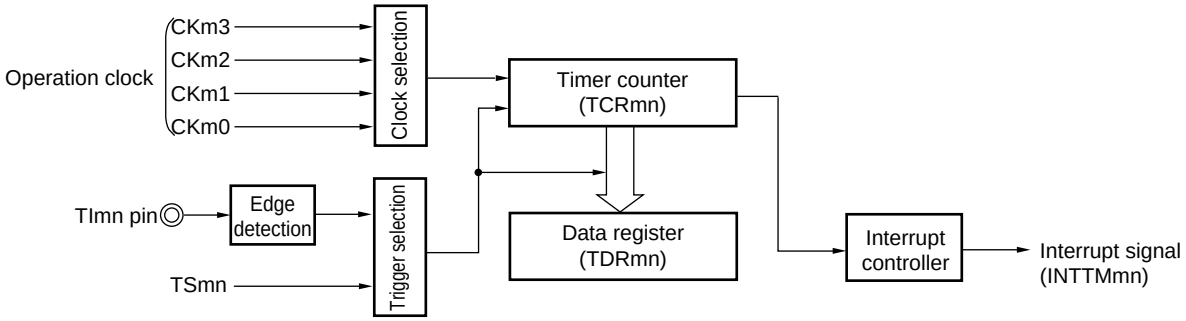
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

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.

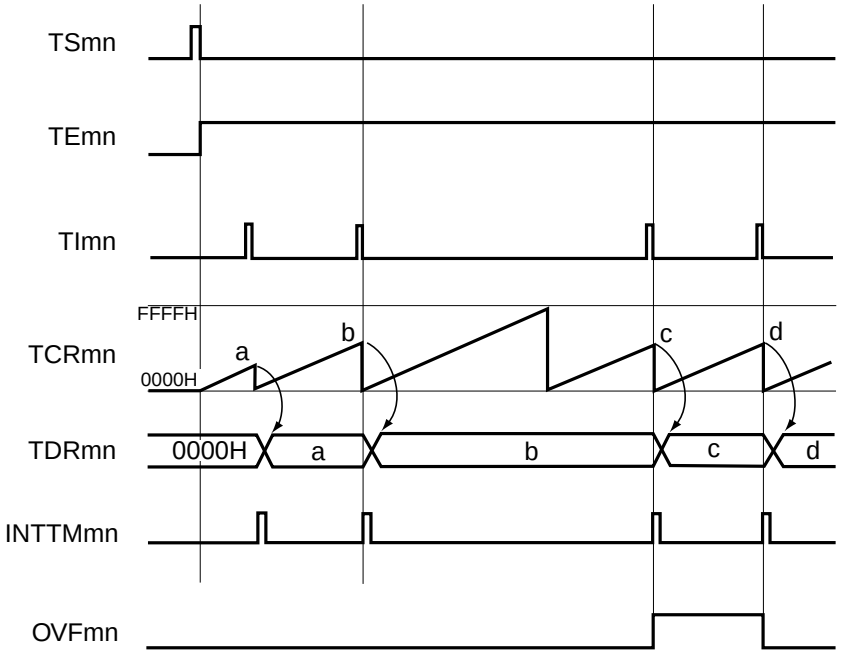
When TEmn = 1, instead of the TImn pin input, a software operation (TSmn = 1) can be used as a capture trigger.

Figure 6-63. Block Diagram of Operation as Input Pulse Interval Measurement



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

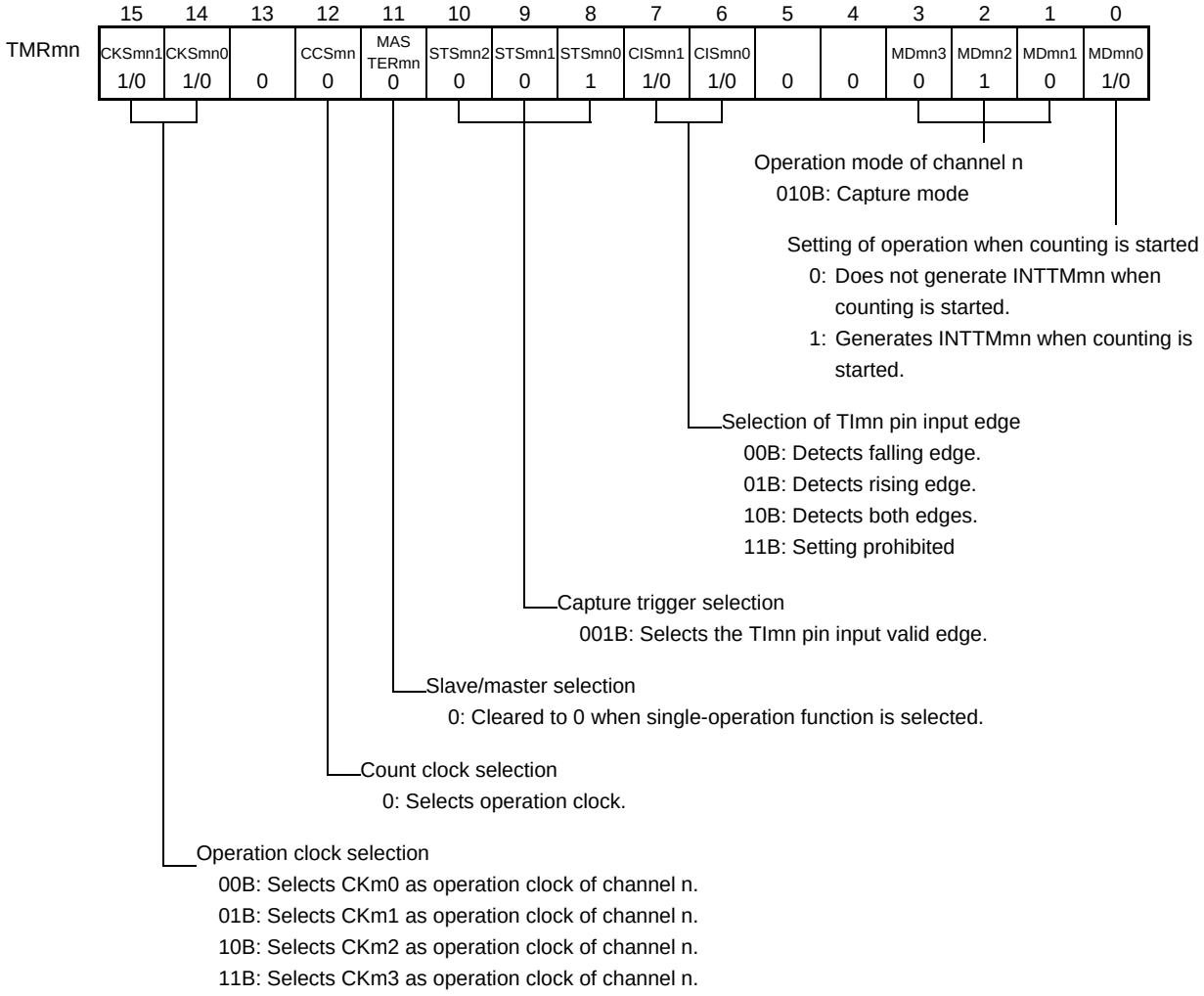
Figure 6-64. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



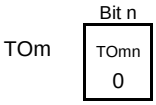
Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-65. Example of Set Contents of Registers to Measure Input Pulse Interval (1/2)

(a) Timer mode register mn (TMRmn)

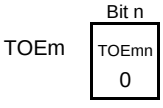


(b) Timer output register m (TOm)



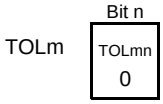
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (toggle mode).

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-65. Example of Set Contents of Registers to Measure Input Pulse Interval (2/2)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn	0: Sets toggle mode.
	0	

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-66. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 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 TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 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 TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to 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 TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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 (TSmn) is set to 1, TEmn 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.

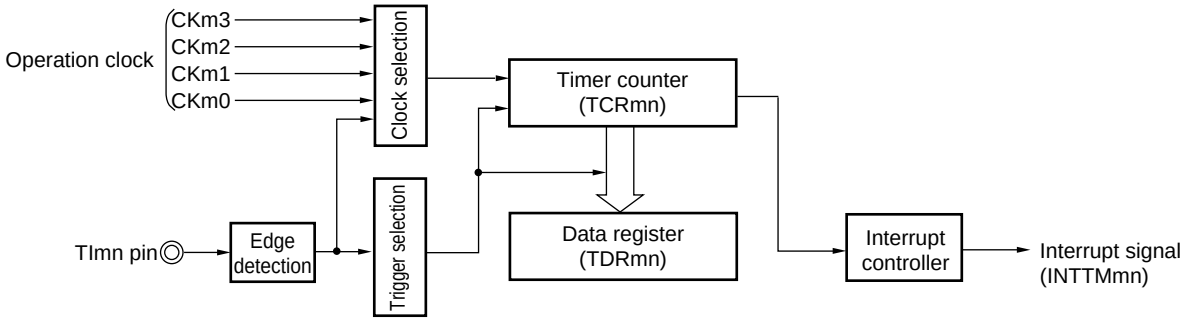
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, TSmn cannot be set to 1 while TEmn is 1.

CISmn1, CISmn0 of TMRmn = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn = 11B: High-level width is measured.

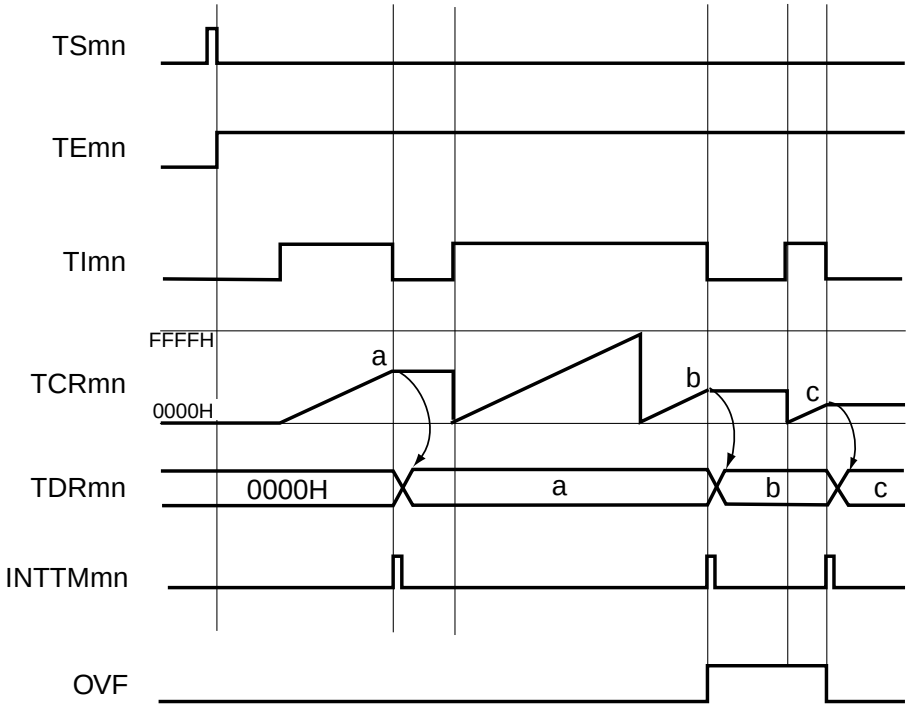
Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-67. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

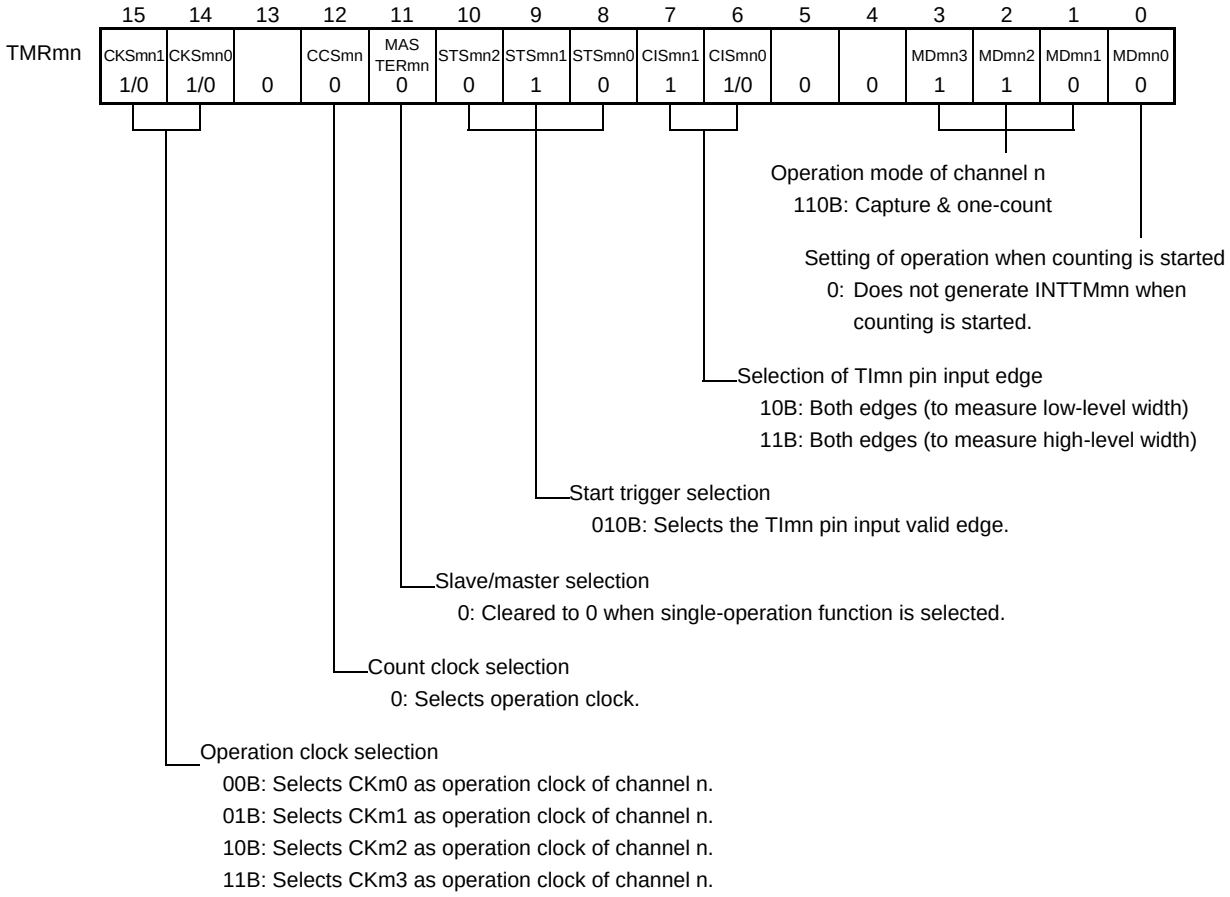
Figure 6-68. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-69. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn	0: Cleared to 0 when TOMmn = 0 (toggle mode).
	0	

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn	0: Sets toggle mode.
	0	

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-70. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Clears TOEmn to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects TImn pin input count start valid edge. →	Clears TCRmn to 0000H and starts counting up.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	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 TTmn bit is set to 1. → TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

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 (TSmn) is set to 1, INTTMmn is output. TCRmn counts down starting from the loaded value of TDRmn, in synchronization with the count clock. When TCRmn = 0000H, INTTMmn is output. TCRmn loads the value of TDRmn again. After that, it continues the similar operation.

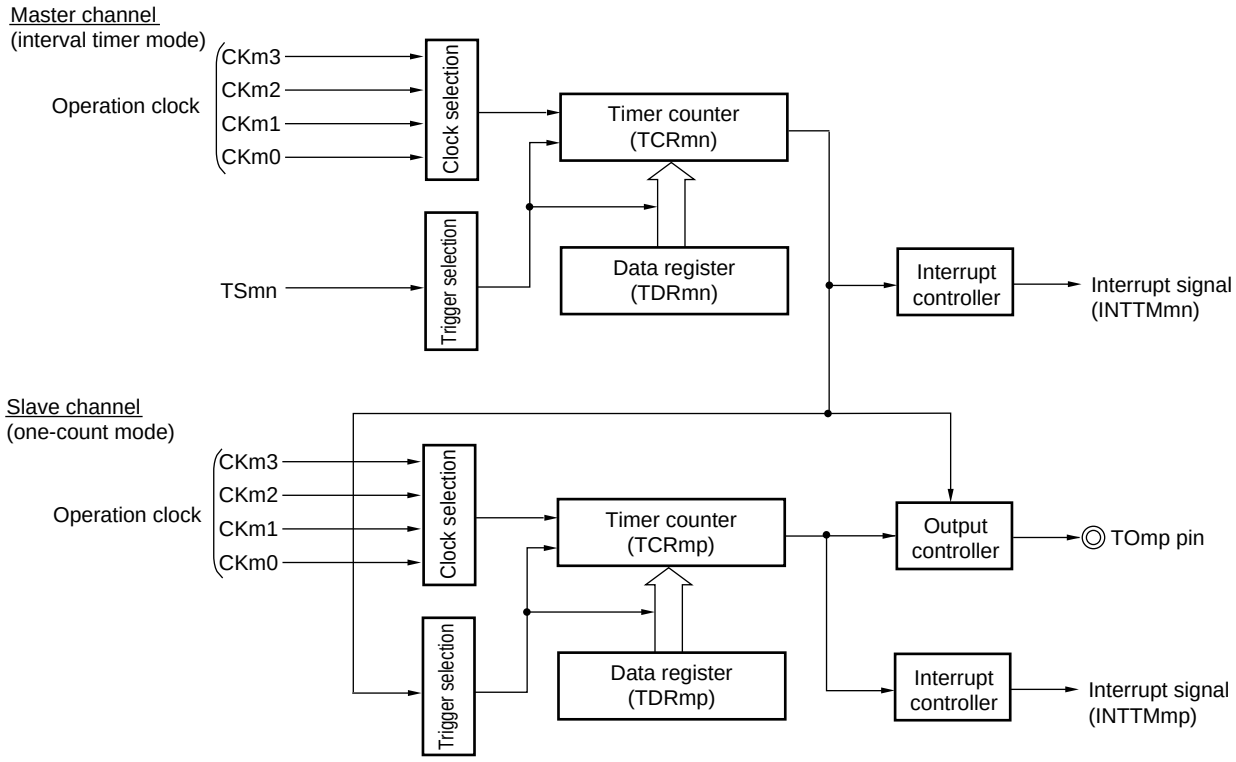
TCRmp of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. TCRmp of the slave channel loads the value of TDRmp, using INTTMmn of the master channel as a start trigger, and stops counting until the next start trigger (INTTMmn of the master channel) is input.

The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Caution To rewrite both TDRmn of the master channel and TDRmp of the slave channel, a write access is necessary two times. The timing at which the values of TDRmn and TDRmp are loaded to TCRmn and 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.

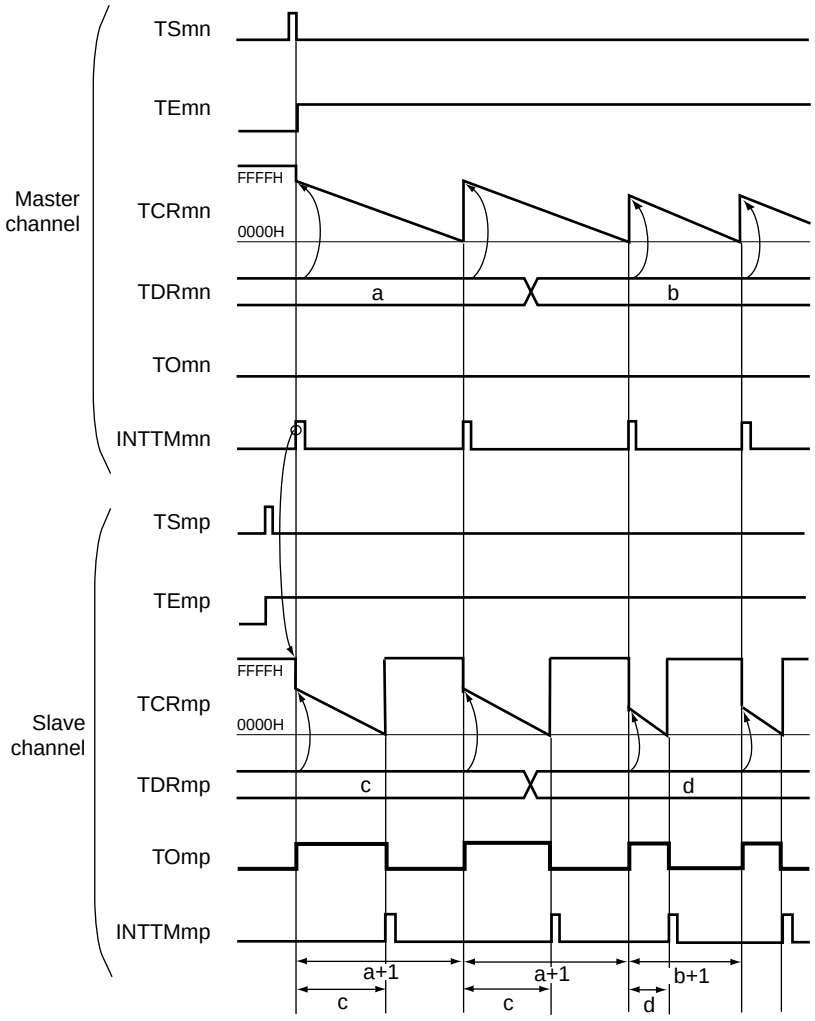
Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-71. Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

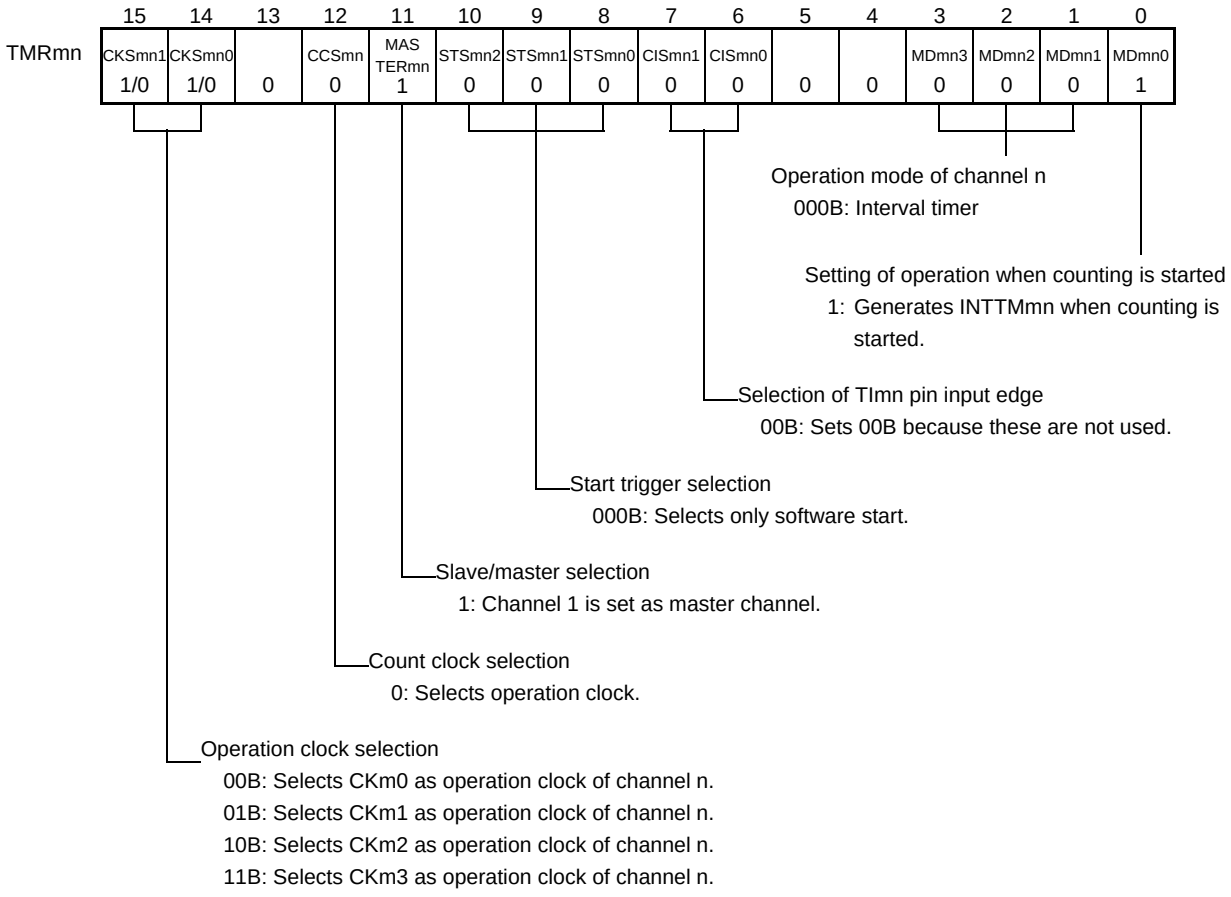
Figure 6-72. Example of Basic Timing of Operation as PWM Function



Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-73. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

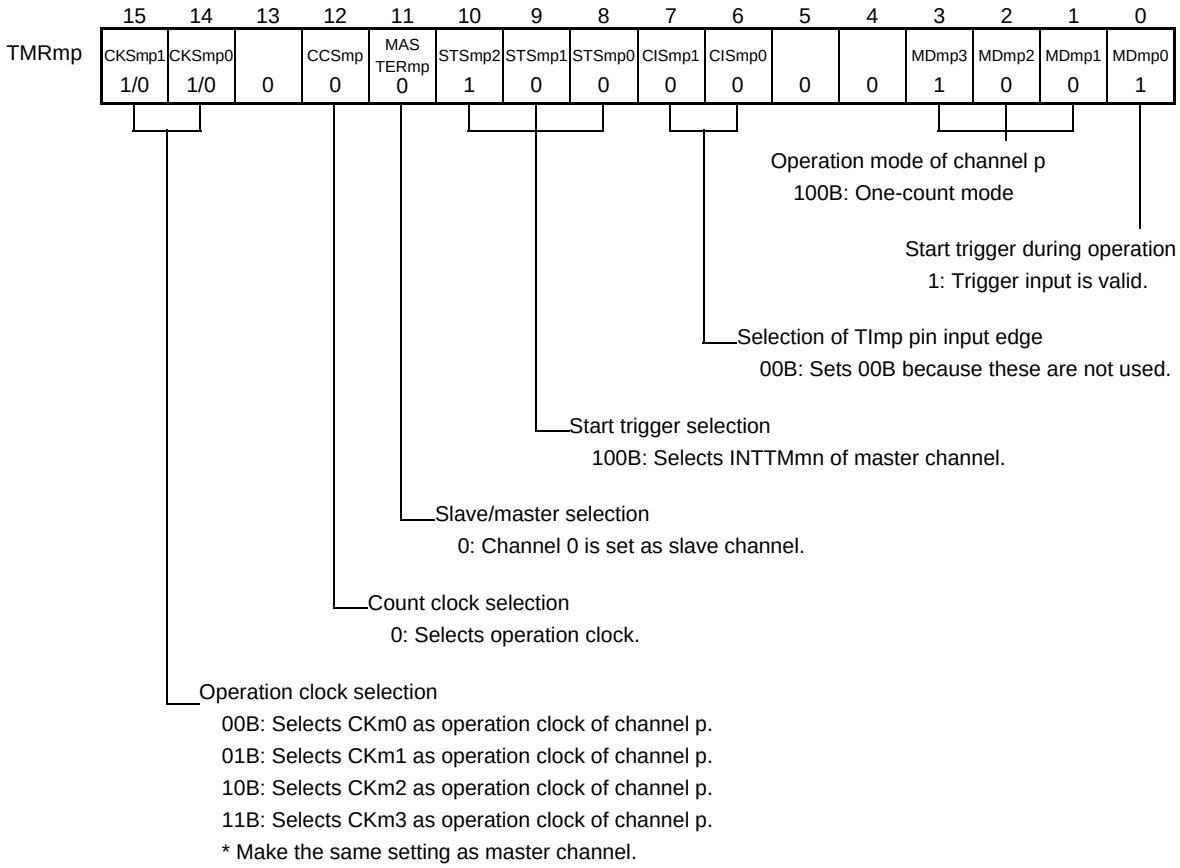
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-74. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the combination operation mode.

Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-75. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 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. 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 TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-75. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOEmp (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of the TSm register are set to 1 at the same time. →	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers cannot be changed.	The counter of the master channel loads the TDRmn value to TCRmn, and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to TCRmn, and the counter starts counting down again. At the slave channel, the value of TDRmp is loaded to TCRmp, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	
	TOEmp 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. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Switches the port mode register to input mode. →	The TOmp pin output level goes into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Operation is resumed.

Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

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.

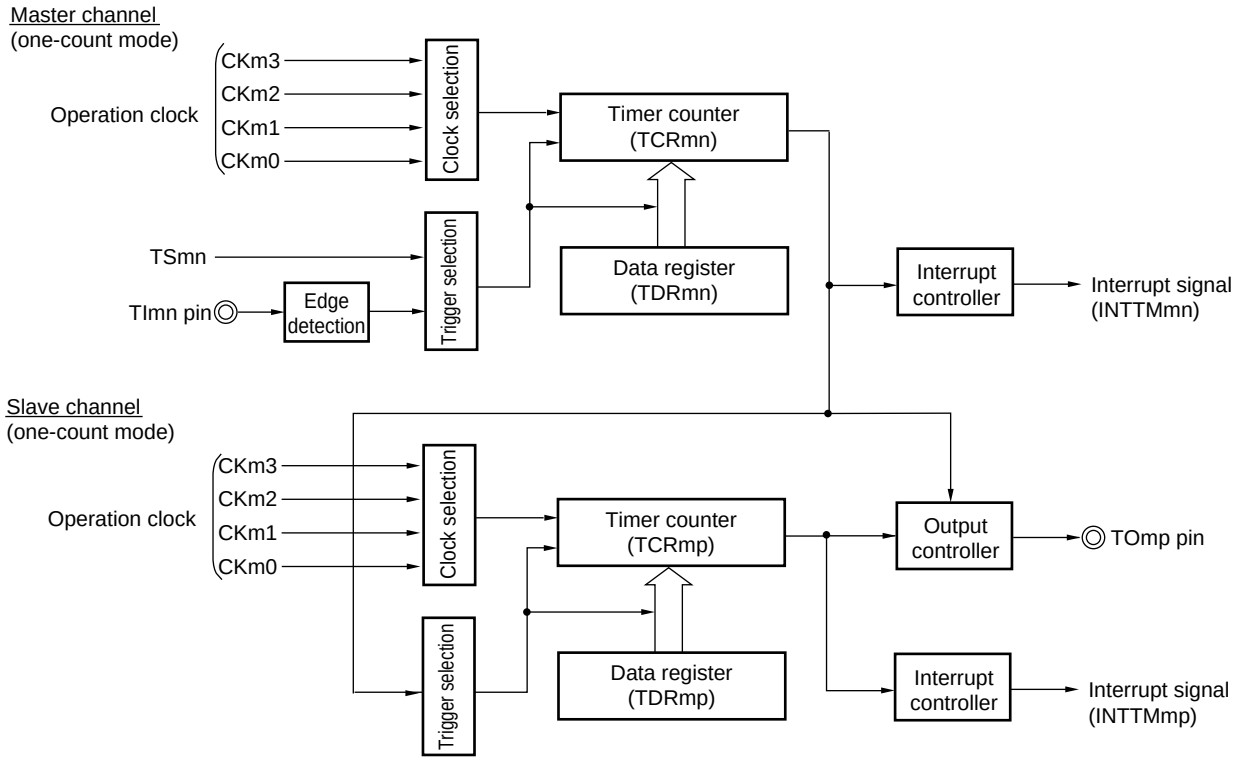
The slave channel operates in the one-count mode and counts the pulse width. TCRmp of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the TDRmp value. TCRmp counts down from the value of TDRmp it has loaded, in synchronization with the count value. 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 (TSmn = 1) as a start trigger.

Caution The timing of loading of TDRmn of the master channel is different from that of TDRmp of the slave channel. If TDRmn and TDRmp are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn after INTTMmn is generated and the TDRmp after INTTMmp is generated.

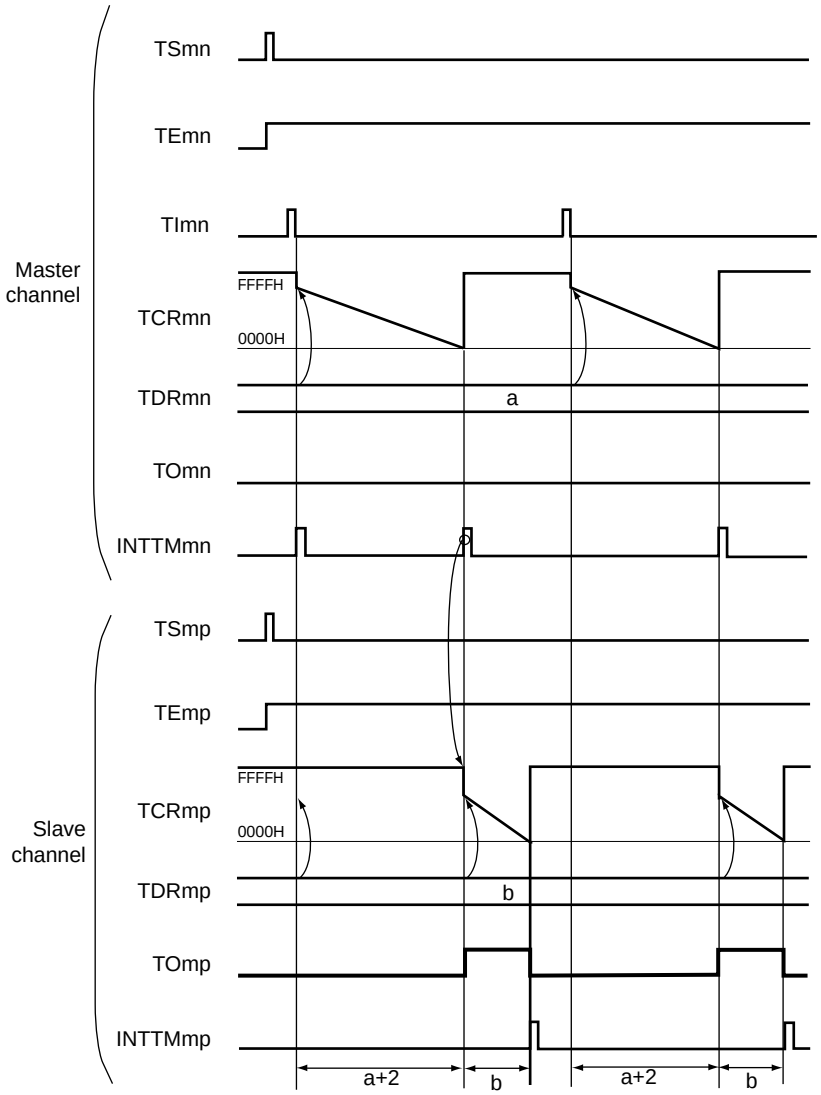
Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-76. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

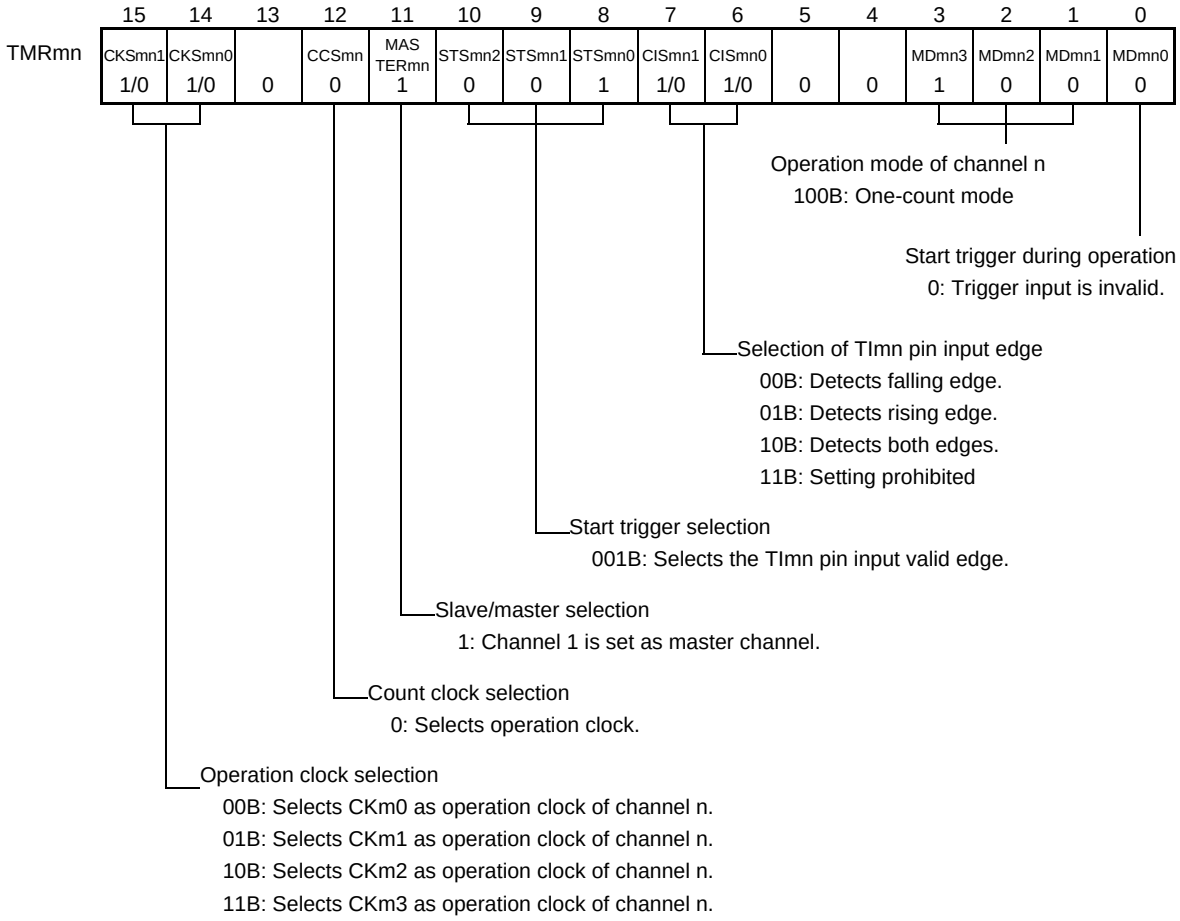
Figure 6-77. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-78. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

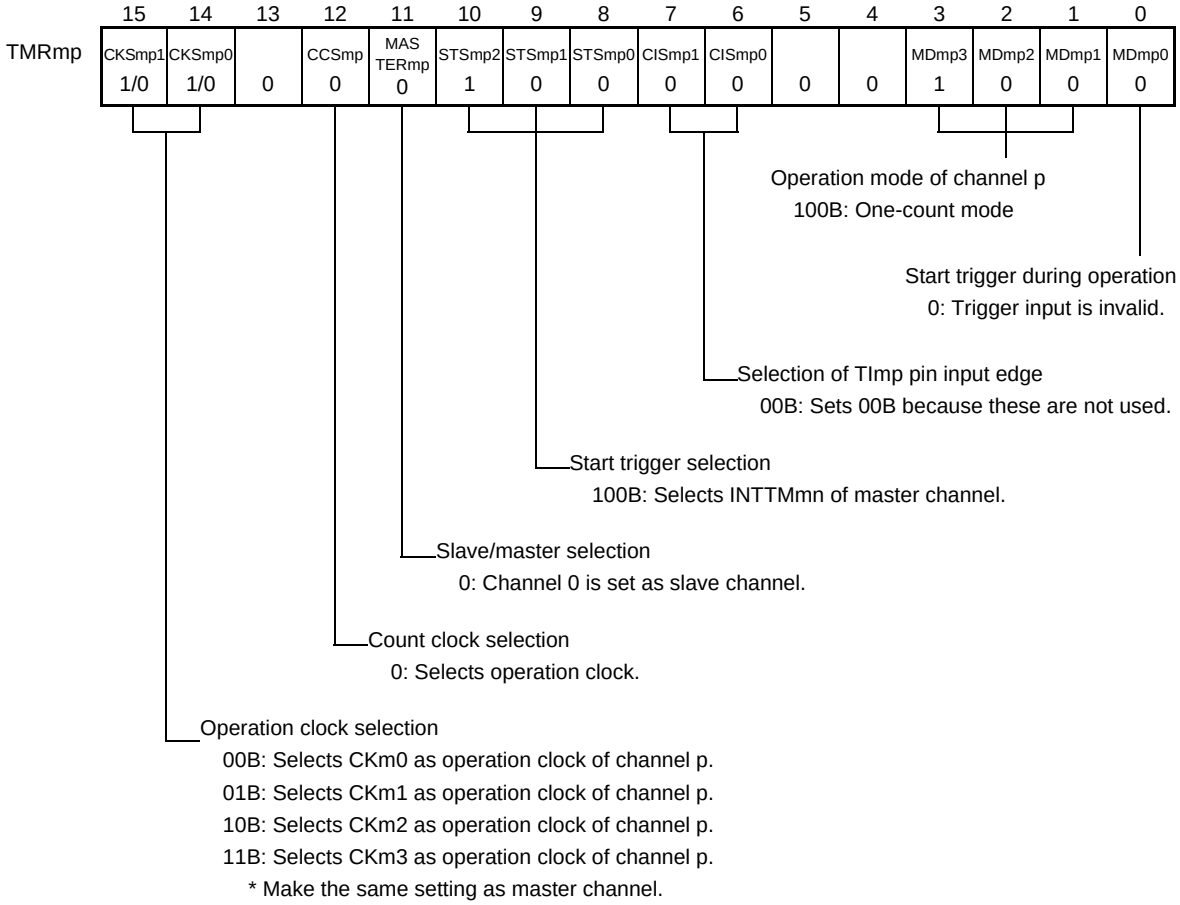
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-79. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the combination operation mode.

Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1)

Figure 6-80. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 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 TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1),

Figure 6-80. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOEmp (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of the TSm register are set to 1 at the same time. →	TEmn and TEmp 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. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers 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 TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEmp of slave channel is cleared to 0 and value is set to the TOmp 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. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Switches the port mode register to input mode. →	The TOmp pin output level goes into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Operation is resumed.

Remark m: Unit number (m = 0 to 2), n: Master channel number, p: Slave channel number (p = n + 1),

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.

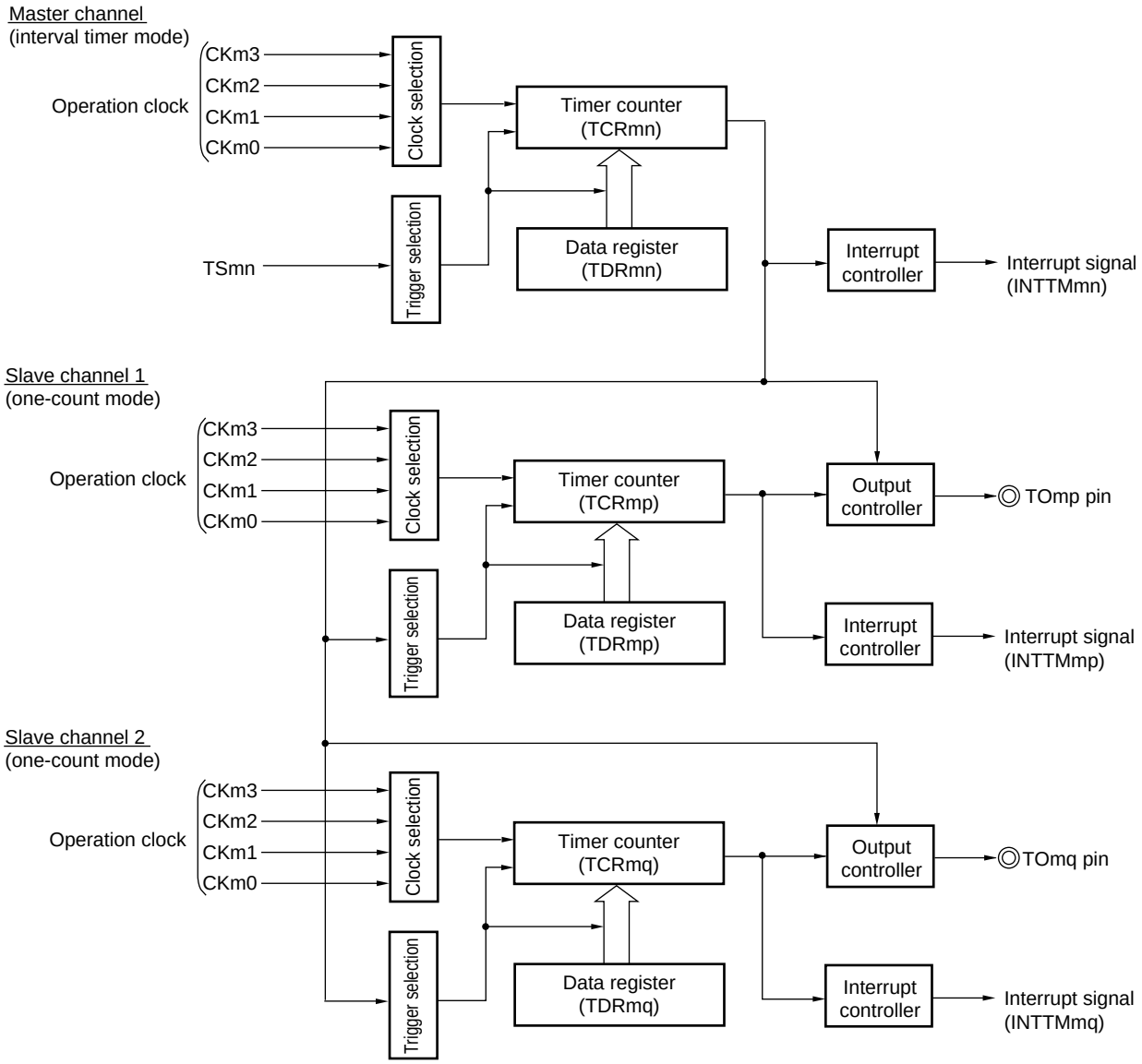
When channel 0 is used as the master channel as above, up to seven types of PWM signals can be generated for the timer array units 0 to 2.

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.)

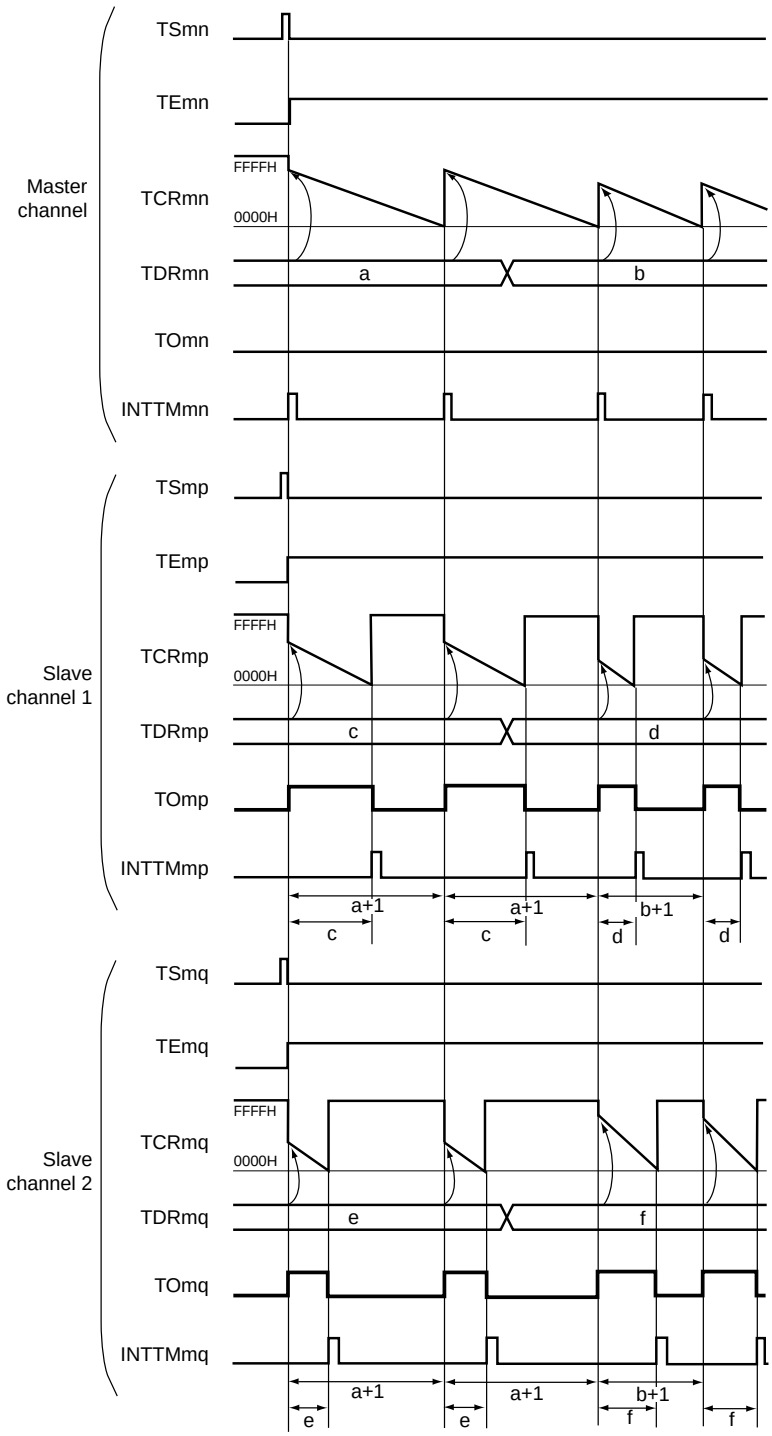
Remark m: Unit number (m = 0 to 2), n: Channel number, p, q: Slave channel number 1, 2 ($n < p < q \leq 7$)

Figure 6-81. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark m: Unit number (m = 0 to 2), n: Channel number, p, q: Slave channel number 1, 2 (n < p < q ≤ 7)

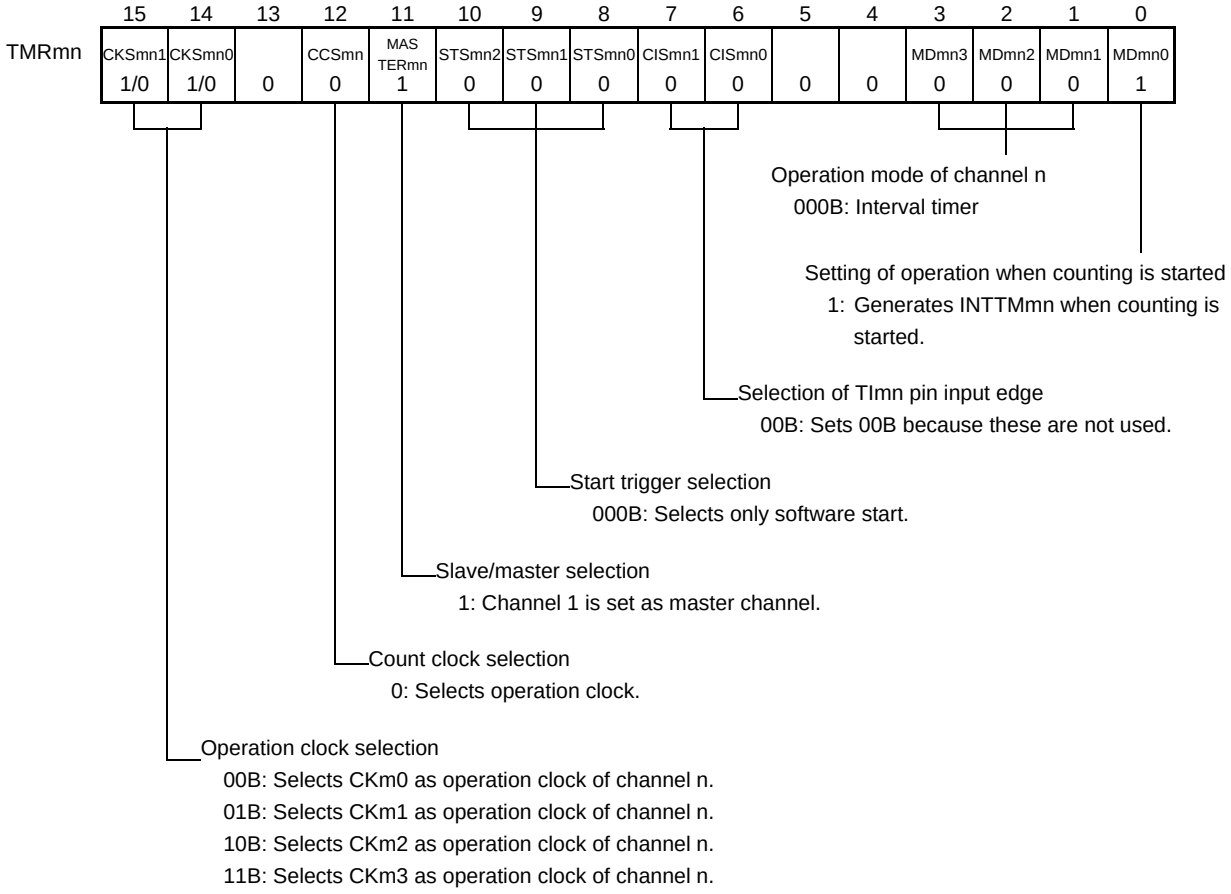
Figure 6-82. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark m: Unit number (m = 0 to 2), n: Channel number, p, q: Slave channel number 1, 2 ($n < p < q \leq 7$)

Figure 6-83. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

Remark m: Unit number (m = 0 to 2)
n: Channel number (n = 0 to 7)

Figure 6-84. 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.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	TOmp
	1/0	1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEmq	TOEmp
	1/0	1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq	TOLmp
	1/0	1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

Remark m: Unit number (m = 0 to 2), n: Channel number, p, q: Slave channel number 1, 2 (n < p < q ≤ 7)

Figure 6-84. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (2/2)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p	
TOMm	TOMmq	TOMmp	1: Sets the combination operation mode.
	1	1	

Remark m: Unit number (m = 0 to 2), n: Channel number, p, q: Slave channel number 1, 2 ($n < p < q \leq 7$)

Figure 6-85. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 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). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp and TOEmq 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 TOEmp and TOEmq (slave) to 1 (only when operation is resumed). The TSmn bit (master), and TSmp and TSmq (slave) bits of the TSm register are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.

Operation is resumed (on the next page).

Figure 6-85. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed (on the before page).

	Software Operation	Hardware Status
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used. Set values of the TOMm, TOLm, TOm, and TOEm registers 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 TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, and TEmq = 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.
	TOEmp or TOEmq 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.	The TOmp and TOmq pin output levels are held by port function.
	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 go into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0 to 2), n: Channel number, p, q: Slave channel number 1, 2 ($n < p < q \leq 7$)

CHAPTER 7 REAL-TIME CLOCK

7.1 Functions of Real-time Clock

The real-time clock has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)
- Pin output function of 1 Hz

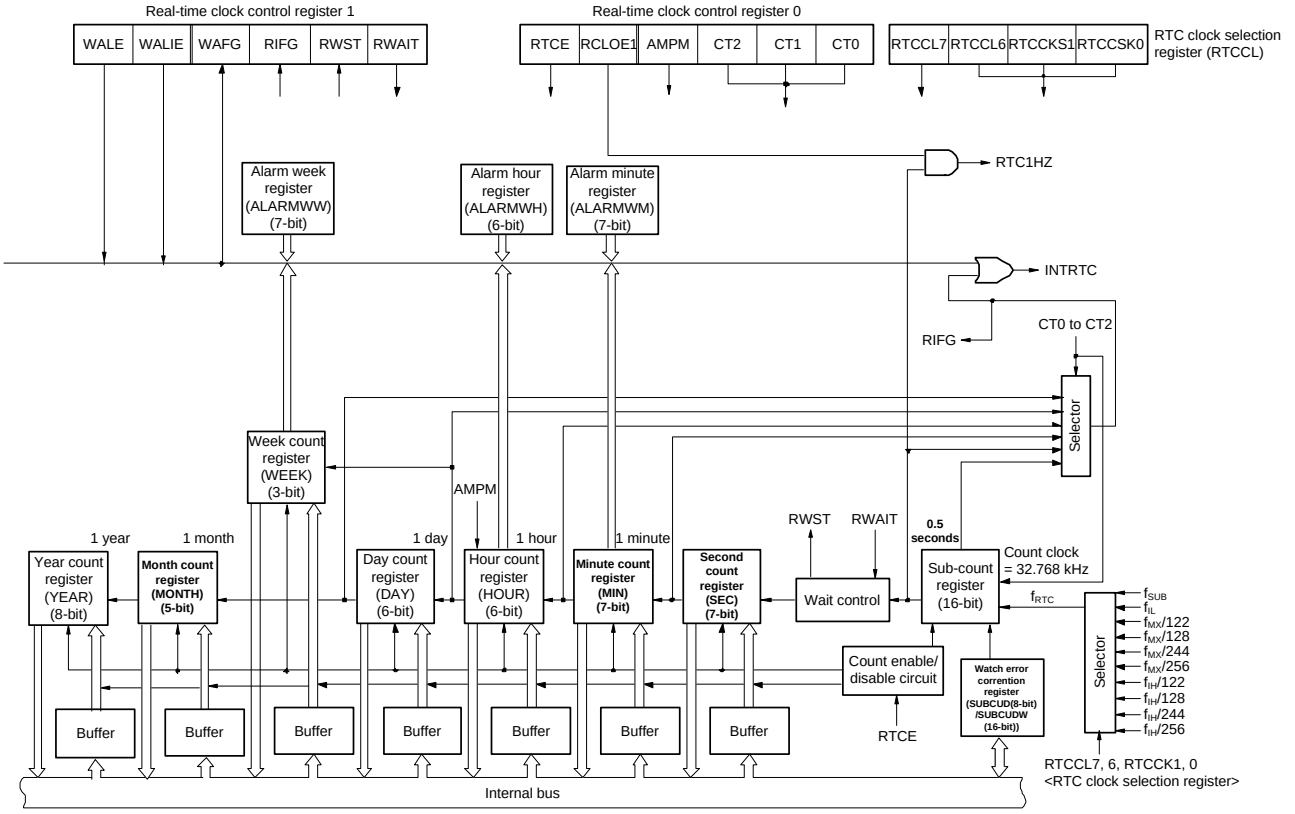
7.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 7-1. Configuration of Real-time Clock

Item	Configuration
Counter	Sub-count register
Control registers	Peripheral enable register 0 (PER0)
	RTC clock selection register (RTCCL)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	Watch error correction register (SUBCUDW)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)
	RTC1Hz pin select register (RTCSEL)

Figure 7-1. Block Diagram of Real-time Clock



Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) or the divided clock of f_{MX} and f_{IH} nearly equal to 32.768kHz is selected as the operation clock of the real-time clock.

When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

7.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- RTC clock selection register (RTCCL)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Watch error correction register (SUBCUDW)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- RTC1Hz pin select register (RTCSEL)

(1) Peripheral enable register 0 (PER0)

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

When the real-time clock is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

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

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

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

Cautions 1. When using the real-time clock, first set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCEN = 0, writing to a control register of the real-time clock or interval timer is ignored, and, even if the register is read, only the default value is read.

2. The subsystem clock supply to peripheral functions other than the real-time clock and interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the operation speed mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

(2) RTC clock selection register (RTCCL)

Figure 7-3. Format of RTC clock selection register (RTCCL)

Address: F00F9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	0	0	0	0	RTCCKS1	RTCCKS0

RTCCL7	Operation clock source selection for RTC/interval timer
0	RTC/interval timer uses External Main clock (f_{MX})
1	RTC/interval timer uses High-speed on-chip oscillator clock (f_{IH})

RTCCKS1	RTCCKS0	RTCCL6	Operation selection of RTC macro and interval timer
0	0	0/1	Sub clock (f_{SUB})
0	1		Low-speed on-chip oscillator clock (f_{IL},15K@typ) (WUTMMCK0 should be “1” to use this selection)
1	0	0	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7) /2^7
1	1	0	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7) /2^8
1	0	1	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7) /122
1	1	1	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7)/244

Caution WUTMMCK0 should be set to “1” when f_{IL} is used for RTC/interval timer clock.

(3) Watch error correction register (SUBCUDW)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the sub-count register to the second count register (SEC) (reference value: 7FFFH).

The SUBCUDW register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

<R> **Figure 7-4. Format of Watch Error Correction Register (SUBCUDW)**

Address: FFF34H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
SUBCUDW	DEV	0	0	F12	F11	F10	F9	F8

7	6	5	4	3	2	1	0
F7	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F12	Setting of watch error correction value
0	Increases by {(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.[Note]
When (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (0,0,0,0,0,0,0,0,0,0,0,0,0), (0,0,0,0,0,0,0,0,0,0,0,0,1), (1,0,0,0,0,0,0,0,0,0,0,0,0) or (1,0,0,0,0,0,0,0,0,0,0,0,1), the watch error is not corrected. Range of correction value: (when F12 = 0) 2, 4, 6, 8, … , 8186, 8188 (when F12 = 1) –2, –4, –6, –8, … , -8186, -8188	

Note "/" means bit-inverted values.

The range of value that can be corrected by using the watch error correction register (SUBCUDW) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	-12496.9 ppm to 12496.9 ppm	-4165.6 ppm to 4165.6 ppm
Maximum excludes quantization error	± 1.53 ppm	± 0.51 ppm
Minimum resolution	± 3.05 ppm	± 1.02 ppm

Caution When correcting the RTC, use either this register or the watch error correction register (SUBCUD) in (13).

Remark If the correctable range is -4165.6 ppm or lower and 4165.6 ppm or higher, set DEV to 0.

(4) Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the constant-period interrupt function.

The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of Real-time Clock Control Register 0 (RTCC0)

Address: FFF5DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system. • Table 7-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Caution Do not change the value of the RTCLOE1 bit when RTCE = 1.

Remark ×: don't care

(5) Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-6. Format of Real-time Clock Control Register 1 (RTCC1) (1/2)

Address: FFF5EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Figure 7-7. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST	Wait status flag of real-time clock
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time clock
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the sub-count register is continuing to run, complete reading or writing within one second and turn back to 0. When RWAIT = 1, it takes up to 1 clock (f_{RTC}) until the counter value can be read or written (RWST = 1). When the sub-count register overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

(6) Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-8. Format of Second Count Register (SEC)

Address: FFF52H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

(7) Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-9. Format of Minute Count Register (MIN)

Address: FFF53H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

(8) Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Figure 7-10. Format of Hour Count Register (HOUR)

Address: FFF54H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Cautions 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

Table 7-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 7-2. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

(9) Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days. It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-11. Format of Day Count Register (DAY)

Address: FFF56H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

(10) Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Set a decimal value of 00 to 06 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-12. Format of Week Count Register (WEEK)

Address: FFF55H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Cautions 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

(11) Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-13. Format of Month Count Register (MONTH)

Address: FFF57H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

(12) Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (f_{RTC}) later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-14. Format of Year Count Register (YEAR)

Address: FFF58H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

(13) Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the sub-count register to the second count register (SEC) (reference value: 7FFFH).

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

Reset signal generation clears this register to 00H.

<R>

Figure 7-15. Format of Watch Error Correction Register (SUBCUD)

Address: FFF59H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.[Note1]
When (F6,F5,F4,F3,F2,F1,F0)=(0,0,0,0,0,0,0) or (0,0,0,0,0,0,1), the watch error is not corrected. Range of correction value: (when F6=0) 2,4,6,7,...,120,122 (when F6=1) -2,-4,-6,-8,...,-120,-122,-124[Note 2]	

Notes 1. "/" means bit-inverted values.

2. It is not recommended to set (F6,F5,F4,F3,F2,F1,F0)=(1,0,0,0,0,0,0) or (1,0,0,0,0,0,1).

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	-189.2 ppm to 189.2 ppm	-63.1 ppm to 63.1 ppm
Maximum excludes quantization error	± 1.53 ppm	± 0.51 ppm
Minimum resolution	± 3.05 ppm	± 1.02 ppm

Caution When correcting the RTC, use either this register or the watch error correction register (SUBCUDW) in (3). When SUBCUDW is used, however, the correction value cannot be judged correctly by reading SUBCUD.

Remark If a correctable range is -63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

(14) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-16. Format of Alarm Minute Register (ALARMWM)

Address: FFF5AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(15) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-17. Format of Alarm Hour Register (ALARMWH)

Address: FFF5BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(16) Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 7-18. Format of Alarm Week Register (ALARMWW)

Address: FFF5CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

(17) RTC1Hz pin select register (RTCSEL)

This register is used to select the pin to output RTC1Hz signal. The RTCSEL register can be selected by 1-bit or 8-bit memory manipulation instruction. Reset signal generation clears this register to 00H.

Figure 7-19. Format of RTC1Hz pin Select Register (RTCSEL)

Address: FFFF36 After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCSEL	RTCOSEL1	RTCOSEL0	0	0	RTCTIS11	RTCTIS10	RTCTIS01	RTCTIS00

RTCTIS00	Switch RTC1Hz output to TAU TI06 input or not
0	Disconnected to TI06
1	Connected to TI06

RTCTIS01	Switch RTC1Hz output to TAU TI07 input or not
0	Disconnected to TI07
1	Connected to TI07

RTCTIS10	Switch RTC1Hz output to TAU TI16 input or not
0	Disconnected to TI16
1	Connected to TI16

RTCTIS11	Switch RTC1Hz output to TAU TI17 input or not
0	Disconnected to TI17
1	Connected to TI17

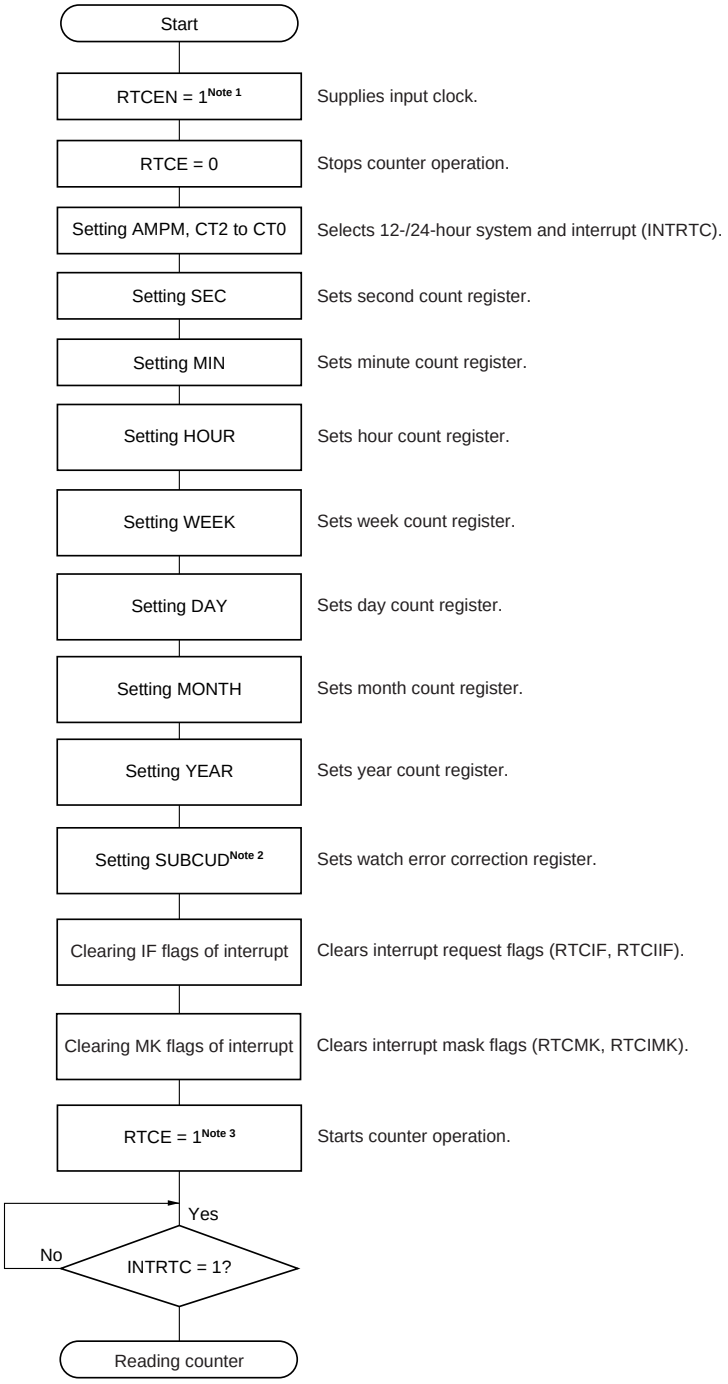
RTCOSEL1	RTCOSEL0	RTC1Hz output pin selection
0	0	P64
0	1	P15
1	0	P94
1	1	No port is selected (Output disabled)

To measure 1Hz, two channels of TAU should be used because 16-bit counter will be overflowed if f_{CLK} is fast frequency. A channel is operated in pulse width measurement mode. Low-level or high-level width of 1Hz pulse is typically 500ms. Another channel is operated in interval timer mode (start trigger is set to TIN edge) and number of overflow should be counted by software at the interrupt timing. The measurement is finished when interrupt by capture channel is occurred. The interval time can be calculated by software-overflow-counter and TDR register of capture channel.

7.4 Real-time Clock Operation

7.4.1 Starting operation of real-time clock

Figure 7-20. Procedure for Starting Operation of Real-time Clock



Notes 1. First set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable.

2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.6 Example of watch error correction of real-time clock**.

3. Confirm the procedure described in **7.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

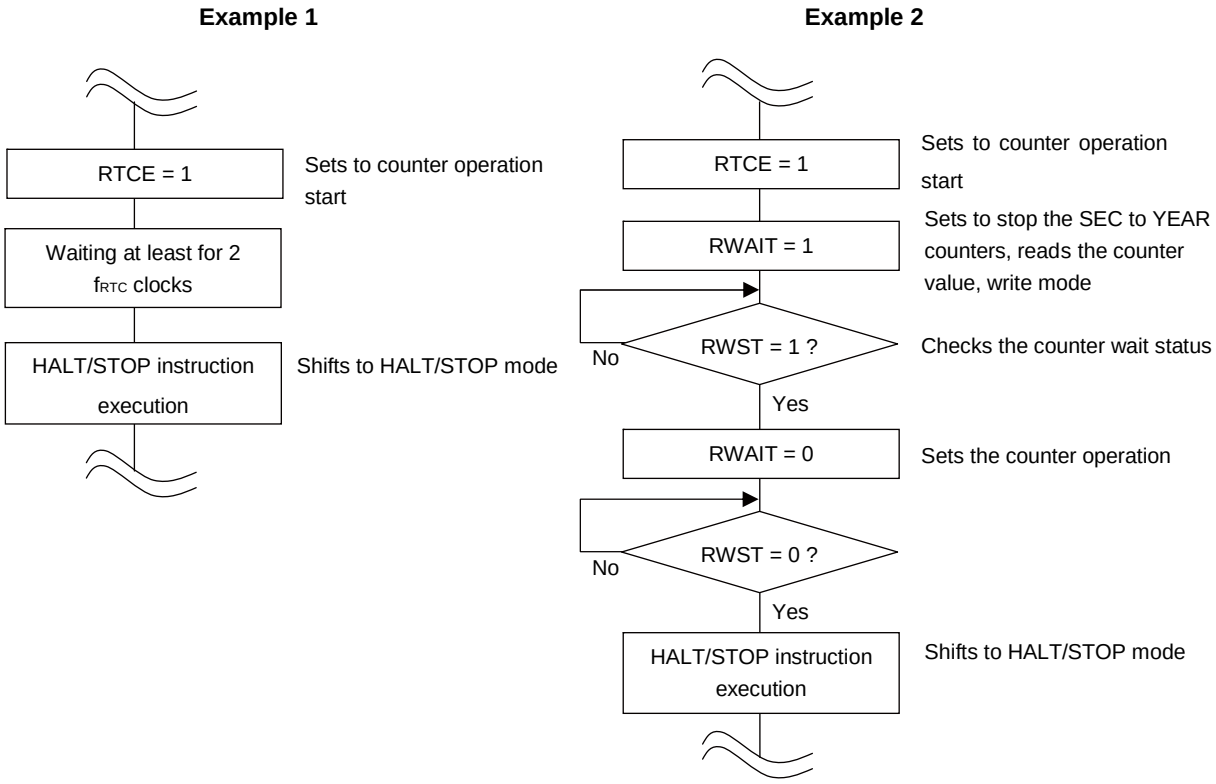
7.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after the first INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two input clocks (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 7-21**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 7-21**, **Example 2**).

Figure 7-21. Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1

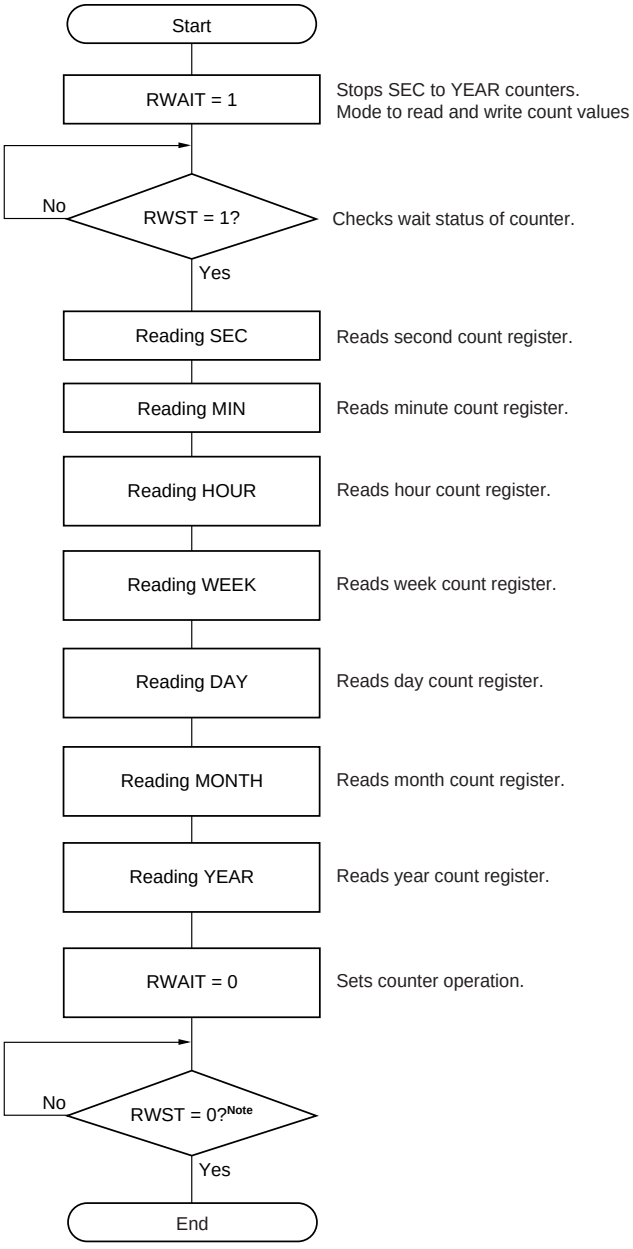


7.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

Figure 7-22. Procedure for Reading Real-time Clock



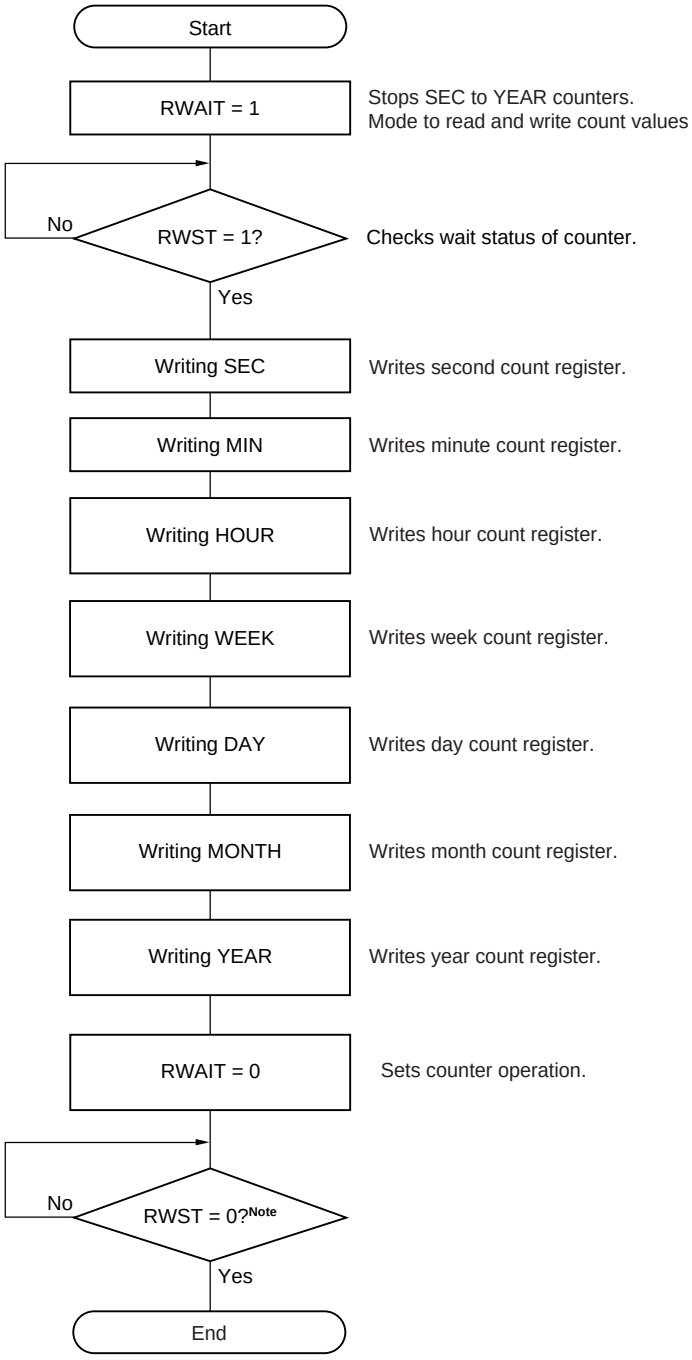
Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence.

All the registers do not have to be set and only some registers may be read.

Figure 7-23. Procedure for Writing Real-time Clock



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

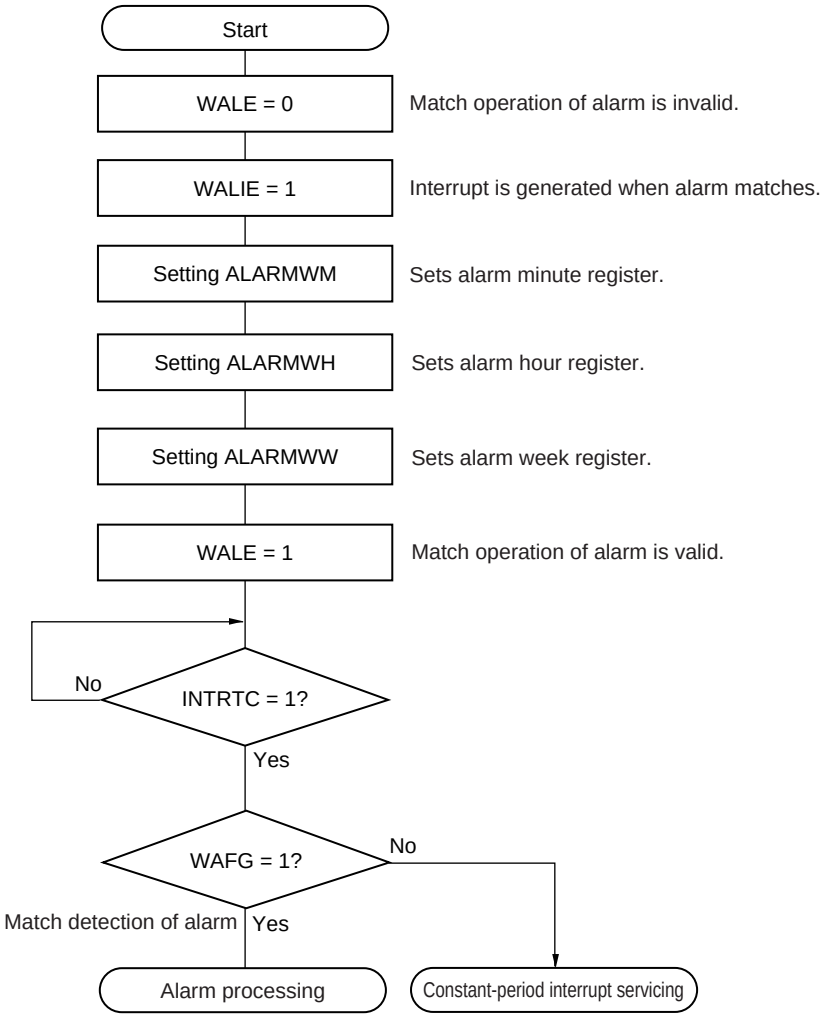
Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

7.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE first.

Figure 7-24. Alarm Setting Procedure

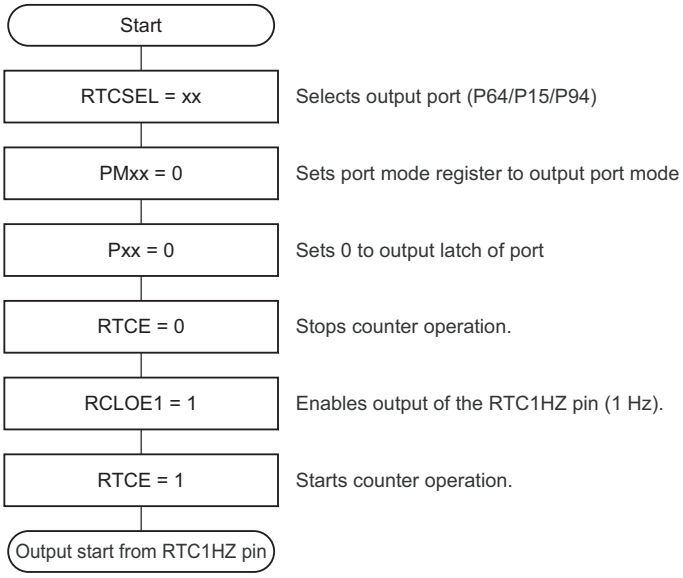


Remarks 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

7.4.5 1 Hz output of real-time clock

Figure 7-25. 1 Hz Output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the input clock (f_{SUB}) is stable.

7.4.6 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is -4165.6 ppm or less, or 4165.6 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 12 to 0 of the watch error correction register (SUBCUDW).

(When F12 = 0) Correction value = {(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) – 1} × 2

(When F12 = 1) Correction value = – {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F12 to F0) is (*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1.

<R> /F11 to /F0 are bit-inverted values (000000000011 when 111111111100).

<R> **Remarks 1.** The correction value is 2, 4, 6, 8, … 8186, 8188 or -2, -4, -6, -8, … -8186, -8188.

2. The oscillation frequency is the input clock (f_{RTC}).

It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUDW) is set to its initial value (0000H).

Note See **7.4.5 1 Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

(When the output frequency from the RTC1Hz pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F12 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when quickening), assume F12 to be 1.

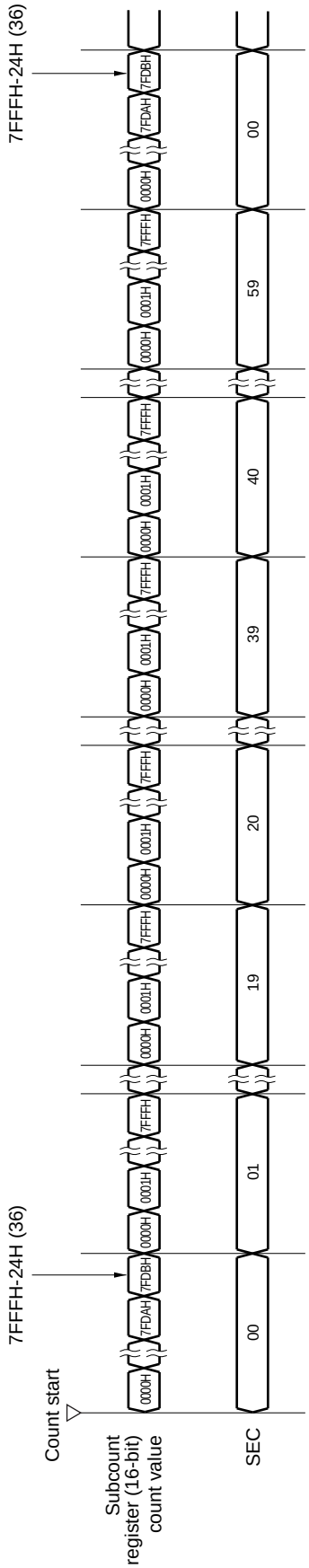
Calculate (F11 to F0) from the correction value.

− {(/F11 to /F0) − 1} × 2	= −36
(/F11 to /F0)	= 17
(/F11 to /F0)	= (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1)
(F11 to F0)	= (1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 12 to 0 of the SUBCUDW register: 1, 1, 1, 1, 1, 1, 1,0,1,1,1,0) results in 32768 Hz (0 ppm). <R>

Figure 7-26 shows the operation when (DEV, F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-26. Operation when (DEV, F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 8 INTERVAL TIMER

8.1 Functions of Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP Mode.

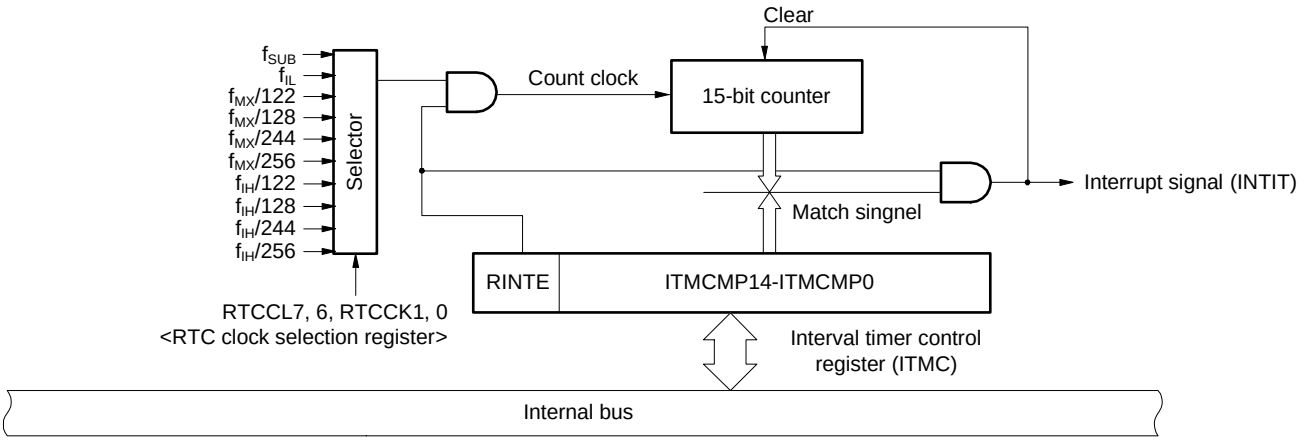
8.2 Configuration of Interval Timer

The interval timer includes the following hardware.

Table 8-1. Configuration of Interval Timer

Item	Configuration
Counter	15-bit counter
Control registers	Peripheral enable register 0 (PER0)
	RTC clock selection register (RTCCL)
	Interval timer control register (ITMC)

Figure 8-1. Block Diagram of Interval Timer



8.3 Registers Controlling Interval Timer

The interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- RTC clock selection register (RTCCL)
- Interval timer control register (ITMC)

(1) Peripheral enable register 0 (PER0)

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

When the interval timer is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 8-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

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

Note 1. The input clock that can be controlled by the RTCEN bit is used when the register that is used by the real-time clock (RTC) and interval timer are accessed from the CPU. The RTCEN bit cannot control supply of the operating clock (f_{SUB}) to RTC and interval timer.

Cautions 1. When using the interval timer, first set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCEN = 0, writing to a control register of the real-time clock or interval timer is ignored, and, even if the register is read, only the default value is read.

2. Clock supply to peripheral functions other than the real-time clock and interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the operation speed mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

(2) RTC clock selection register (RTCCL)

Figure 8-3. Format of RTC Clock Selection Register (RTCCL)

Address: F00F9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	0	0	0	0	RTCCKS1	RTCCKS0

RTCCL7	Operation clock source selection for RTC/interval timer
0	RTC/interval timer uses External Main clock (f_{MX})
1	RTC/interval timer uses High-speed on-chip oscillator clock (f_{IH})

RTCCKS1	RTCCKS0	RTCCL6	Operation selection of RTC macro and interval timer
0	0	0/1	Sub clock (f_{SUB})
0	1		Low-speed on-chip oscillator clock (f_{IL}, 15K @ typ) (WUTMMCK0 should be “1” to use this selection)
1	0	0	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7)/2^7
1	1	0	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7)/2^8
1	0	1	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7)/122
1	1	1	External Main or High-speed on-chip oscillator clock (after selected by RTCCL7)/244

Caution WUTMMCK0 should be set to “1” when f_{IL} is used for RTC/interval timer clock.

(3) Interval timer control register (ITMC)

This register is used to set up the starting and stopping of the interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 7FFFH.

Figure 8-4. Format of Interval Timer Control Register (ITMC)

Address: FFF50H After reset: 7FFFH R/W

Symbol	15	14 to 0
ITMC	RINTE	ITMCMP14 to ITMCMP0

RINTE	Interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITMCMP14 to ITMCMP0	Specification of the interval timer compare value
0001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITMCMP setting + 1)).
•	
•	
•	
7FFFH	

Example interrupt cycles when 0001H or 7FFFH is specified for ITMCMP14 to ITMCMP0

- ITMCMP14 to ITMCMP0 = 0001H, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] ≅ 61.03 [μs]
- ITMCMP14 to ITMCMP0 = 7FFFH, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (32767 + 1) = 1000 [ms]

Cautions 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

4. Only change the setting of the ITMCMP14 to ITMCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITMCMP14 to ITMCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

8.4 Interval Timer Operation

」

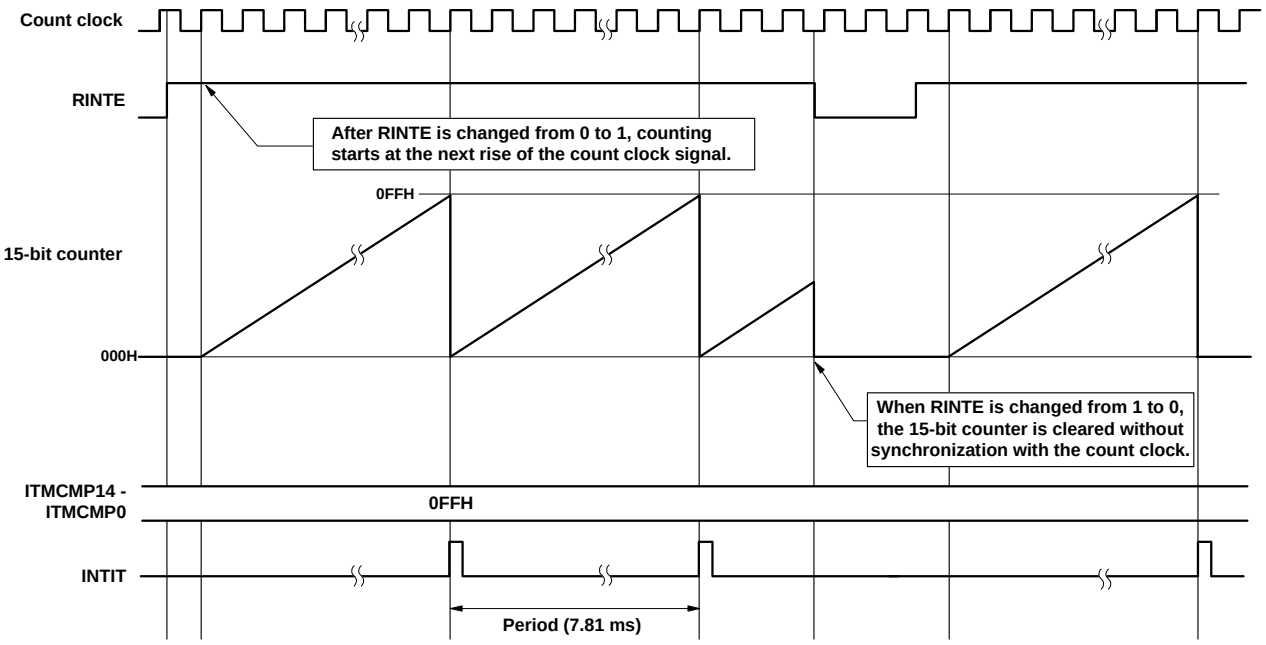
The count value specified for the ITMCMP14 to ITMCMP0 bits is used as an interval to operate an interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 15-bit counter starts counting.

When the 15-bit counter value matches the value specified for the ITMCMP14 to ITMCMP0 bits, the 15-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the interval timer is as follows.

Figure 8-5. Interval Timer Operation Timing (ITMCMP14 to ITMCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)



CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

9.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

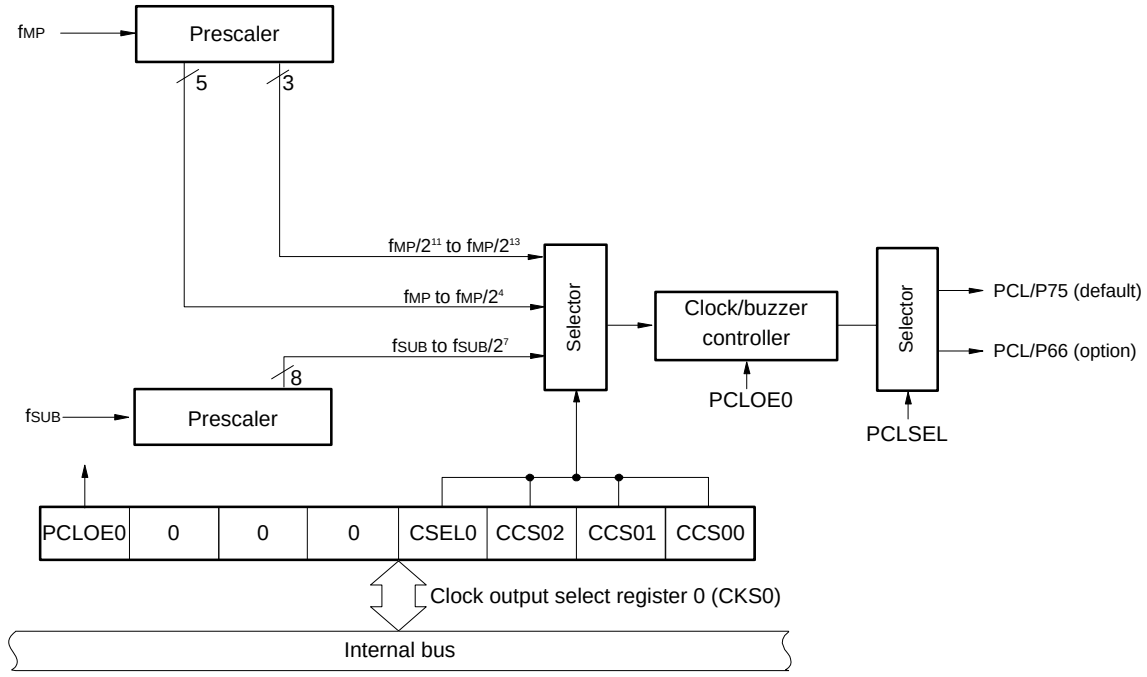
One pin can be used to output a clock or buzzer sound.

The PCL pin outputs a clock selected by clock output select register 0 (CKS0).

Figure 9-1 shows the block diagram of clock output/buzzer output controller.

Caution In the low-consumption RTC mode (when the RTCLPC bit of the operation speed mode control register (OSMC) = 1), it is not possible to output the subsystem clock (f_{SUB}) from the PCL pin.

Figure 9-1. Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCL, refer to **33.5 AC characteristics** and **34.5 AC characteristics**.

9.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 9-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register 0 (CKS0) Port mode register 7 (PM7)/6 (PM6) Port register 7 (P7)/6 (P6)

9.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output select registers 0 (CKS0)
- Port mode register 7 (PM7)/6 (PM6)

(1) Clock output select registers 0 (CKS0)

This register sets output enable/disable for clock output or for the buzzer frequency output pin (PCL), and set the output clock.

Select the clock to be output from the PCL pin by using the CKS0 register.

The CKS0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 9-2. Format of Clock Output Select Register 0 (CKS0)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	CSEL0	CCS02	CCS01	CCS00

PCLOE0	PCL pin output enable/disable specification
0	Output disable (default)
1	Output enable

CSEL0	CCS02	CCS01	CCS00	PCL pin output clock selection				
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 32 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz[Note]	Setting prohibited[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz[Note]	16 MHz[Note]
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.76 kHz	15.63 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	0	0	0	f_{SUB}	32.768 kHz			
1	0	0	1	$f_{SUB}/2$	16.384 kHz			
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz			
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz			
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz			
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz			
1	1	1	0	$f_{SUB}/2^6$	512 Hz			
1	1	1	1	$f_{SUB}/2^7$	256 Hz			

Note Use the output clock within a range of 16 MHz. Furthermore, the available output frequency depends on the grade. For details, refer to **33.5 AC Characteristics** or **34.5 AC Characteristics**.

Cautions 1. Change the output clock after disabling clock output (PCLOE0 = 0).

2. To shift to STOP mode when the main system clock is selected (CSEL0 = 0), set PCLOE0 = 0 before executing the STOP instruction. When the subsystem clock is selected (CSEL0 = 1), PCLOE0 = 1 can be set because the clock can be output in STOP mode.

3. In the low-consumption RTC mode (when the RTCLPC bit of the operation speed mode control register (OSMC) = 1), it is not possible to output the subsystem clock (f_{SUB}) from the PCL pin.

Remark f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

(2) Port mode register 7 (PM7)

This register sets input/output of port 7 in 1-bit units. Port 7 alternate function about a PCL output is shown in **Table 9-2**.

When using the P75/PCL pin for clock output and buzzer output, clear the PM75 bit and the output latches of P75 to 0.

The PM7 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 9-3. Format of Port Mode Register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	PM7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Table 9-2. Settings of Register, and Output Latch When Using Alternate Function

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P75	TI22	Input	1	x	0	TIS20.5,4 = 00	-
	TO22	Output	0	0	0	TOS20.5,4 = 00	SGSEL.3 = 1
	PCL	Output	0	0	0	SGSEL.3 = 0	TOS20.5,4 = 01/10
	SI01	Input	1	x	0	STSEL0.4 = 0	-
	SEG27	Output	x	x	1	-	-

(3) Port mode register 6 (PM6)

This register sets input/output of port 6 in 1-bit units. Port 6 alternate function about a PCL output is shown in **Table 9-3**.

When using the P66/PCL pin for clock output and buzzer output, clear the PM66 bit and the output latches of P66 to 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 9-4. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	PM6n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Table 9-3. Settings of Register, and Output Latch When Using Alternate Function

port	Alternate function		PMxx	Pxx	LCDPFxx	Expanded control setting (Register.bit)	
	Function name	I/O				Enable function	Disable other function
P66	TI24	Input	1	x	N/A	TIS21.1,0 = 00	-
	TO24	Output	0	0		TOS21.1,0 = 00	SGSEL.3 = 0
	PCL	Output	0	0		SGSEL.3 = 1	TOS21.1,0 = 01/10

(4) Sound generator and PCL pin select register (SGSEL)

This register is used for alternate switch of sound generator and PCL output pins.

SGOA output can be stopped when it is not used if SGSEL2 is set to “1”.

Figure 9-5. Format of sound generator and PCL pin select register (SGSEL)

Address: FFF3FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SGSEL	0	0	0	0	PCLSEL	SGSEL2	SGSEL1	SGSEL0

SGSEL2	SGSEL1	SGSEL0	Pin select of sound generator outputs	
			SGO/SGOF	SGOA
0	0	0	P73	P72
0	0	1	P93	P92
0	1	0	P135	P134
0	1	1	Setting prohibit	
1	0	0	P73	No port is selected
1	0	1	P93	(output disabled)
1	1	0	P135	
1	1	1	Setting prohibit	

Note The driving capability of SGO/SGOF alternate pin (P73, P93, P135) is larger than normal buffer.
P93 is also alternated as Stepper-Motor function, so its driving characteristics is the same as SM buffer.
P73 and P135 are the same as SG buffer of 78K0/Dx2.

PCLSEL	PCL output pin selection
0	P75 (default, be available for 48/64/80/100pin)
1	P66 (option for 80/100pin)

9.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

The PCL pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

9.4.1 Operation as output pin

The PCL pin is output as the following procedure.

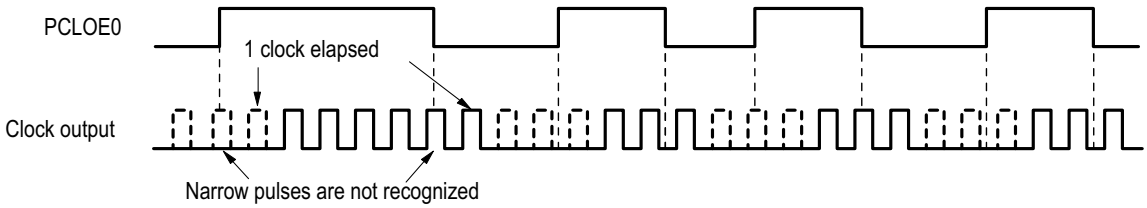
<0> Select the PCL output pin by the PCLSEL bit in the SGSEL register.

<1> Select the output frequency with bits 0 to 3 (CCS00 to CCS02, CSEL0) of the clock output select register 0 (CKS0) of the PCL pin (output in disabled status).

<2> Set bit 7 (PCLOE0) of the CKS0 register to 1 to enable clock/buzzer output.

Remarks The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOE0 bit) is switched. At this time, pulses with a narrow width are not output. Figure 9-6 shows enabling or stopping output using the PCLOE0 bit and the timing of outputting the clock.

Figure 9-6. Remote Control Output Application Example



9.5 Cautions of clock output/buzzer output controller

When the main system clock is selected for the PCL output, if STOP or HALT mode is entered within 1.5 main system clock cycles after the output is disabled (PCLOE0 = 0), the PCL output width becomes shorter.

CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The watchdog timer operates on the low-speed on-chip oscillator clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 23 RESET FUNCTION**.

When 75% + 1/2 f_{IL} of the overflow time is reached, an interval interrupt can be generated.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

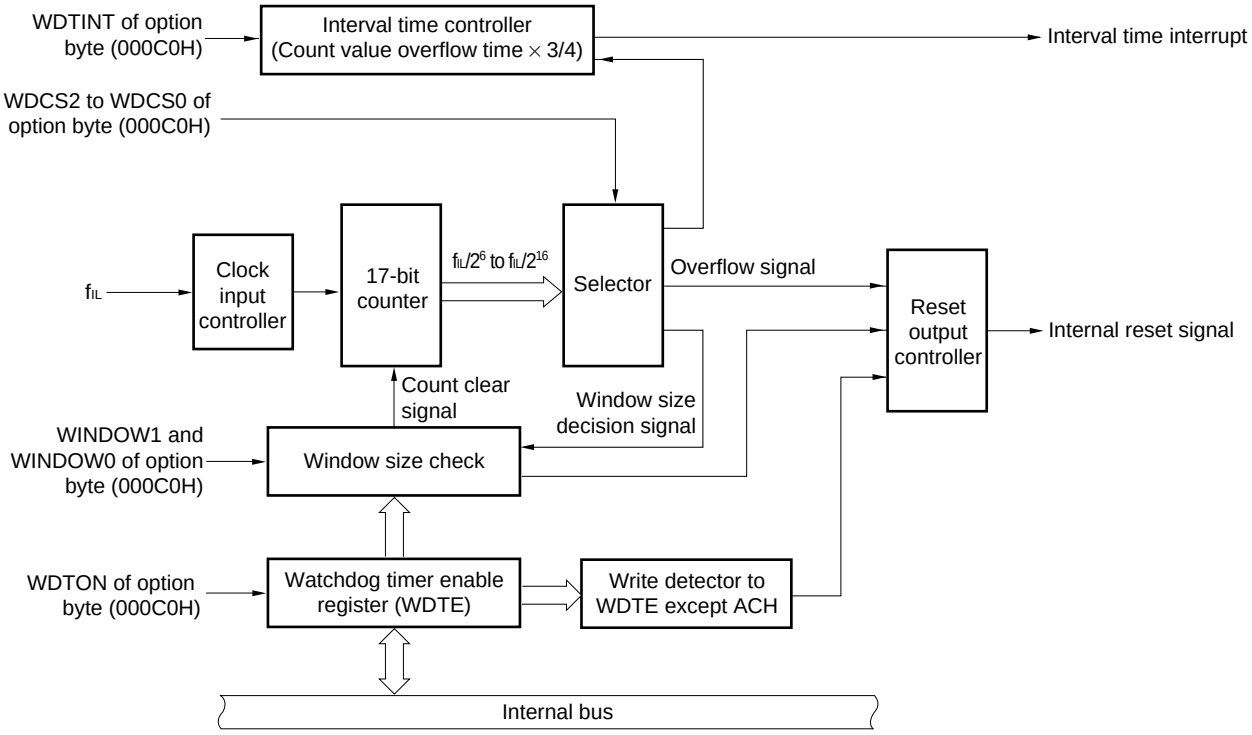
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 10-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 28 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



10.3 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 the WDTE register clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFABH After reset: 9AH/1AH [Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 28**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **10.4.2** and **CHAPTER 28**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **10.4.3** and **CHAPTER 28**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the WDTE register
   - If data other than “ACH” is written to the WDTE register

Cautions 1. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing “ACH” to the WDTE register, the actual overflow time may be different from the overflow time set by the option byte by up to 2/f_{IL} seconds.

3. The watchdog timer can be cleared immediately before the count value overflows.

Cautions 4. The operation of the watchdog timer in the HALT and STOP and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

5. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM™ emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

10.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

Table 10-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19 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.

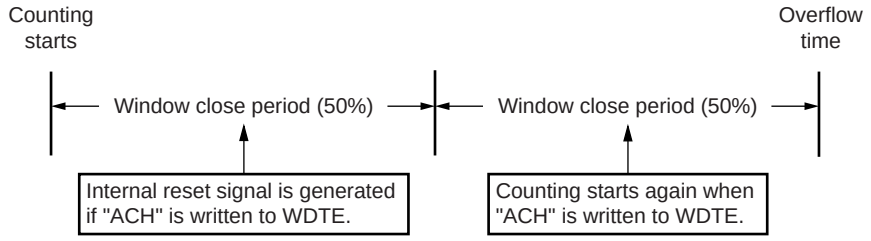
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

10.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

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 the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 − 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 − 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz (MIN.) × 0.5 to 2^9/17.25 kHz (MAX.) = 20.08 to 29.68 ms

10.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% + 1/2 f_{IL} of the overflow time is reached.

Table 10-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of overflow time is reached.

Caution When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 11 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

<R>

	48-pin	64-pin	80-pin	100-pin	128-pin
Analog input channels	3+2 ch	3+2 ch	6+2 ch	7+2 ch	9+2 ch

Caution Most of the following descriptions in this chapter use the 128-pin products as an examples.

11.1 Function of A/D Converter

The A/D converter is a 10-bit resolution[Note] converter that converts analog input signals into digital values, and is configured to control analog inputs, including up to nine channels of A/D converter analog inputs (ANI0 to ANI8).

The A/D converter has the following function.

- **10-bit resolution A/D conversion[Note]**
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI8. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

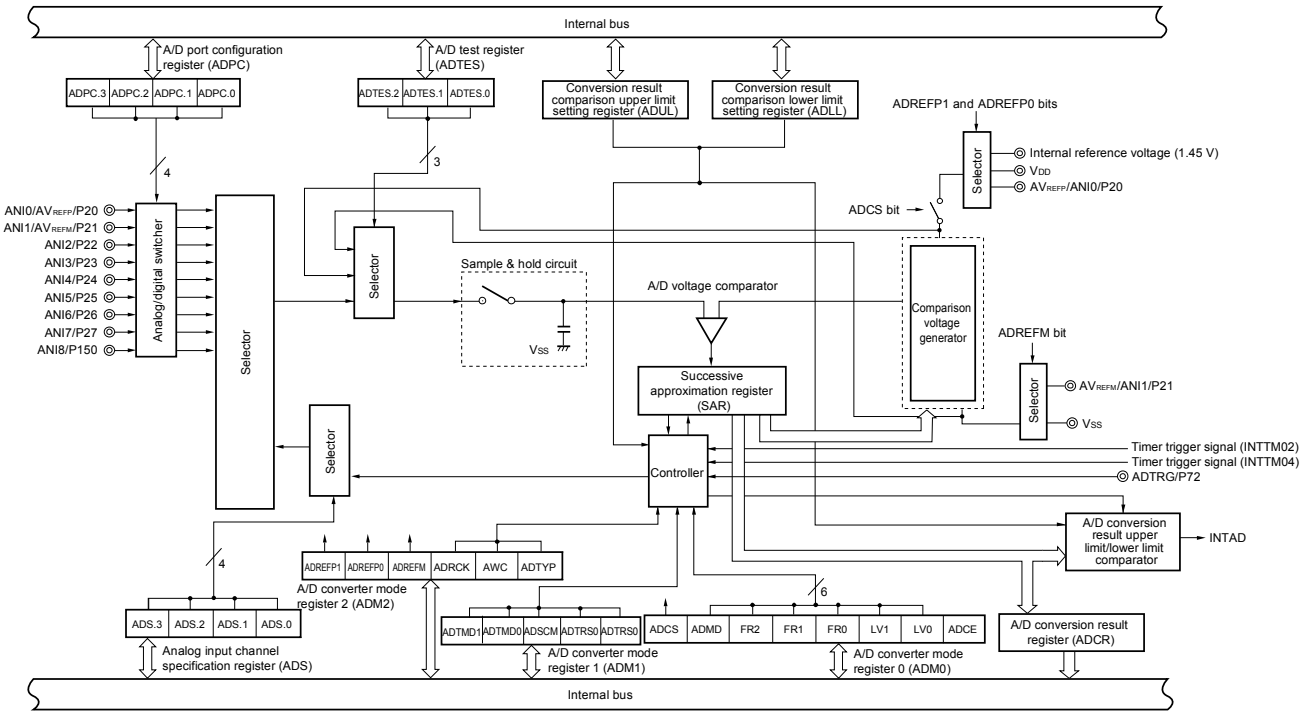
Note 8-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger Mode	Channel Selection Mode	Conversion Operation Mode
• Software trigger Conversion is started by specifying a software trigger. • Hardware trigger no-wait mode Conversion is started by detecting a hardware trigger. • Hardware trigger wait mode The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes.	• Select mode A/D conversion is performed on the analog input of one channel. • Scan mode A/D conversion is performed on the analog input of four channels in order.	• One-shot conversion mode A/D conversion is performed on the selected channel once. • Sequential conversion mode A/D conversion is sequentially performed on the selected channels until it is stopped by software.

<R>

Figure 11-1. Block Diagram of A/D Converter



Remark The analog input pins in the figure are provided in the 128-pin products.

11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI8 pins

These are the analog input pins of the up to 9 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a 10-bit register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{REFP} pin

This pin inputs an external reference voltage (AV_{REFP}).

If using AV_{REFP} as the + side reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively.

The analog signals input to ANI0 to ANI8 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). If using AV_{REFM} as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the – side reference voltage of the A/D converter.

Caution The A/D conversion accuracy differs depending on the used pins or reference voltage setting. For details, see CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT) and CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT).

11.3 Registers Used in A/D Converter

The A/D converter uses the following registers.

- Peripheral enable register 1 (PER1)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- Port mode registers 0 to 9, and 13 to 15 (PM0 to PM9, PM13 to PM15)

(1) Peripheral enable register 1 (PER1)

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

When the A/D converter is used, be sure to set bit 7 (ADCEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00F1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	ADCEN	0	MTRCEN	SGEN	0	0	0	0

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

Cautions When setting the A/D converter, be sure to set the ADCEN bit 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 2 and 15 (PM2, PM15) and A/D port configuration register (ADPC)).

(2) A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-3. Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	DV1 Note 1	DV0 Note 1	ADCE

ADTMD1	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When read Note 2] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: Stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 3
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 11-3 A/D Conversion Time Selection**.

2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Cautions 1. Change the bits ADMD, FR2 to FR0, LV1 and LV0, and ADCE in the conversion stop state or conversion wait state (ADCS = 0).

2. It is prohibited to change the ADCE and ADCS bits from 0 to 1 by an 8-bit manipulation instruction. To change these bits, use the procedure described in 11.7, A/D Converter Setup Flowchart.

Table 11-1. Settings of ADCS and ADCE Bits

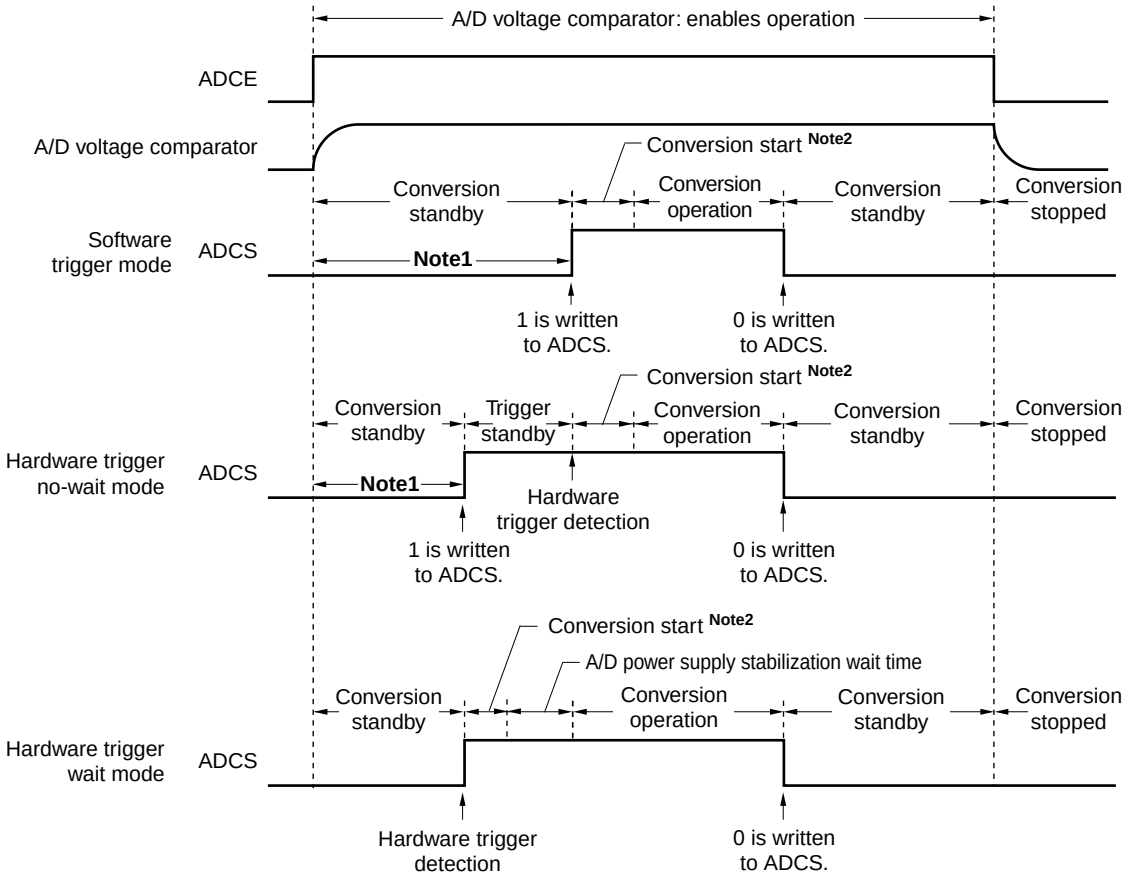
ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion standby mode (only A/D voltage comparator consumes power[Note])
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator: enables operation)

Note In hardware trigger wait mode, the DC power consumption path is not provided even in conversion wait mode.

Table 11-2. Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 11-4. Timing Chart When A/D Voltage Comparator Is Used



Notes 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

2. In starting conversion, the longer will take up to following time.

ADM0			Conversion clock (f_{AD})	Conversion Operation Time (f_{CLK} clock)	
FR2	FR1	FR0		Software trigger mode / Hardware trigger no-wait mode	Hardware trigger wait mode
0	0	0	f_{CLK}/64	63	1
0	0	1	f_{CLK}/32	31	
0	1	0	f_{CLK}/16	15	
0	1	1	f_{CLK}/8	7	
1	0	0	f_{CLK}/6	5	
1	0	1	f_{CLK}/5	4	
1	1	0	f_{CLK}/4	3	
1	1	1	f_{CLK}/2	1	

In the conversion after the second conversion in continuous conversion mode or after the scan 1 in scan mode, the conversion startup time or A/D power supply stabilization wait time is not generated after detection of a hardware trigger.

Cautions 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Cautions 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

4. To complete A/D conversion, the following hardware trigger interval time is required:
In hardware trigger no-wait mode: Two f_{CLK} clock cycles + A/D conversion time
In hardware trigger wait mode: Two f_{CLK} clock cycles + stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (1/6)

(1) 4.0 V ≤ V_{DD} ≤ 5.5 V

When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32					38 μs	19 μs
0	1	0				f_{CLK}/16				38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8			38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6			28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5			23.75 μs	11.875 μs	5.938 μs	2.9688 μs
1	1	0				f_{CLK}/4		38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				f_{CLK}/2	38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32					34 μs	17 μs
0	1	0				f_{CLK}/16				34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8			34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6			25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5			21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs
1	1	1				f_{CLK}/2	34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited
Other than the above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (2/6)

(2) 2.7 V ≤ V_{DD} < 5.5 V

When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32					38 μs	19 μs
0	1	0				f_{CLK}/16				38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8			38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6			28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5			23.75 μs	11.875 μs	5.9375 μs	Setting prohibited
1	1	0				f_{CLK}/4		38 μs	19 μs	9.5 μs	4.75 μs	
1	1	1				f_{CLK}/2	38 μs	19 μs	9.5 μs	4.75 μs	Setting prohibited	
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32					34 μs	17 μs
0	1	0				f_{CLK}/16				34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8			34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6			25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5			21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		34 μs	17 μs	8.5 μs	4.25 μs	Setting prohibited
1	1	1				f_{CLK}/2	34 μs	17 μs	8.5 μs	4.25 μs	Setting prohibited	
Other than the above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (3/6)

(3) 1.8 V ≤ V_{DD} < 5.5 V

When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32					38 μs	19 μs
0	1	0				f_{CLK}/16				38 μs	19 μs	Setting prohibited
0	1	1				f_{CLK}/8			38 μs	19 μs	Setting prohibited	
1	0	0				f_{CLK}/6			28.5 μs	Setting prohibited		
1	0	1				f_{CLK}/5			23.75 μs			
1	1	0				f_{CLK}/4		38 μs	19 μs			
1	1	1				f_{CLK}/2	38 μs	19 μs	Setting prohibited			
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32					34 μs	17 μs
0	1	0				f_{CLK}/16				34 μs	17 μs	Setting prohibited
0	1	1				f_{CLK}/8			34 μs	17 μs	Setting prohibited	
1	0	0				f_{CLK}/6			25.5 μs	Setting prohibited		
1	0	1				f_{CLK}/5			21.25 μs			
1	1	0				f_{CLK}/4		34 μs	17 μs			
1	1	1				f_{CLK}/2	34 μs	17 μs	Setting prohibited			
Other than the above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (4/6)

(4) 4.0 V ≤ V_{DD} ≤ 5.5 V

When there is stabilization wait time (hardware trigger wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						27 μs
0	1	0				f_{CLK}/16					27 μs	13.5 μs
0	1	1				f_{CLK}/8				27 μs	13.5 μs	6.75 μs
1	0	0				f_{CLK}/6				20.25 μs	10.125 μs	5.0625 μs
1	0	1				f_{CLK}/5			33.75 μs	16.875 μs	8.4375 μs	4.2188 μs
1	1	0				f_{CLK}/4			27 μs	13.5 μs	6.75 μs	3.375 μs
1	1	1				f_{CLK}/2		27 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						25 μs
0	1	0				f_{CLK}/16					25 μs	12.5 μs
0	1	1				f_{CLK}/8				25 μs	12.5 μs	6.25 μs
1	0	0				f_{CLK}/6			37.5 μs	18.75 μs	9.375 μs	4.6875 μs
1	0	1				f_{CLK}/5			31.25 μs	15.625 μs	7.8125 μs	3.9063 μs
1	1	0				f_{CLK}/4			25 μs	12.5 μs	6.25 μs	3.125 μs
1	1	1				f_{CLK}/2		25 μs	12.5 μs	6.25 μs	3.125 μs	Setting prohibited
Other than the above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (5/6)

(5) 2.7 V ≤ V_{DD} < 5.5 V

When there is stabilization wait time (hardware trigger wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						27 μs
0	1	0				f_{CLK}/16					27 μs	13.5 μs
0	1	1				f_{CLK}/8				27 μs	13.5 μs	6.75 μs
1	0	0				f_{CLK}/6				20.25 μs	10.125 μs	5.0625 μs
1	0	1				f_{CLK}/5			33.75 μs	16.875 μs	8.4375 μs	4.2188 μs
1	1	0				f_{CLK}/4			27 μs	13.5 μs	6.75 μs	3.375 μs
1	1	1				f_{CLK}/2		27 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						25 μs
0	1	0				f_{CLK}/16					25 μs	12.5 μs
0	1	1				f_{CLK}/8				25 μs	12.5 μs	6.25 μs
1	0	0				f_{CLK}/6			37.5 μs	18.75 μs	9.375 μs	4.6875 μs
1	0	1				f_{CLK}/5			31.25 μs	15.625 μs	7.8125 μs	3.9063 μs
1	1	0				f_{CLK}/4			25 μs	12.5 μs	6.25 μs	Setting prohibited
1	1	1				f_{CLK}/2		25 μs	12.5 μs	6.25 μs	Setting prohibited	
Other than the above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (6/6)

(6) 1.8 V ≤ V_{DD} < 5.5 V

When there is stabilization wait time (hardware trigger wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						27 μs
0	1	0				f_{CLK}/16					27 μs	Setting prohibited
0	1	1				f_{CLK}/8				27 μs	Setting prohibited	
1	0	0				f_{CLK}/6				20.25 μs		
1	0	1				f_{CLK}/5			33.75 μs	Setting prohibited		
1	1	0				f_{CLK}/4			27 μs			
1	1	1				f_{CLK}/2		27 μs	Setting prohibited			
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						25 μs
0	1	0				f_{CLK}/16					25 μs	Setting prohibited
0	1	1				f_{CLK}/8				25 μs	Setting prohibited	
1	0	0				f_{CLK}/6			37.5 μs	18.75 μs		
1	0	1				f_{CLK}/5			31.25 μs	Setting prohibited		
1	1	0				f_{CLK}/4			25 μs			
1	1	1				f_{CLK}/2		25 μs	Setting prohibited			
Other than the above						Setting prohibited						

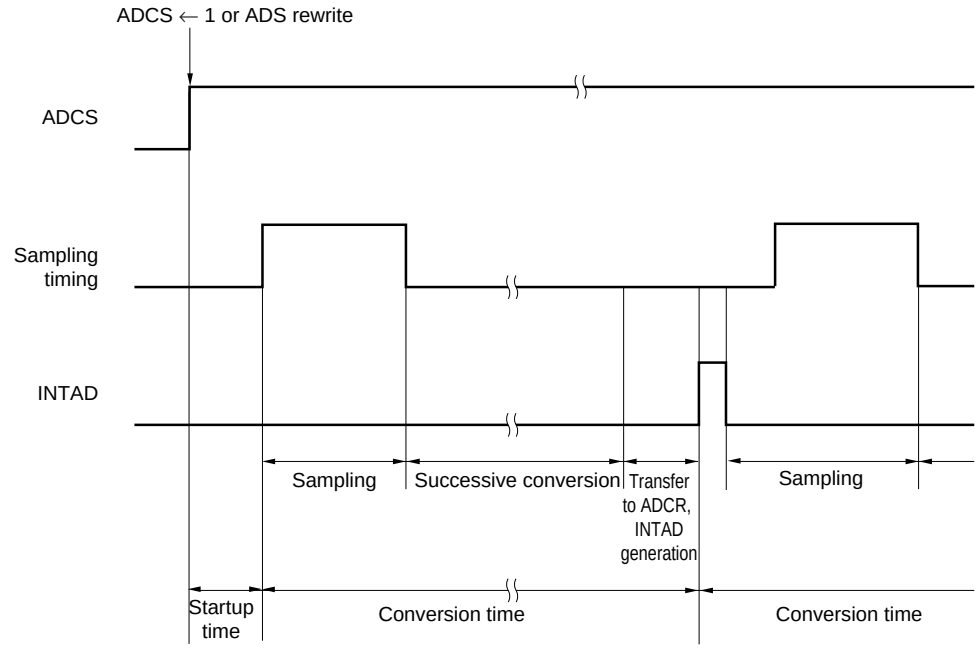
Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 11-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



(3) A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.
Hardware trigger mode with ADTRG is added to be supported. ADTRG is allocated to P72 as alternate function.
The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 11-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	x	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of TAU 02 count or capture interrupt (INTTM02)
0	1	End of TAU 04 count or capture interrupt (INTTM04)
1	0	Hardware trigger from external pin (ADTRG) without noise filter
1	1	Hardware trigger from external pin (ADTRG) with noise filter

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

Cautions 1. Only rewrite the value of the ADM1 register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

2. To complete A/D conversion, the following hardware trigger interval time is required:
In hardware trigger no-wait mode: Two fCLK clock cycles + A/D conversion time
In hardware trigger wait mode: Two fCLK clock cycles + stabilization wait time + A/D conversion time

Remarks 1. ×: don't care

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

(4) A/D converter mode register 2 (ADM2)

This register is used to select the A/D converter reference voltage, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0 (recommended setting)
1	0	Supplied from the internal reference voltage (1.45 V)Note
1	1	Setting prohibited
Rewrite the values of the ADREFP1 and ADREFP0 bits in the following procedure: 1. Set ADCE to 0. 2. Change ADREFP1 and ADREFP0. 3. Count the stabilization wait time (A). 4. Set ADCE to 1. 5. Count the stabilization wait time (B). To set ADREFP1 to 1 and ADREFP0 to 0: A = 5 μs, B = 1 μs To set ADREFP1 to 0 and ADREFP0 to 0, or ADREFP1 to 0 and ADREFP0 to 1: A = no wait, B = 1 μs After step 5, start A/D conversion.		

ADREFM	Selection of the − side reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1 (recommended setting)

Note Can only be selected in HS (high-speed main) mode.

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (<1>).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (<2>) or the ADUL register < the ADCR register (<3>).
Figure 11-8 shows the generation range of the interrupt signal (INTAD) for <1> to <3>.	

Cautions 1. Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock.

3. To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog inputs and port mode register to input mode.

Figure 11-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

AWC	Specification of SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

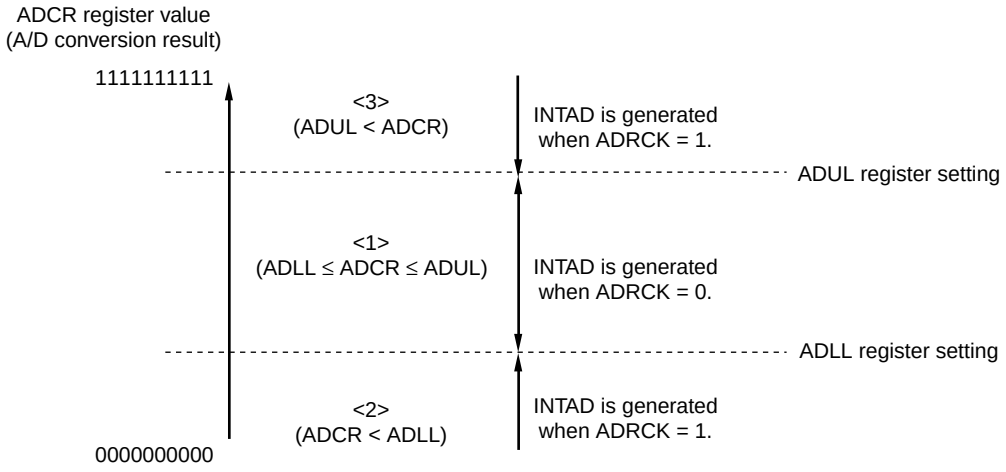
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “transition time to SNOOZE mode[Note] + A/D power supply stabilization wait time + A/D conversion time + two f_{CLK} clock cycles”.
- When using the SNOOZE function in normal operation mode, set AWC to 0, and then change it to 1 immediately before a transition to STOP mode.
  Be sure to change AWC to 0 after returning from STOP mode to normal operation mode.
  If AWC remains 1, A/D conversion is not correctly started regardless whether the subsequent mode is SNOOZE mode or normal operation mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note See the descriptions of “From STOP to SNOOZE” in **22.2.3, SNOOZE mode**.

Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

Figure 11-8. ADRCK Bit Interrupt Signal Generation Range



Remark: If INTAD is not generated, the A/D conversion results are not stored in the ADCR or ADCRH register.

(5) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH. Note

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register (see figure 11-8)), the value is not stored.

Figure 11-9. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCR register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

2. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (ADCR1 and ADCR0).

3. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

(6) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Note If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register (see figure 11-8)), the value is not stored.

Figure 11-10. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

(7) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 11-11. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
ADS	0	0	0	0	ADS.3	ADS.2	ADS.1	ADS.0

○ Select mode (ADMD = 0)

ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	1	0	ANI2	P22/ANI2 pin
0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	ANI4	P24/ANI4 pin
0	1	0	1	ANI5	P25/ANI5 pin
0	1	1	0	ANI6	P26/ANI6 pin
0	1	1	1	ANI7	P27/ANI7 pin
1	0	0	0	ANI8	P150/ANI8 pin
<R> 1	0	0	1	ANI9	P151/ANI9 pin
<R> 1	0	1	0	ANI10	P152/ANI10 pin
Other than the above				Setting prohibited	

○ Scan mode (ADMD = 1)

ADS3	ADS2	ADS1	ADS0	Analog input channel			
				Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	1	0	1	ANI5	ANI6	ANI7	ANI8
<R> 0	1	1	0	ANI6	ANI7	ANI8	ANI9
<R> 0	1	1	1	ANI7	ANI8	ANI9	ANI10
Other than the above				Setting prohibited			

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

Cautions 1. Be sure to clear bits 4, 5, 6, and 7 to 0.

2. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2, 15 (PM2, PM15).

3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

4. If using AV_{REFP} as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

5. If using AV_{REFM} as the – side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

6. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock.

7. The corresponding ANI pin does not exist depending on the product. In this case, ignore the conversion result.

(8) Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 11-8**).

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

Reset signal generation sets this register to FFH.

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Figure 11-12. Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL.7	ADUL.6	ADUL.5	ADUL.4	ADUL.3	ADUL.2	ADUL.1	ADUL.0

(9) Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 11-8**).

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

Reset signal generation clears this register to 00H.

Figure 11-13. Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL.7	ADLL.6	ADLL.5	ADLL.4	ADLL.3	0	ADLL.1	ADLL.0

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADLL register.

(10) A/D test register (ADTES)

This register is used to select the + side reference voltage (AV_{REFP}) or - side reference voltage (AV_{REFM}) of the A/D converter, or the analog input channel (ANIxx) as the A/D conversion target for the A/D test function.

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

Reset signal generation clears this register to 00H.

Figure 11-14. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	ADTES2	ADTES1	ADTES0

ADTES2	ADTES1	ADTES0	A/D conversion target
0	0	0	ANIxx (This is specified using the analog input channel specification register (ADS).)
0	1	0	AV_{REFM}
0	1	1	AV_{REFP}
Other than the above			Setting prohibited

<R> **(11) A/D port configuration register (ADPC)**

This register switches the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI10/P152 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation clears this register to 00H.

Figure 11-15. Format of A/D Port Configuration Register (ADPC)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	ADPC.3	ADPC.2	ADPC.1	ADPC.0

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching										
				ANI10/P152	ANI9/P151	ANI8/P150	ANI7/P27	ANI6/P26	ANI5/P25	ANI4/P24	ANI3/P23	ANI2/P22	ANI1/AV_{REFM}/P21	ANI0/AV_{REFP}/P20
0	0	0	0	A	A	A	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	D	D	D	A	A	A	A	A
0	1	1	1	D	D	D	D	D	A	A	A	A	A	A
1	0	0	0	D	D	D	D	A	A	A	A	A	A	A
1	0	0	1	D	D	D	A	A	A	A	A	A	A	A
1	0	1	0	D	D	A	A	A	A	A	A	A	A	A
1	0	1	1	D	A	A	A	A	A	A	A	A	A	A
Other than the above				Setting prohibited										

Cautions 1. Set the port to analog input by ADPC register to the input mode by using port mode registers 2, 15 (PM2, PM15).

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

3. To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog inputs and port mode register to input mode.

<R> **(12) Port mode registers (PM0 to PM15)**

When using the ANI0 to ANI8 pin for an analog input port, set the PMmn bit to 1. The output latches of Pnm at this time may be 0 or 1.

If the PMmn bits are set to 0, they cannot be used as analog input port pins.

The PMmn registers 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.

Remark m = 0 to 15, n = 0 to 7

Figure 11-16. Format of Port Mode Register (128-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25	FFH	R/W
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29	FFH	R/W
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100	FFF2A	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFF2B	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	1	FFF2C	FFH	R/W
PM13	1	PM136	PM135	PM134	PM133	PM132	PM131	0	FFF2D	FEH	R/W
PM14	1	1	1	1	1	1	1	PM140	FFF2E	FFH	R/W
PM15	1	1	1	1	1	PM152	PM151	PM150	FFF2F	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 15 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog inputs and port mode register to input mode.

Remark For details of the port mode register other than 100-pin products, see 4. 3 Registers Controlling Port Function.

The ANI0/P20 to ANI7/P27 pins are as shown below depending on the settings of the A/D port configuration register (ADPC), analog input channel specification register (ADS), and PM2 registers.

Table 11-4. Setting Functions of ANI0/P20 to ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 to ANI7/P27 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.	

The ANI8/P150 pin is as shown below depending on the settings of the A/D port configuration register (ADPC), analog input channel specification register (ADS), and PM15 registers.

Table 11-5. Setting Functions of ANI8/P150 Pins

ADPC	PM15	ADS	ANI8/P150 to ANI10/P152 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.	

11.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched[Note1].
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated[Note1].

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0[Note2].
To stop the A/D converter, clear the ADCS bit to 0.

Notes 1. If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated. In this case, the result value is not stored in the ADCR or ADCRH register.

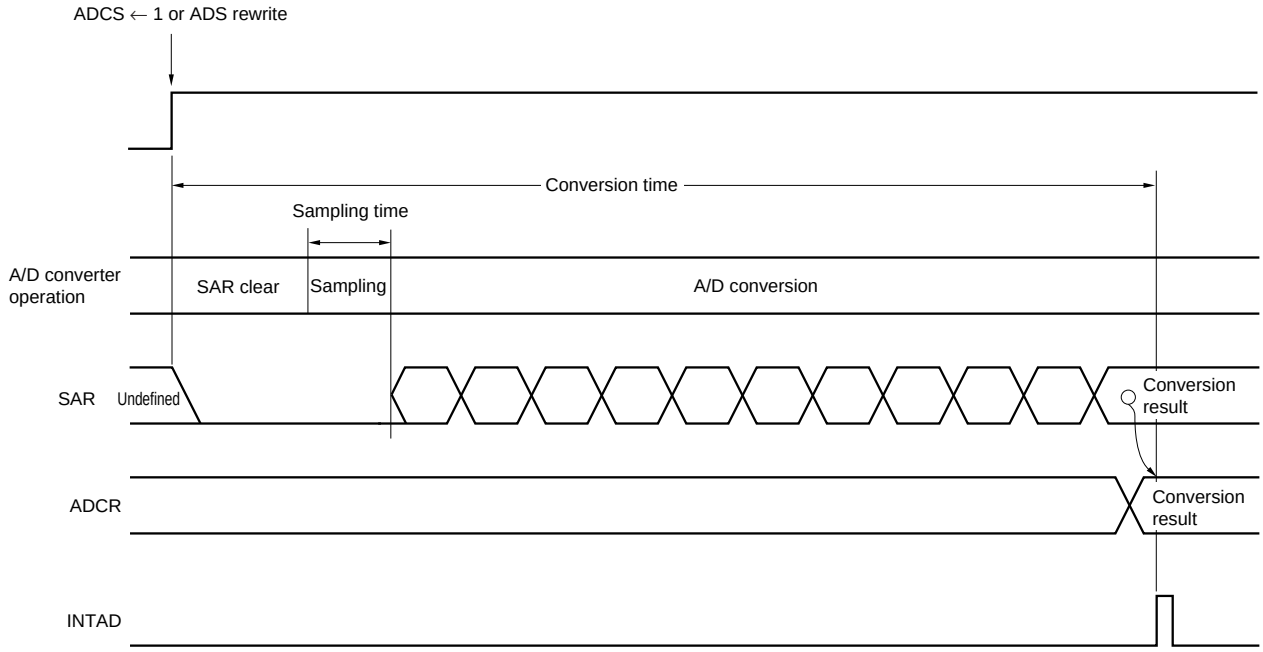
2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remarks 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Figure 11-17. Conversion Operation of A/D Converter (Software Trigger Mode)



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) 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 register (ADCR, ADCRH) to 0000H or 00H.

11.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI8) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

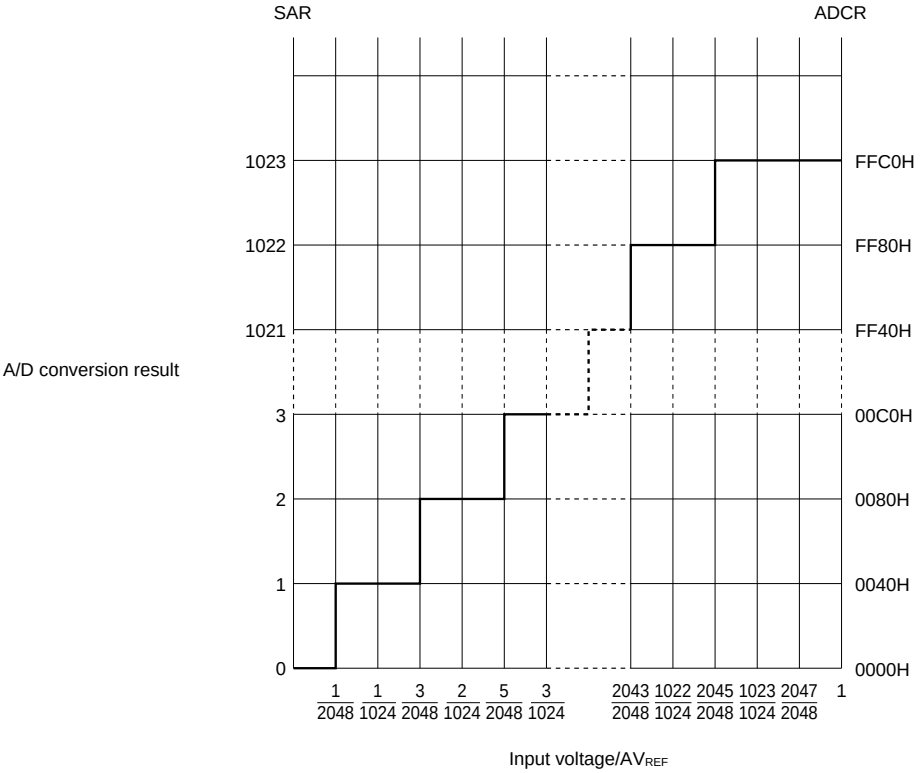
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 11-18 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-18. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

11.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **11.7 A/D Converter Setup Flowchart**.

11.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

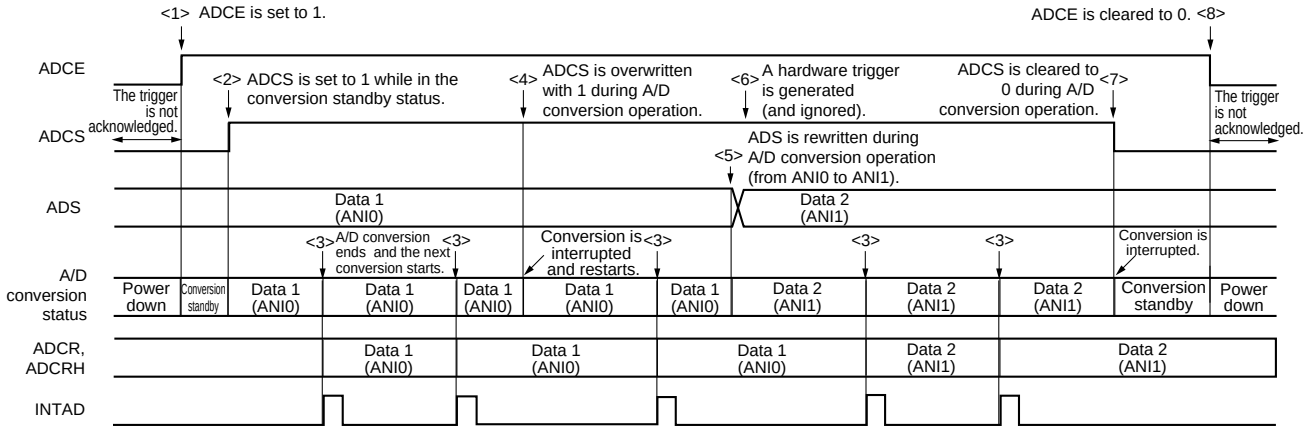
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-19. Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



11.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

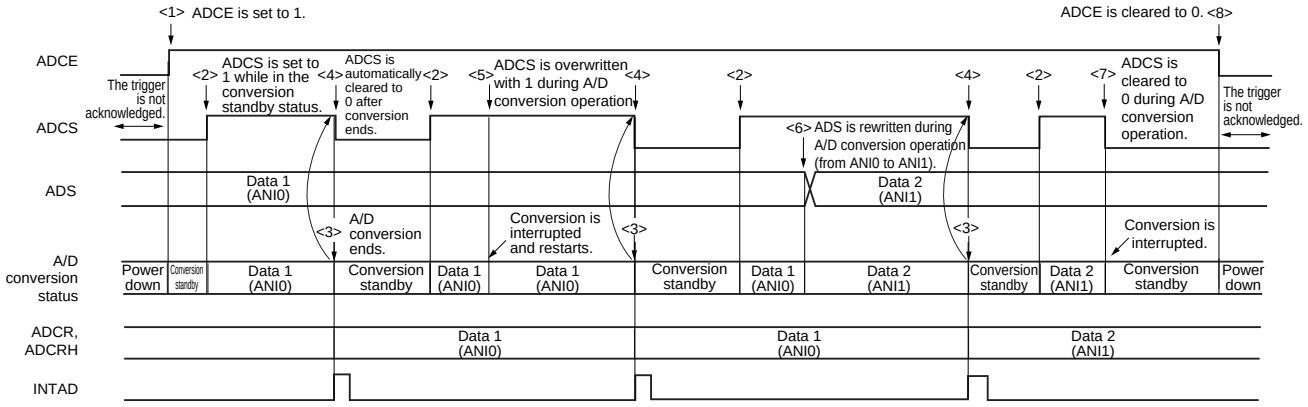
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 11-20. Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



11.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

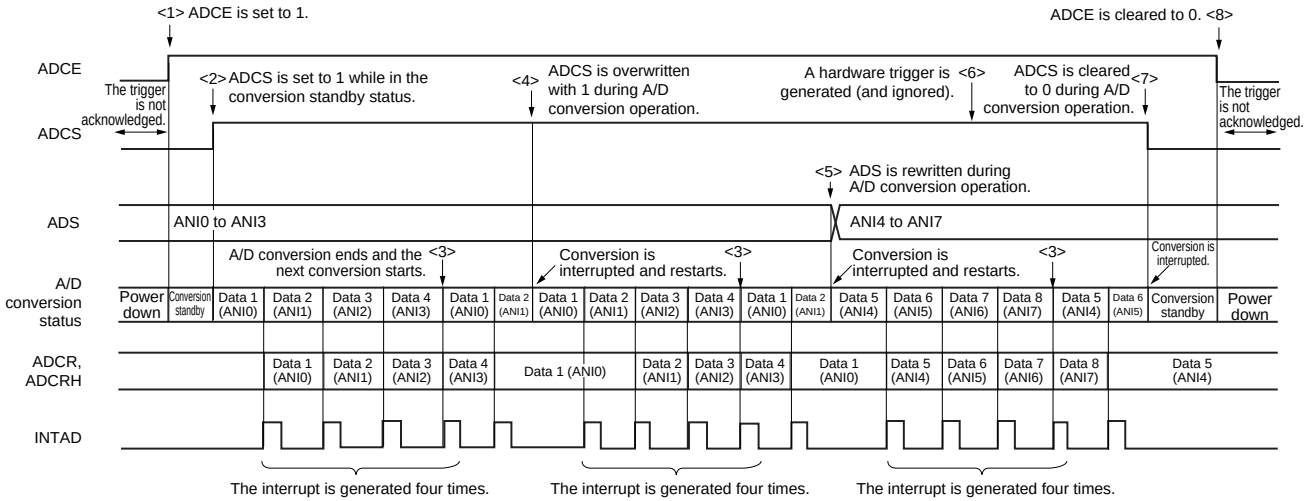
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-21. Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



11.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

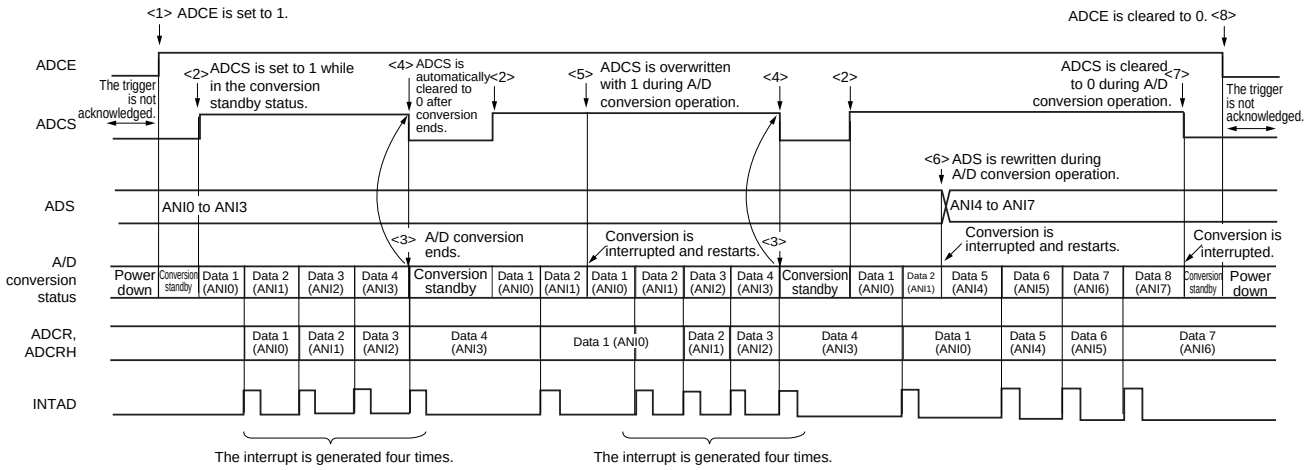
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 11-22. Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



11.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

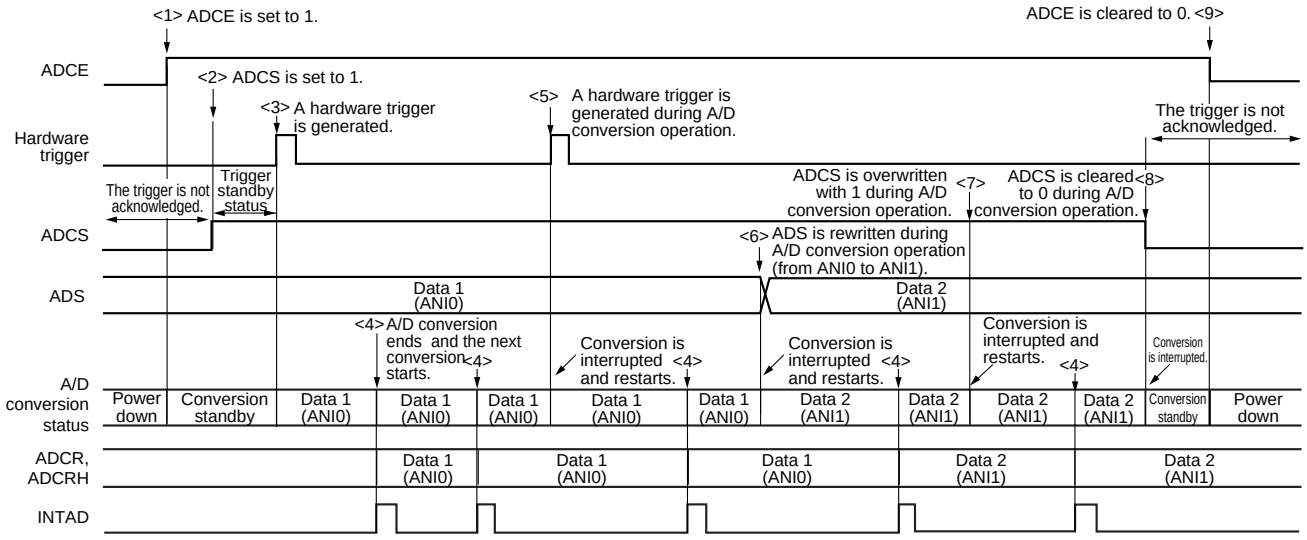
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-23. Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



11.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

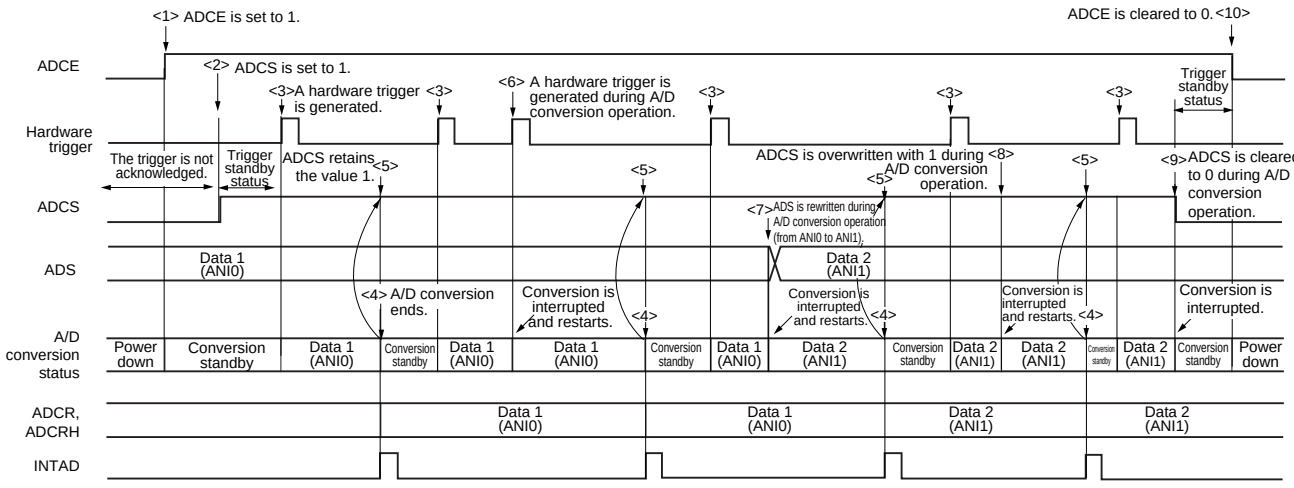
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-24. Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



11.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

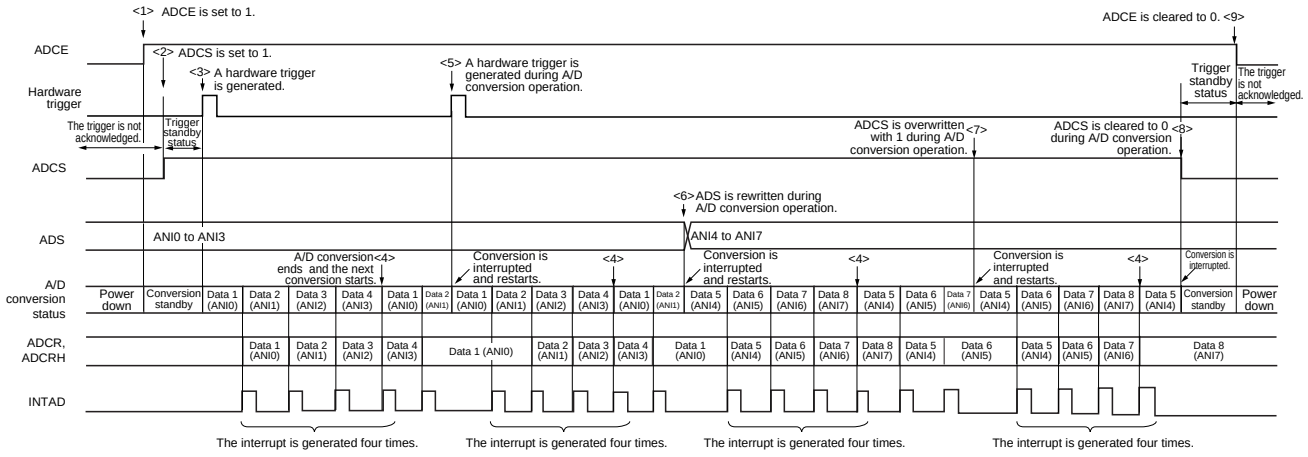
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-25. Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



11.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

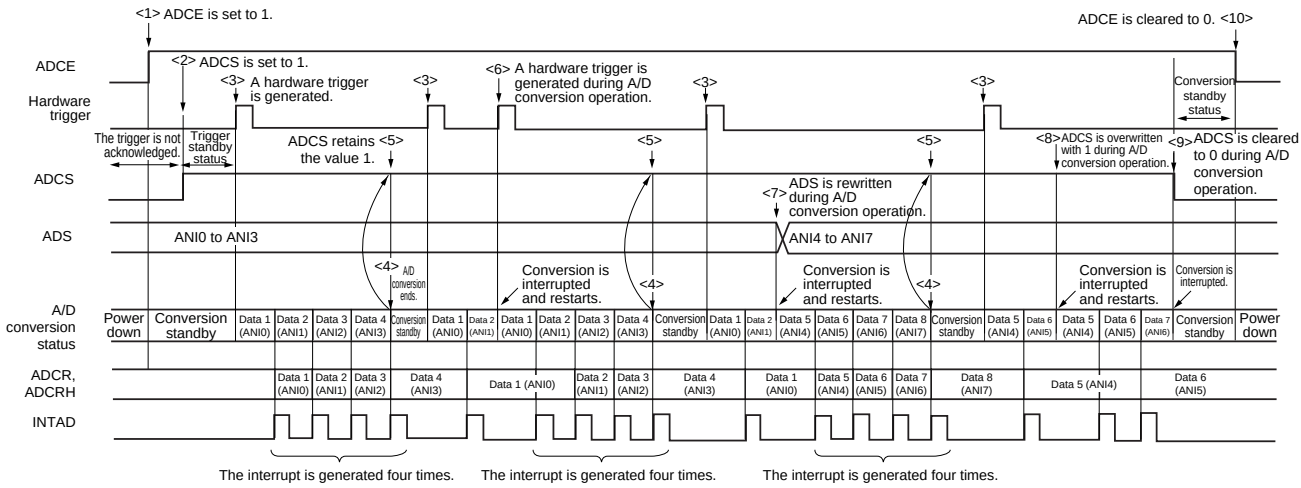
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-26. Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



11.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

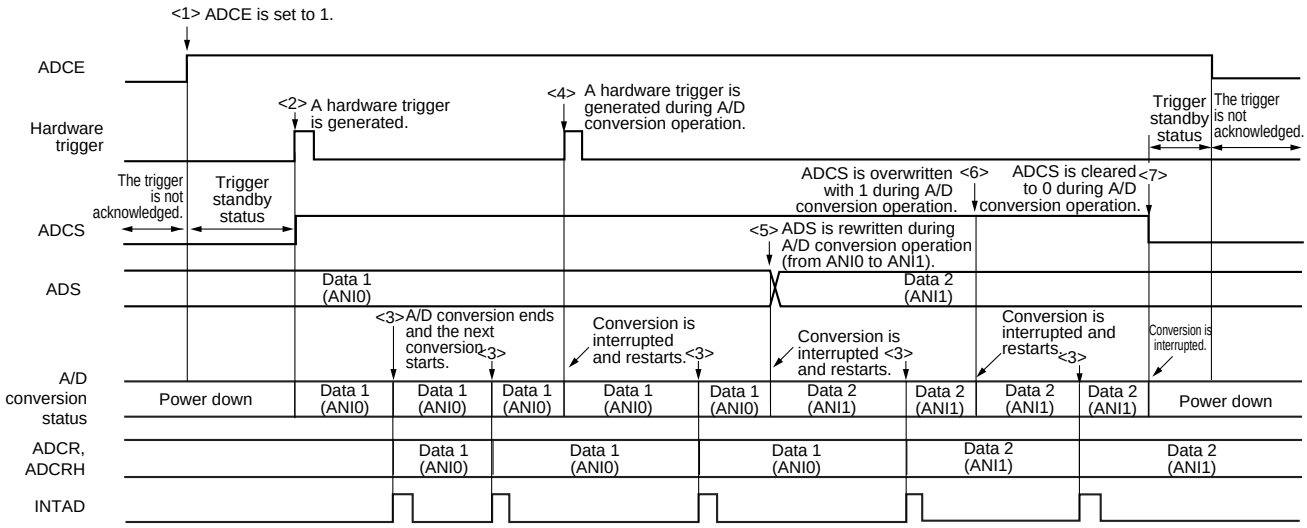
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-27. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



11.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the power-down status.

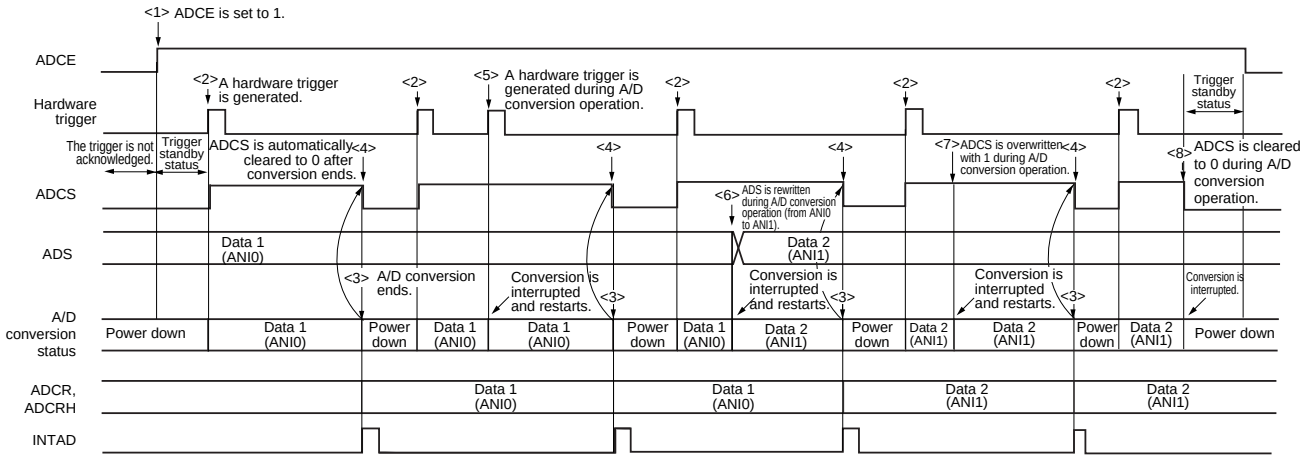
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-28. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



11.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

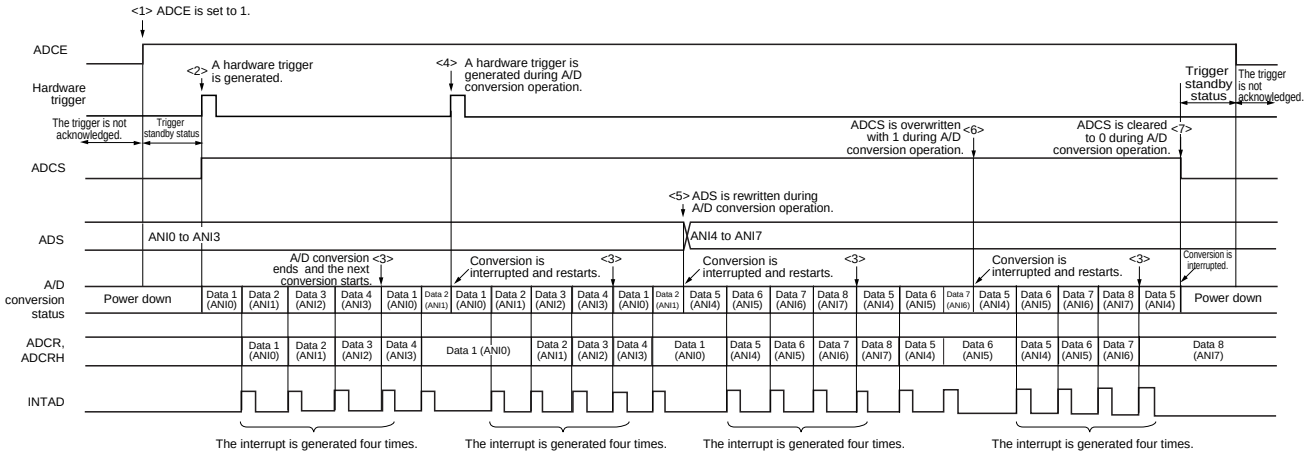
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-29. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



11.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the power-down status.

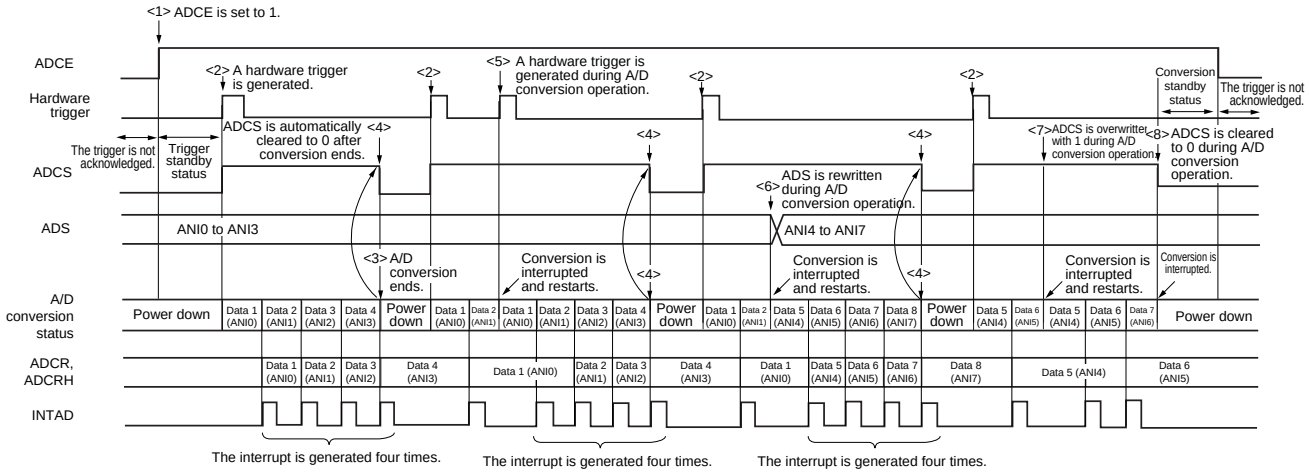
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-30. Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

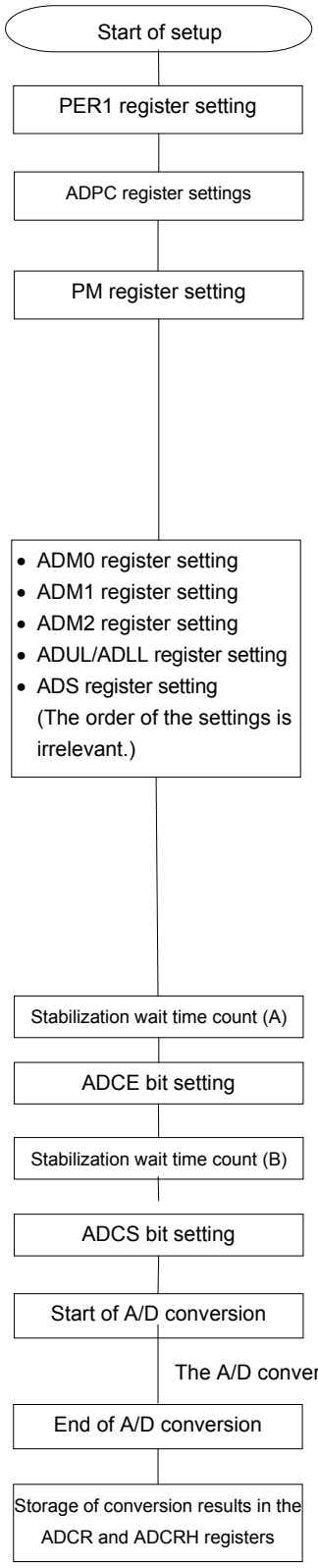


11.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

11.7.1 Setting up software trigger mode

Figure 11-31. Setting up Software Trigger Mode



The ADCEN bit of the PER1 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI8 pins: Set using the ADPC register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS.3 to ADS.0 bits: These are used to select the analog input channels.

The stabilization wait time A is necessary for changing ADREFP1 and ADREFP0.
A = 5 μs when ADREFP1 and ADREFP0 are set to 1 and 0, respectively.
No wait time is necessary when ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The software counts up to the stabilization wait time (1 μs).

After counting up to the stabilization wait time ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

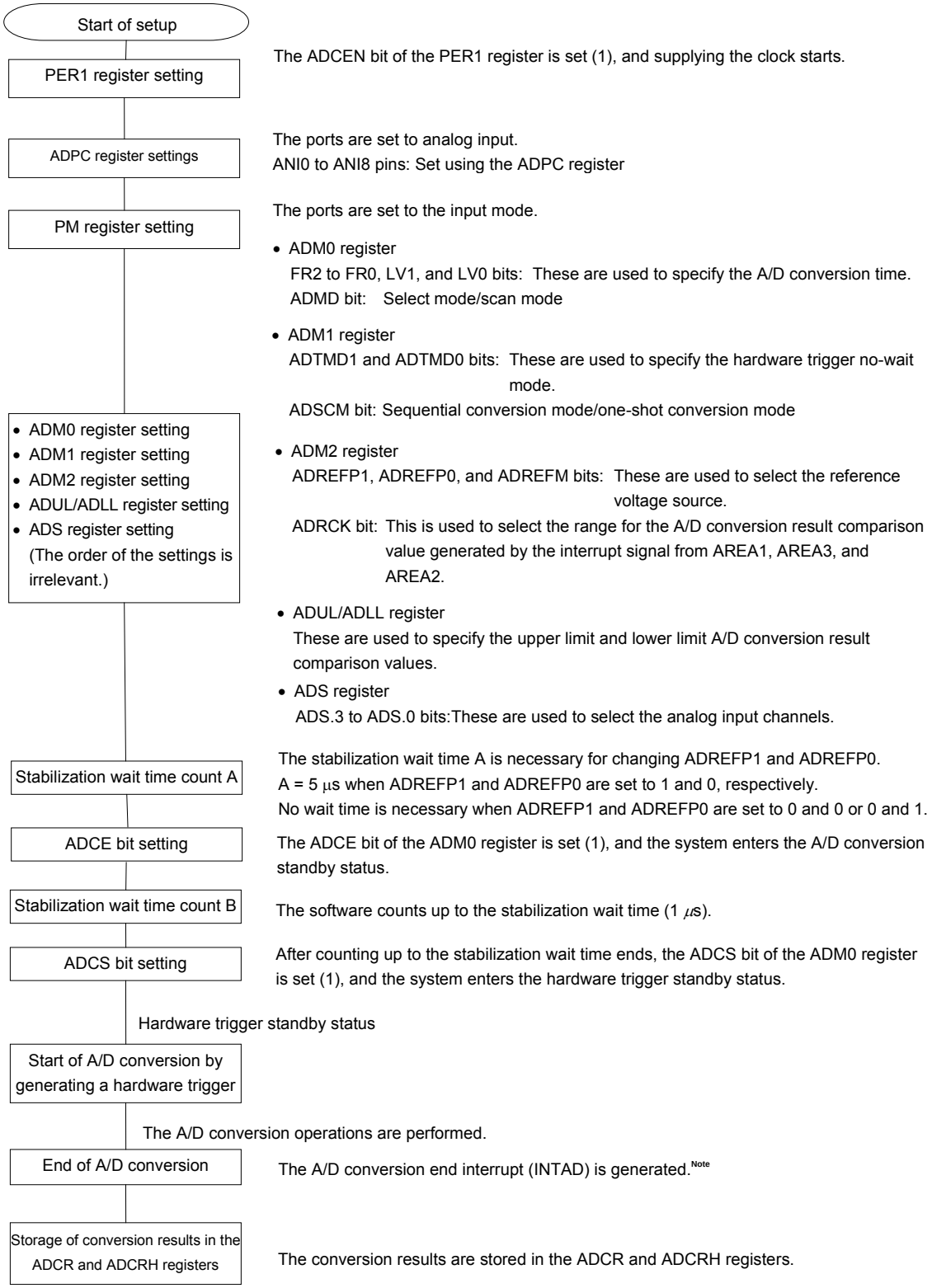
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

11.7.2 Setting up hardware trigger no-wait mode

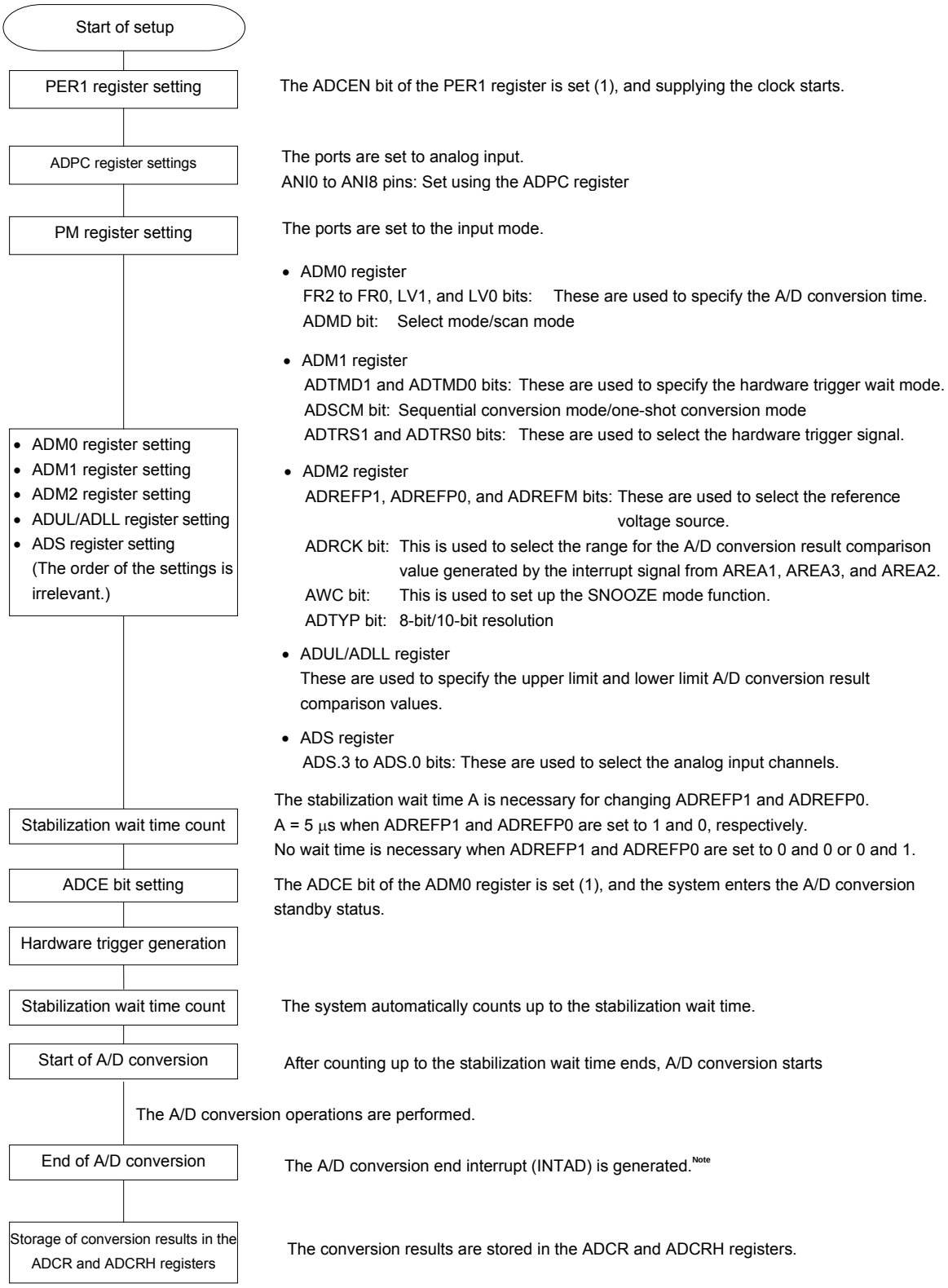
Figure 11-32. Setting up Hardware Trigger No-Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

11.7.3 Setting up hardware trigger wait mode

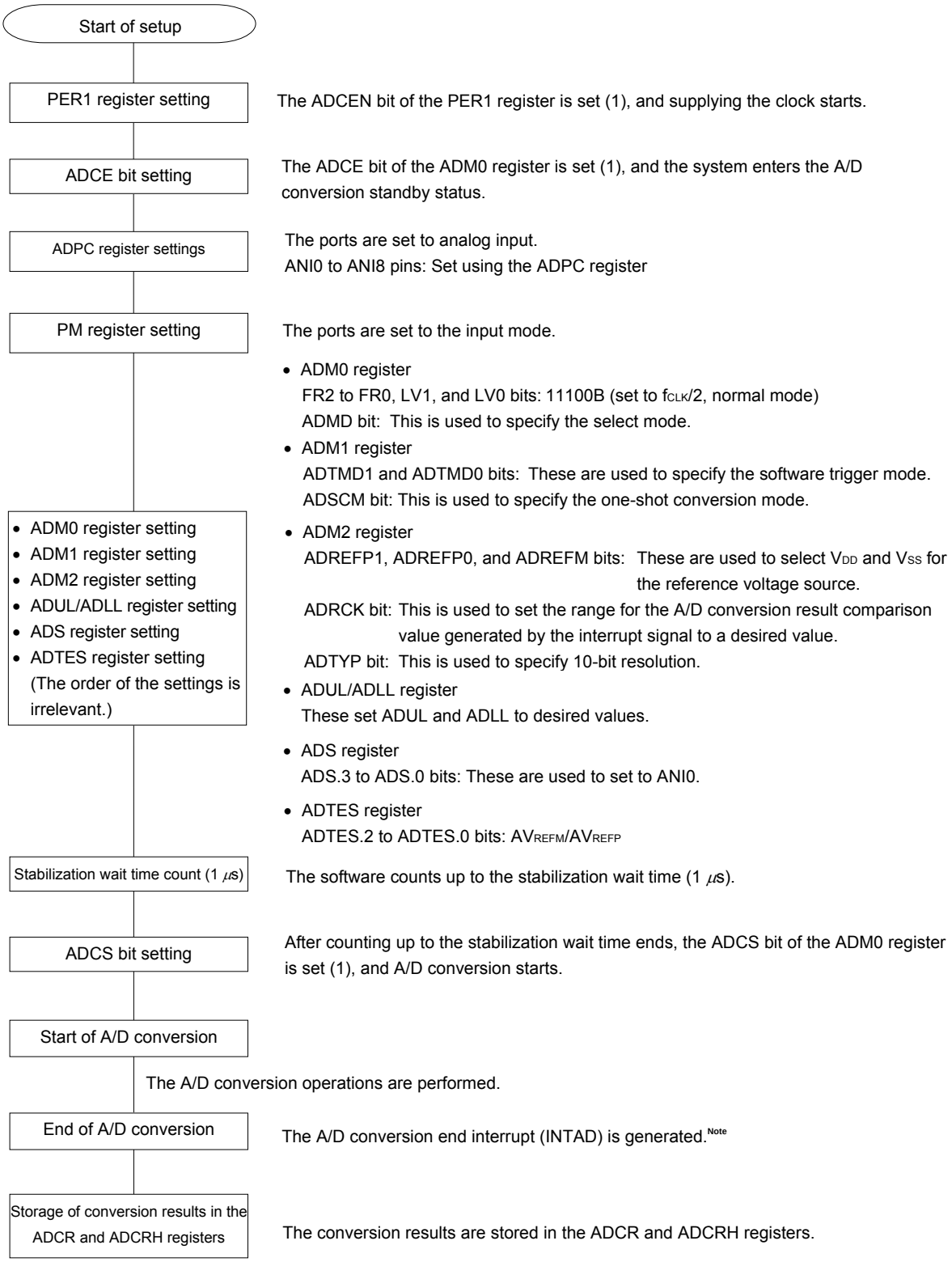
Figure 11-33. Setting up Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

11.7.4 Setting up test mode

Figure 11-34. Setting up Test Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

11.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.

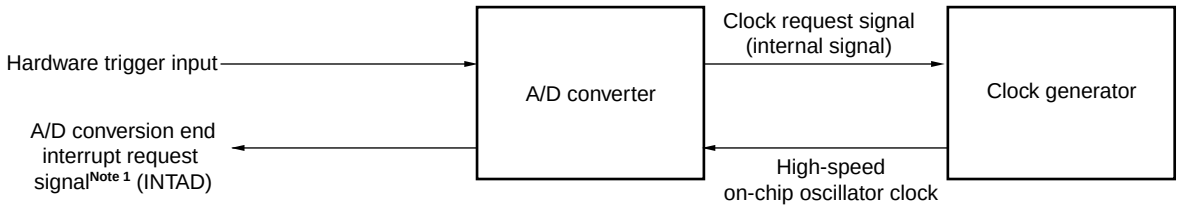
By specifying the conversion result range with ADUL and ADLL in SNOOZE mode, A/D conversion results can be checked at the specified interval, which allows the monitoring of power supply voltage and check of A/D input keys.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

Note that the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 11-35. Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode. (For details about these settings, see **11.7.3 Setting up hardware trigger wait mode Note 2**.) At this time, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

Notes 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

2. Be sure to set the ADM1 register to E2H or E3H.

Remark The hardware trigger is ADTRG.
Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is within the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

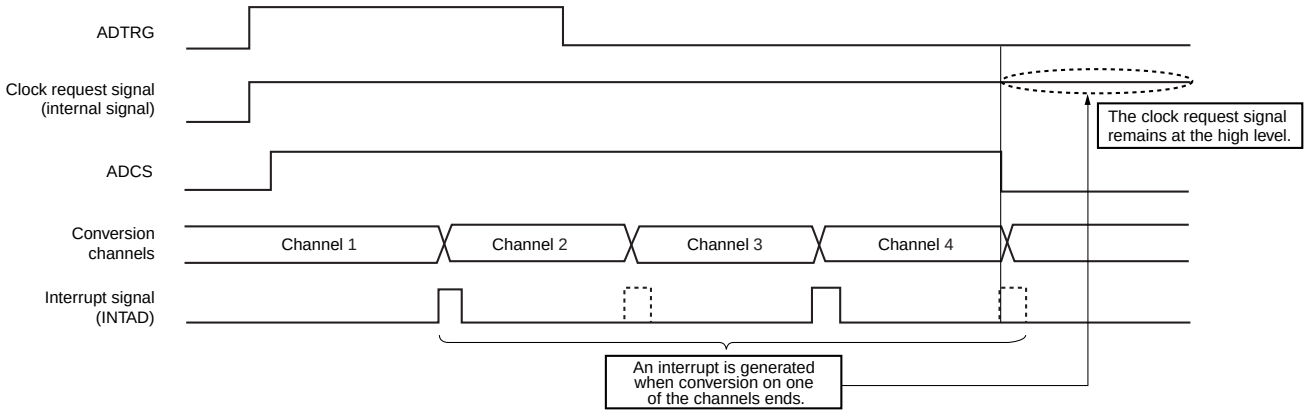
- While in the select mode

  After A/D conversion ends and the A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, clear bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 0 to release SNOOZE mode. If AWC remains 1, A/D conversion is not correctly started regardless whether the subsequent mode is SNOOZE mode or normal operation mode.

- While in the scan mode

  If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, clear bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 0 to release SNOOZE mode. If AWC remains 1, A/D conversion is not correctly started regardless whether the subsequent mode is SNOOZE mode or normal operation mode.

Figure 11-36. Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

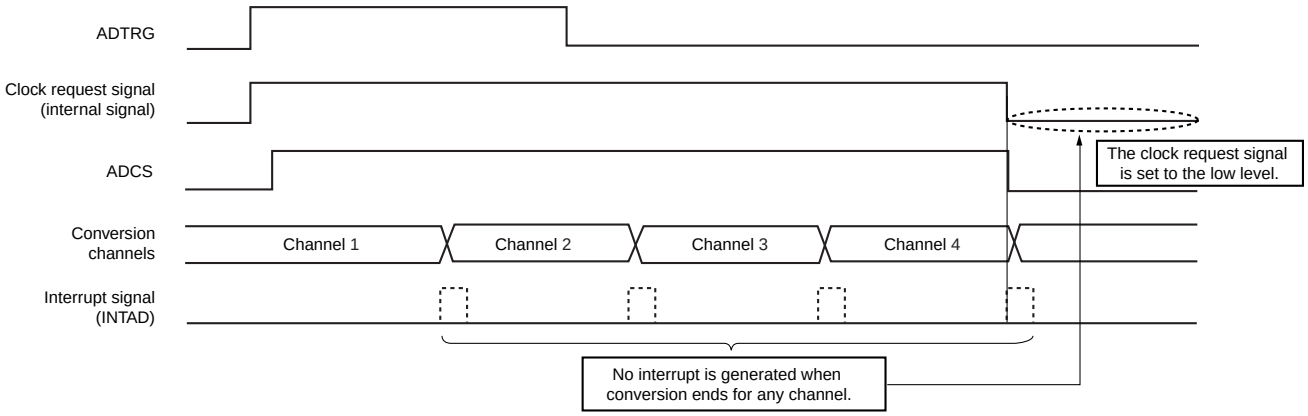
- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 11-37. Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



11.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 11-38. Overall Error

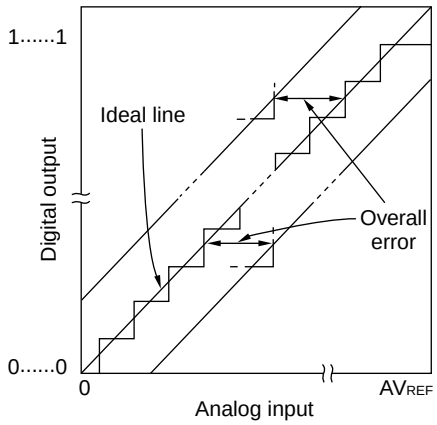
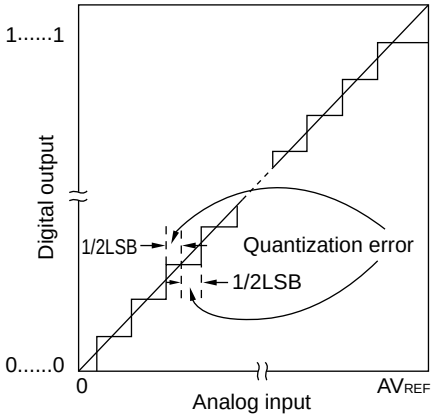


Figure 11-39. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale − 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 11-40. Zero-Scale Error

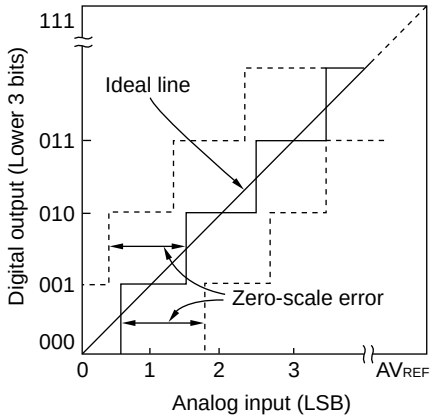


Figure 11-41. Full-Scale Error

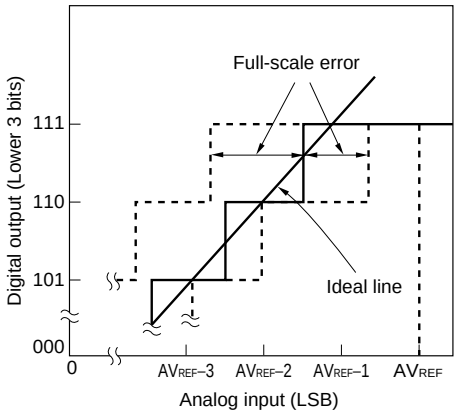


Figure 11-42. Integral Linearity Error

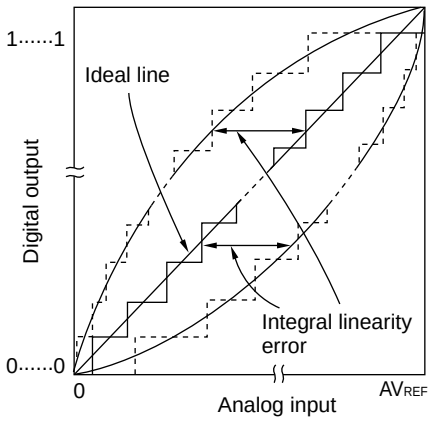
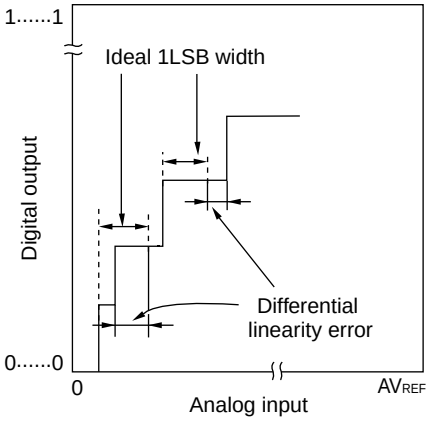


Figure 11-43. Differential Linearity Error



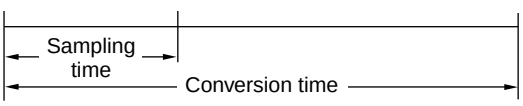
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



11.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

<R> **(2) Input range of ANI0 to ANI10 pins**

Observe the rated range of the ANI0 to ANI8 pins input voltage. If a voltage of V_{DD} and AV_{REFP} or higher and V_{SS} and AV_{REFM} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

If the internal reference voltage (1.45 V) is selected as the reference voltage on the plus side of the A/D converter, do not apply the voltage of 1.45 V or more to the pin selected by the ADS register. To the other pins, there are no problems if the applied voltage is 1.45 V or more.

Note The internal reference voltage (1.45 V) can only be selected in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI8 pins.

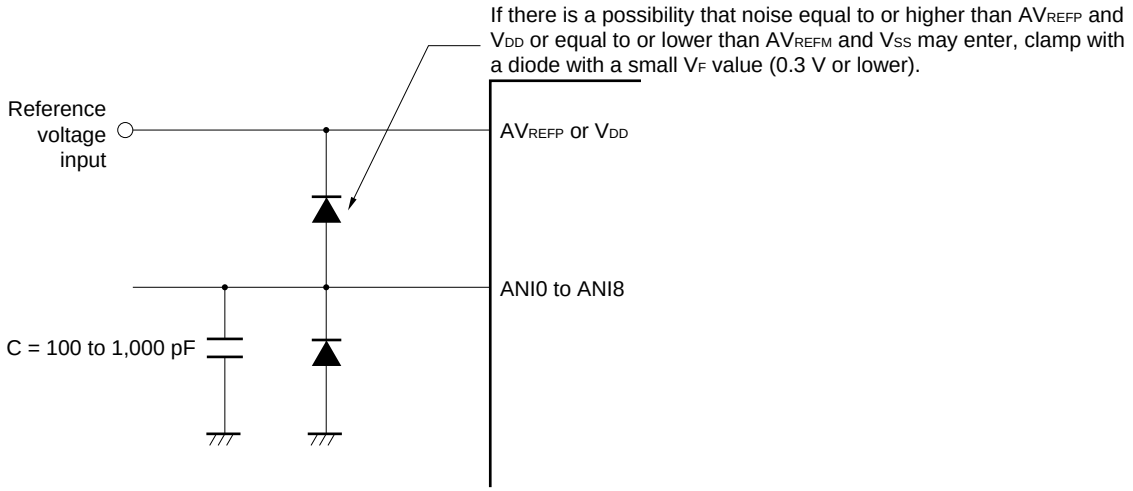
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 11-44 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 11-44. Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI8) are also used as input port pins (P20 to P27, P150).
When A/D conversion is performed with any of the ANI0 to ANI8 pins selected, do not change the output value to P20 to P27 and P150 while conversion is in progress; otherwise the conversion accuracy may be degraded.

<2> If the pins adjacent to the pins currently used for A/D conversion is used as digital I/O ports, the expected value of the A/D conversion may not be obtained due to coupling noise. Take care not to input or output such a pulse to these pins.

(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI8 pins (see **Figure 11-44**).

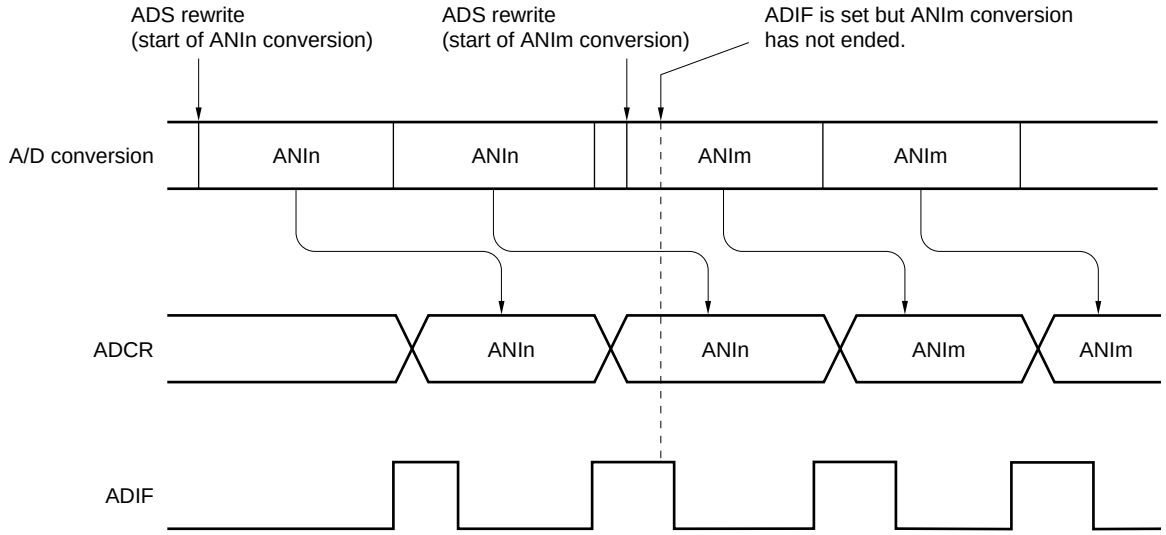
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 11-45. Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or ADPC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 11-46. Internal Equivalent Circuit of ANIn Pin

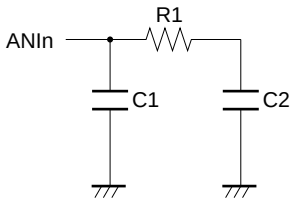


Table 11-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn pin	R1 [kΩ]	C1 [pF]	C2 [pF]
$4.0 \leq V_{DD} \leq 5.5$	ANI0 to ANI7	14	8	2.5
	ANI8	18	8	7.0
$2.7 \leq V_{DD} \leq 4.0$	ANI0 to ANI7	39	8	2.5
	ANI8	53	8	7.0
$1.8 \leq V_{DD} \leq 2.7$	ANI0 to ANI7	231	8	2.5
	ANI8	321	8	7.0

Remark The resistance and capacitance values shown in Table 11-6 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

CHAPTER 12 SERIAL ARRAY UNIT

Serial array unit 0 has two serial channels, serial array unit 1 has two serial channels. Serial array unit 0 channel can
<R> achieve 3-wire serial (CSI) , and UART. Serial array unit 1 channel can achieve 3-wire serial (CSI), and simplified I^2C communication.

Function assignment of each channel supported by the RL78/D1A is as shown below.

- 48, 64, 80-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	–	–
	1	CSI01	–	–
1	0	–	–	–
	1	–	–	IIC11

- 100 products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	–	–
	1	CSI01	–	–
1	0	CSI10	–	–
	1	–	–	IIC11

<R> - 128-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	CSI01		–
1	0	CSI10	–	–
	1	–	–	IIC11

<R> When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and CSI01 cannot be used, but CSI10, or IIC11 can be used.

Caution Most of the following descriptions in this chapter use the units and channels of the 128-pin products as an example.

12.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/D1A has the following features.

12.1.1 3-wire serial I/O (CSI00, CSI01, CSI10)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10) Communication.**

[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

<R> **12.1.2 UART (UART0)**

This is an asynchronous communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

For details about the settings, see **12.6 Operation of UART (UART0) Communication.**

[Data transmission/reception]

- Data length of 7, 8, 9 or 16 bits (UART0)
- 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/reception interrupt

[Error detection flag]

- Framing error, parity error, or overrun error

12.1.3 Simplified I^2C (IIC11)

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^2C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **12.6 Operation of Simplified I^2C (IIC11) Communication**.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function **Note** and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I^2C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEm.n bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **12.6.3 (2)** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

12.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

<R>

Table 12-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	16 bits
Buffer register	Serial data register mn (SDRmn)Note
Serial clock I/O	SCK00, SCK01, SCK10 pins (for 3-wire serial I/O), SCL11 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10 pins (for 3-wire serial I/O) , RxD0 pin (for UART)
Serial data output	SO00, SO01, SO10 pins (for 3-wire serial I/O) , TxD0 pin (for UART)
Serial data I/O	SDA11 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable registers 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm)
	<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 communication pin select register (STSEL)
	• Port input mode registers 0, 1, 3, 5 to 7, 13 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13) • Port output mode register (POM) • Port mode registers 0, 1, 3, 5 to 7, 13 (PM0, PM1, PM3, PM5 to PM7, PM13) • Port registers 0, 1, 3, 5 to 7, 13 (P0, P1, P3, P5 to P7, P13)

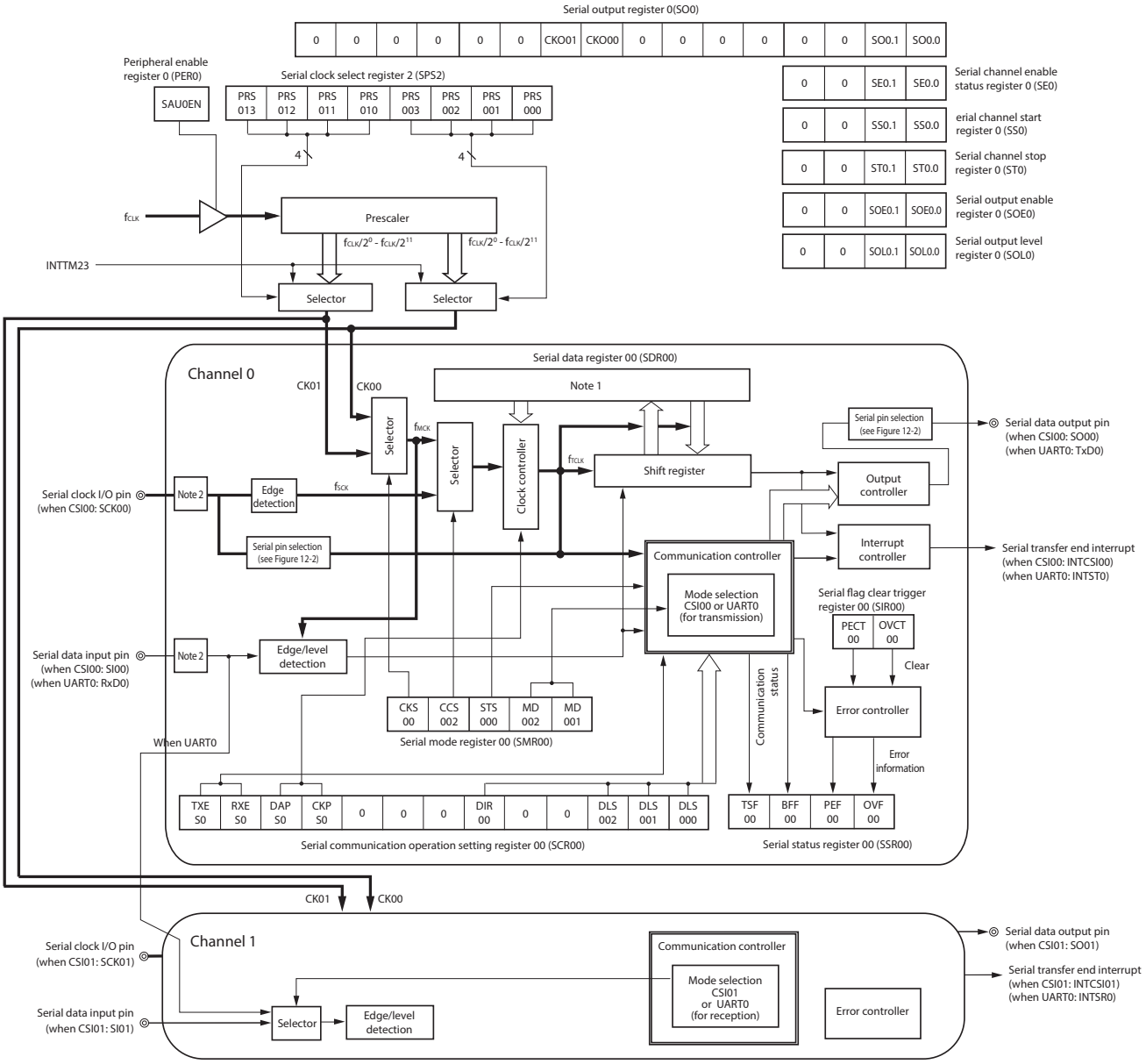
Note. During operation (SEmn = 1)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10), q: UART number (q = 0), r: IIC number (r = 11)

Figure 12-1 shows the block diagram of the serial array unit 0.

<R>

Figure 12-1. Block Diagram of Serial Array Unit 0



Notes 1. When operation is stopped (SEmn = 0), the higher 7 bits become the clock division setting section and the lower bits are fixed to 0.
During operation (SEmn = 1), it becomes a buffer register.

2. Serial pin selection (see Figure 12-2)

<R>

Figure 12-2. Port Configuration Diagram of Serial Array Unit 0

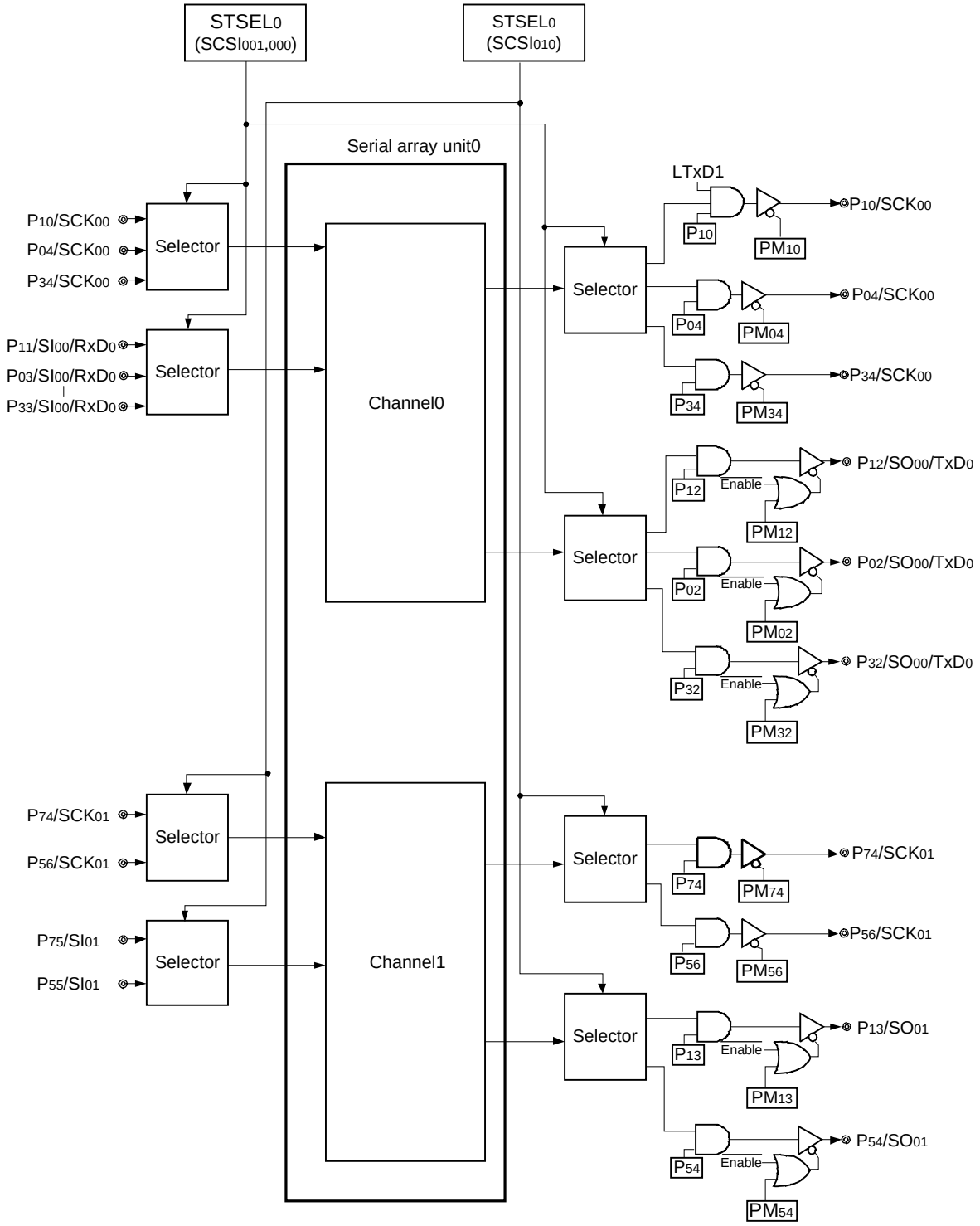
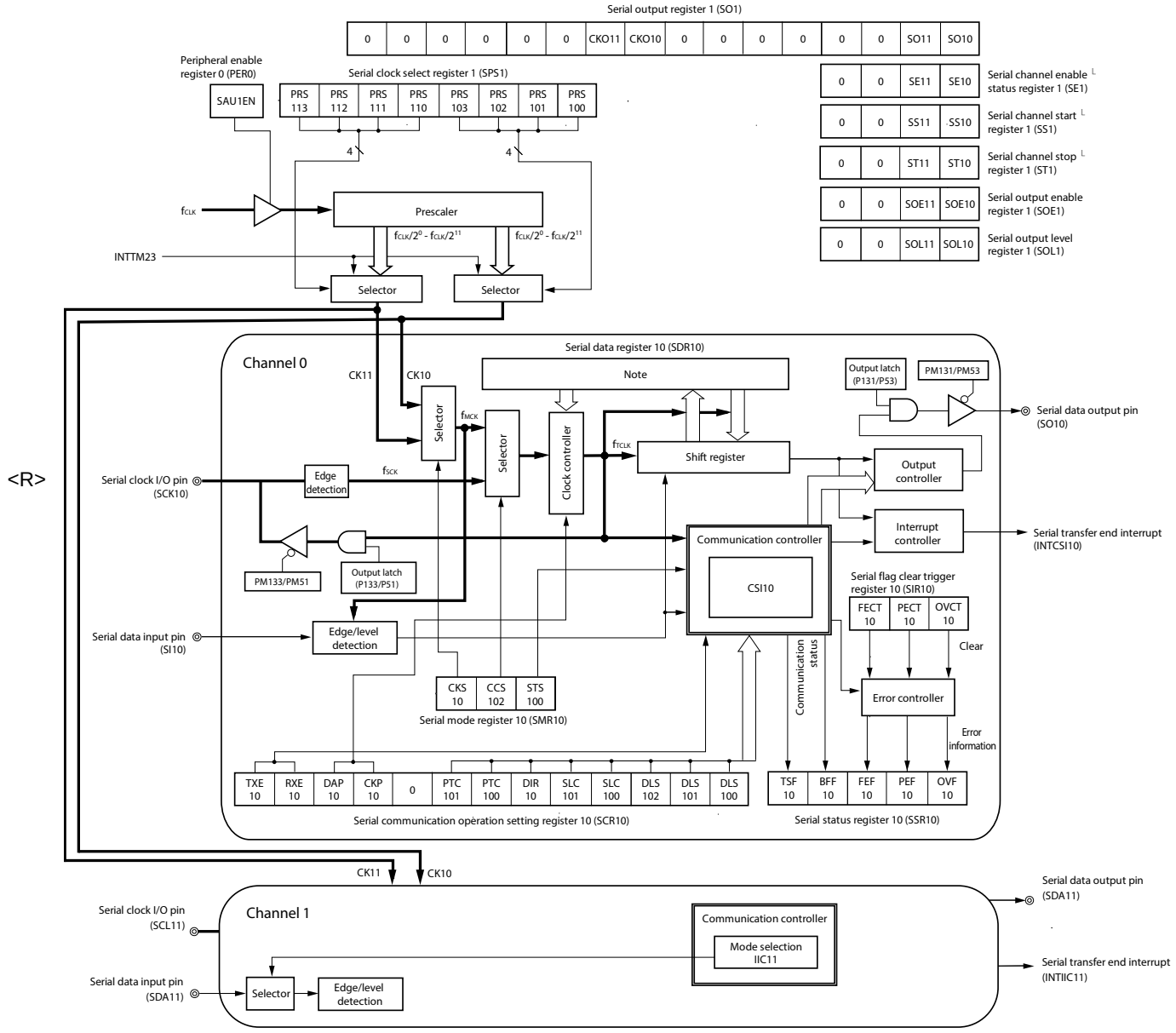


Figure 12-3 shows the block diagram of the serial array unit 1.

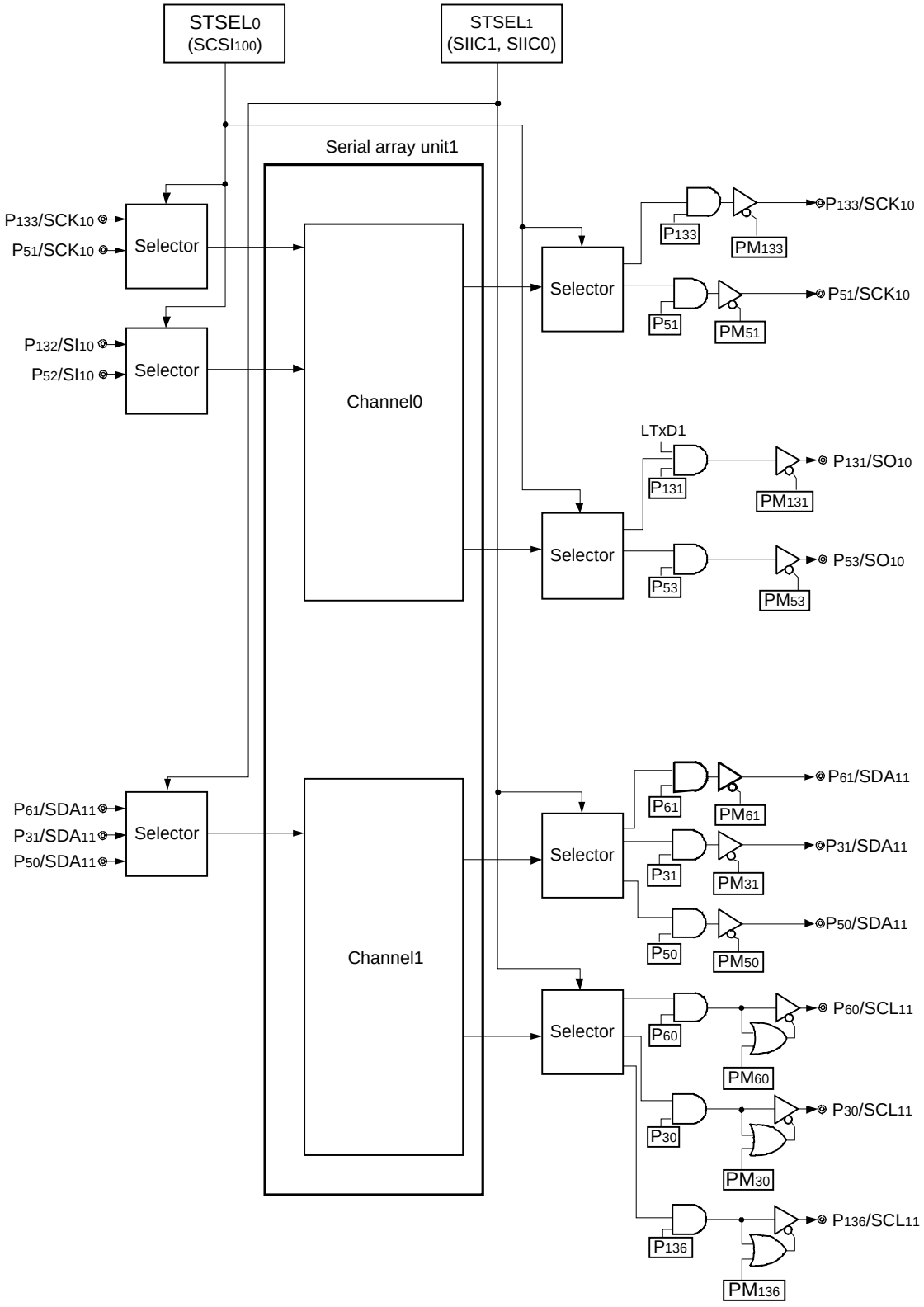
Figure 12-3. Block Diagram of Serial Array Unit 1



Notes 1. When operation is stopped (SEmn = 0), the higher 7 bits become the clock division setting section and the lower bits are fixed to 0.
During operation (SEmn = 1), it becomes a buffer register.

2. Serial pin selection (see Figure 12-2)

Figure 12-4. Port Configuration Diagram of Serial Array Unit 1



(1) Shift register

This is an 16-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 (SEmn = 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 4 to 0 (DLSmn4 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)
- 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 … SIOpL
- <R> UARTq reception … RxDq (UARTq receive data register)
- UARTq transmission … TxDq (UARTq transmit data register)
- IICr communication … SDRrL (IICr data register)

Reset signal generation clears this register to 0000H.

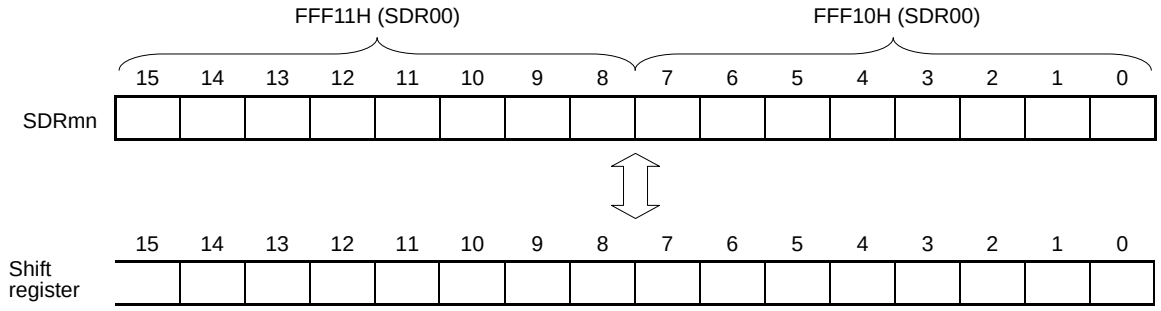
Note Writing in 8-bit units is prohibited when the operation is stopped (SEm.n = 0).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),

<R> p: CSI number (p = 00, 01, 10), q: UART number (q = 0 to 2), r: IIC number (r = 11)

Figure 12-5. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF14H, FFF15H (SDR10), FFF16H, FFF17H (SDR11) After reset: 0000H R/W



Remarks 1. For the function of the higher 7 bits of SDRmn, see **12.3 Registers Controlling Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),

12.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial communication pin select register (STSEL)
- Port input mode registers 0, 1, 3, 5 to 7, 13 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13)
- Port output mode register (POM)
- Port mode registers 0, 1, 3, 5 to 7, 13 (PM0, PM1, PM3, PM5 to PM7, PM13)
- Port registers 0, 1, 3, 5 to 7, 13 (P0, P1, P3, P5 to P7, P13)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(1) Peripheral enable registers 0 (PER0)

PER0 are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 3 (SAU0EN) of PER0.

When serial array unit 1 is used, be sure to set bit 4 (SAU1EN) of PER0.

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

Reset signal generation clears this register to 00H.

Figure 12-6. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN

SAUmEN	Control of serial array unit m input clock supply (m = 0, 1)
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Enables input clock supply. • SFR used by serial array unit m can be read/written.

Cautions 1. When setting serial array unit m, be sure to set the SAUmEN bit 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 Serial communication pin select register (STSEL), port input mode register (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13), port output mode register (POM), port mode registers (PM0, PM1, PM3, PM5 to PM7, PM13), and port registers (P0, P1, P3, P5 to P7, P13)).

2. After setting the SAUmEN bit to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark m: Unit number (m = 0 to 1)

(2) Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEm.n = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 12-7. Format of Serial Clock Select Register m (SPSm)

Address: F0116H, F0117H (SPS0), F0146H, F0147H (SPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

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	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	4 MHz	8 MHz	16 MHz	24 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	2 MHz	4 MHz	8 MHz	12 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	1 MHz	2 MHz	4 MHz	6 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	500 kHz	1 MHz	2 MHz	3 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	250 kHz	500 kHz	1 MHz	1.5 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	125 kHz	250 kHz	500 kHz	750 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	62.5 kHz	125 kHz	250 kHz	375 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	31.3 kHz	62.5 kHz	125 kHz	187.5 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	15.6 kHz	31.3 kHz	62.5 kHz	93.75 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	7.81 kHz	15.6 kHz	31.3 kHz	46.88 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	3.91 kHz	7.81 kHz	15.6 kHz	23.44 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	1.95 kHz	3.91 kHz	7.81 kHz	11.72 kHz	15.6 kHz
1	1	1	1	INTTM23					
Other than the 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).

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting bit 3 (SAU0EN) and bit 4 (SAU1EN) of the PER0 register to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

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

2. m: Unit number (m = 0, 1), p = 0, 1

<R> **(3) Serial mode register mn (SMRmn)**

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (CSI, UART, or I^2C), and an interrupt source.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEm.n = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 12-8. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0108H, F0109H (SMR00), F010AH, F010BH (SMR01), F0138H, F0139H (SMR10), F013AH, F013BH (SMR11) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STS mn Note	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01 register only.

Caution Be sure to clear bits 13 to 9, 7, 6, 4, and 3 to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

Figure 12-8. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0108H, F0109H (SMR00), F010AH, F010BH (SMR01), F0138H, F0139H (SMR10), F013AH, F013BH (SMR11)

After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0 Note1	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note1	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 Note2
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Notes 1. The SMR01 register only.

2. The SMR00 and SMR01 registers.

Remarks 1. See Table 12-1 Serial Function Assignment of Each Product for details of the modes implemented for each unit and product.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

<R> **(4) Serial communication operation setting register mn (SCRmn)**

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEm.n = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SCRmn register to 0087H.

Figure 12-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F010CH, F010DH (SCR00), F010EH, F010FH (SCR01), F013CH, F013DH (SCR10), F013EH, F013FH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	0	PTC mn1	PTC mn0	DIR mn	0	SLC mn1 Note 1	SLC mn0	DLS mn3	DLS mn2	DLS mn1 Note 2	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in CSI mode	Type
0	0	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	1
0	1	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	2
1	0	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	3
1	1	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

Notes 1. The SCR00 register only.

2. The SCR00 and SCR01 registers only. For other registers, the bit is fixed to 1.

Caution Be sure to clear bits 6, 10 and 11 of SCRmn to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 12-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F010CH, F010DH (SCR00), F010EH, F010FH (SCR01), F013CH, F013DH (SCR10), F013EH, F013FH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	0	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	DLSmn3	DLSmn2	DLSmn1 Note 2	DLSmn0

PTCmn1	PTCmn0	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 3	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C mode.			

DIRmn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 Note 1	SLCmn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

Notes 1. The SCR00 register only.

2. The SCR00 and SCR01 registers only. For other registers, the bit is fixed to 1.

3. 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, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10)

Figure 12-9. 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)

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 Note 1	SLC mn0	DLS mn3	DLS mn2	DLS mn1 Note 2	DLS mn0

DLS mn3	DLS mn2	DLS mn1 Note 2	DLS mn0	Setting of data length in CSI mode	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 the above				Setting prohibited			
Be sure to set DLSmn3 to DLSmn0 = 0111B in the simplified I^2C mode.							

Notes 1. The SCR00 register only.

2. The SCR00 and SCR01 registers only. For other registers, the bit is fixed to 1.

Caution Be sure to clear bits 6 , 10 and 11 of SCRmn to “0”.

Remark m: Unit number (m = 0, 1), 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 CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of SDRmn is used as the transfer clock.

See **12.2 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 12-10. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF14H, FFF15H (SDR10), FFF16H, FFF17H (SDR11) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0	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}
0	0	0	0	0	0	1	$f_{MCK}/2$
0	0	0	0	0	1	0	$f_{MCK}/3$
0	0	0	0	0	1	1	$f_{MCK}/4$
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{MCK}/127$
1	1	1	1	1	1	1	$f_{MCK}/128$

Cautions 1. When operation is stopped (SEm.n = 0), be sure to clear bits 8 to 0 to "0".

<R> **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 **12.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

<R> **(6) Serial flag clear trigger register mn (SIRmn)**

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 12-11. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0104H, F0105H (SIR00), F0106H, F0107H (SIR01), F0134H, F0135H (SIR10), F0136H, F0137H (SIR11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn Note	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECT11	Clear trigger of parity error of SCL11
0	No trigger operation
1	Clears PEF_{11} bit of SSR_{11} register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Cautions 1. Be sure to clear bits 15 to 2 of SIRmn 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, 1), n: Channel number (n = 0, 1)

<R> **(7) Serial status register mn (SSRmn)**

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n.

The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 12-12. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0130H, F0131H (SSR10), F0132H, F0133H (SSR11) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STm.n bit of the STm register is set to 1 (communication is stopped) or the SSm.n bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STm.n bit of the STm register is set to 1 (communication is stopped) or the SSm.n bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note The SSR01 register only.

Caution If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

Figure 12-12. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0130H, F0131H (SSR10), F0132H, F0133H (SSR11) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn Note	PEFmn	OVFmn

FEFmn Note	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEFmn	Parity error detection flag of channel 11
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in CSI mode.	

Note The SSR01 register only.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), PEF bit is SSR11 only

(8) Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEm.n) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSm.n bit is a trigger bit, it is cleared immediately when SEm.n = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 12-13. Format of Serial Channel Start Register m (SSm)

Address: F0112H, F0113H (SS0), F0142H, F0143H (SS1), After reset: 0000H R/W

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	SSm1	SSm0

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEm.n bit to 1 and enters the communication wait status Note.

Note If the SSmn bit is set to 1 during communication, the communication stops and the communication wait state is entered. At this time, the values of the control registers and shift register and the status of the SCKmn and SOmn pins and the PEFmn, and OVFmn flags are held.

Caution Be sure to clear bits 15 to 2 of SSm to "0".

Remarks 1. When the SSm register is read, 0000H is always read.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(9) Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STm.n), the corresponding bit (SEm.n) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STm.n bit is a trigger bit, it is cleared immediately when SEm.n = 0.

The STm register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

Figure 12-14. Format of Serial Channel Stop Register m (STm)

Address: F0114H, F0115H (ST0), F0144H, F0145H (ST1), After reset: 0000H R/W

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	STm1	STm0

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEm.n bit to 0 and stops the communication operation[Note].

Note Communication stops while holding the value of the control register and shift register, and the status of the serial clock I/O pin, serial data output pin, and each error flag (PEFmn: parity error flag, OVFmn: overrun error flag).

Caution Be sure to clear bits 15 to 2 of STm to "0".

Remarks 1. When the STm register is read, 0000H is always read.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(10) Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 12-15. Format of Serial Channel Enable Status Register m (SEm)

Address: F0110H, F0111H (SE0), F0140H, F0141H (SE1) After reset: 0000H R

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	SEm1	SEm0

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(11) Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOm.n bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOm.n bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 12-16. Format of Serial Output Enable Register m (SOEm)

Address: F011AH, F011BH (SOE0), F014AH, F014BH (SOE1), After reset: 0000H R/W

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	SOEm1	SOEm0

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 2 of SOEm.

Remark m: Unit number (m = 0, 1), n: Channel number (n =0, 1)

(12) Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOm.n bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOm.n bit 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.

The CKOmn bit 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 the CKOmn bit can be changed only by a serial communication operation.

To use the serial interface pin as a port function pin, set the corresponding CKOmn and SOm.n bits to “1”.

The SOm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOm register to 0F0FH.

Figure 12-17. Format of Serial Output Register m (SOm)

Address: F0118H, F0119H (SO0), F0148H, F0149H (SO1), After reset: 0303H R/W

Symbol	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

CKOmn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SOm.n	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 “0”.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

<R> **(13) Serial output level register m (SOLm)**

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 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 SOm.n bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEm.n = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 12-18. Format of Serial Output Level Register m (SOLm)

Address: F0120H, F0121H (SOL0), F0150H, F0151H (SOL1), After reset: 0000H R/W

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	0	SOLm.0

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 1 to “0”.

Remark m: Unit number (m = 0, 1)

(14) Serial communication pin select register 0, 1 (STSEL0, STSEL1)

These registers are used for alternate switch of serial input/output pins.

STSEL0 and STSEL1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-19. Serial communication pin select register 0 (STSEL0)

Address: FFF3CH After reset: 00H R/W (Note: Bits 5 and 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
STSEL0	0	SCSI100	0	SCSI010	SCSI001	SCSI000	SUARTF1	SUARTF0

SUARTF0	Communication pin selection of UARTF0	
	LTxD0	LRxD0
0	P71	P70
1	P15	P14

SUARTF1	Communication pin selection of UARTF1	
	LTxD1	LRxD1
0	P10	P11
1	P131	P132

<R>

STSCSI00	SCSI000	CSI00/UART0 communication pin selection		
		SCK00	SI00	SO00
0	0	P10	P11	P12
0	1	P04	P03	P02
1	0	P34	P33	P32
Other tan above		Setting prohibited (same as “00” setting)		

SCSI010	CSI01 communication pin selection		
	SCK01	SI01	SO01
0	P74	P75	P13
1	P56	P55	P54

SCSI100	CSI10 communication pin selection		
	SCK10	SI10	SO10
0	P133	P132	P131
1	P51	P52	P53

Figure 12-20. Serial communication pin select register 1 (STSEL1)

Address: FFF3D After reset: 00H R/W (Note: Bits 4, 5 are read only bit)

Symbol	7	6	5	4	3	2	1	0
STSEL1	SIIC1	SIIC0	0	0	SCAN1	SCAN0	TMCAN1	TMCAN0

SIIC1	SIIC0	Communication pin selection of IIC11	
		SCL11	SDA11
0	0	P60	P61
0	1	P30	P31
1	0	P136	P50
Other than the above		Setting prohibited	

(15) Port input mode registers 0, 1, 3, 5 to 7, and 13 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13)

These registers set the input buffer of ports 0, 1, 3, 5 to 7, and 13 in 1-bit units.

The PIM0, PIM1, PIM3, PIM5 to PIM7, and PIM13 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PIM0, PIM1, PIM3, PIM5 to PIM7, and PIM13 registers to 00H.

Figure 12-21. Format of Port Input Mode Registers 0, 1, 3, 5 to 7, and 13 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13)

Address: F0040H After reset: 00H R/W (Note: Bits 0, 2 to 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM0	0	0	0	0	0	0	PIM0.1	0

Address: F0041H After reset: 00H R/W (Note: Bits 2 to 6 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM1	PIM1.7	0	0	0	0	0	PIM1.1	PIM1.0

Address: F0043H After reset: 00H R/W (Note: Bits 0, 2 to 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM3	0	0	0	0	0	0	PIM3.1	0

Address: F0045H After reset: 00H R/W (Note: Bits 3, 4 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM5	PIM5.7	PIM5.6	PIM5.5	0	0	PIM5.2	PIM5.1	PIM5.0

Address: F0046H After reset: 00H R/W (Note: Bits 0, 2, 4 to 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM6	0	0	0	0	PIM6.3	0	PIM6.1	0

Address: F0047H After reset: 00H R/W (Note: Bits 1 to 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM7	0	0	0	0	0	0	0	PIM7.0

Address: F004DH After reset: 00H R/W (Note: Bits 0 to 4, 6, 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
PIM13	0	0	PIM13.5	0	0	0	0	0

PIMmn	Port input threshold selection
0	Schmit1 input mode
1	Schmit3 input mode

Bit name	PIM5.2	PIM5.1	PIM5.0	PIM3.1	PIM1.7	PIM1.1	PIM1.0	PIM0.1
Port input function	P52/SI10	P51/SCK10	P50/SDA11	P31/SDA11	P17	P11/LRxD1/ SI00	P10/SCK00	P01/ CRxD0
Bit name		PIM135	PIM70	PIM63	PIM61	PIM57	PIM56	PIM55
Port input function		P135/ CRxD1	P70/ CRxD0/ LRxD0	P63/ CRxD1	P61/SDA11	P57	P56/SCK01	P55/SI01

Remark For details of the Port Input mode registers (PIM), see **4.3 (4) Port Input mode registers**.

(16) Port output mode register (POM)

<R> These registers set the output mode of P30, P31, P50, P60, P61, P136 in 1-bit units.

Port output mode is set by 1-bit unit. N-ch open drain output (VDD tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA11 and SCL11 pins during simplified I^2C communication with an external device of the same potential.

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

Reset signal generation clears this register to 00H.

Figure 12-22. Format of Port Output Mode Register (POM)

Address: F006FH After reset: 00H R/W (Note: Bits 6, 7 are read only bit)

Symbol	7	6	5	4	3	2	1	0
POM	0	0	POM5	POM4	POM3	POM2	POM1	POM0

POMnx	Port input threshold selection
0	Normal output (CMOS) mode
1	Nch-OD output (VDD tolerance) mode

Bit name	POM5	POM4	POM3	POM2	POM1	POM0
Port output function	P50/ TO02/ SDA11	P136/ TO00/ SCL11	P31/ TO21/ SDA11	P30/ TO20/ SCL11	P61/ TO21/ SDA11	P60/ TO20/ SCL11

If use the alternate function of IIC, port output need to be set as Nch open-drain (Nch-OD) output. At that time, output signal can also enter into port input (ENI=ON) and on -chip pull-up resistors should not be active (disabled by circuit).

Remark For details of the Port output mode register (POM), see 4.3 (5) Port output mode register.

<R> **(17) Port mode registers 0, 1, 3, 5 to 7, 13 (PM0, PM1, PM3, PM5 to PM7, PM13)**

These registers set input/output of ports 0, 1, 3, 5 to 7, 13 in 1-bit units.

When using the ports (such as P02/ SO00/TxD0/TI02/TO02/ TI12/TO12) to be shared with the serial data output pin for serial data output, set the port mode register (PMxx) bit corresponding to each port to 0. And set the port register (Pxx) bit corresponding to each port to 1.

Example: When using P02/SO00/TxD0/TI02/TO02/TI12/TO12 for serial data output or serial clock output
Set the PM0.2 bit of the port mode register 0 to 0.
Set the P0.2 bit of the port register 0 to 1.

When using the ports (such as P03/SI00/RxD0/TI03/TO03/TI13/TO13) to be shared with the serial data input pin for serial data input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P03/SI00/RxD0/TI03/TO03/TI13/TO13 for serial data input
Set the PM0.3 bit of port mode register 0 to 1.
Set the P0.3 bit of port register 0 to 0 or 1.

The PM0, PM1, PM3, PM5 to PM7, PM13 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets the PM0, PM1, PM3, PM5 to PM7, PM13 registers to FFH.

Figure 12-23. Format of Port Mode Registers 0, 1, 3, 5 to 7, 13 (PM0, PM1, PM3, PM5 to PM7, PM13)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM0.7	PM0.6	PM0.5	PM0.4	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	PM3.7	PM3.6	PM3.5	PM3.4	PM3.3	PM3.2	PM3.1	PM3.0

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: FFF2DH After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	PM13.6	PM13.5	PM13.4	PM13.3	PM13.2	PM13.1	0

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5 to 7, 13 ; n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

12.4 Operation stop mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

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

12.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 3 (SAU0EN) of PER0 to 0.

To stop the operation of serial array unit 1, set bit 4 (SAU1EN) of PER0 to 0.

Figure 12-24. Peripheral Enable Registers 0 (PER0) 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	RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN
	×	×	×	0/1	0/1	×	×	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Caution If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read.
Note that this does not apply to the following registers.

- **Serial communication pin select register (STSEL)**
- **Port input mode registers 0, 1, 3, 5 to 7, and 13 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13)**
- **Port output mode register (POM)**
- **Port mode registers 0, 1, 3, 5 to 7, 13 (PM0, PM1, PM3, PM5 to PM7, PM13)**
- **Port registers 0, 1, 3, 5 to 7, 13 (P0, P1, P3, P5 to P7, P13)**

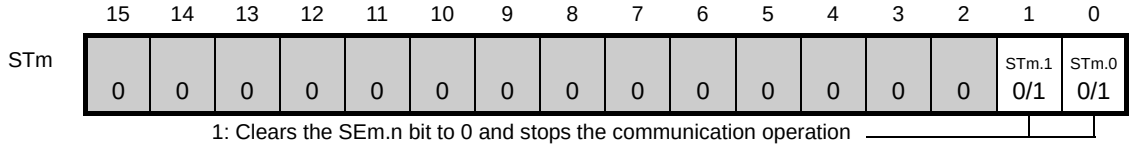
Remark m: Unit number (m = 0, 1)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

12.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

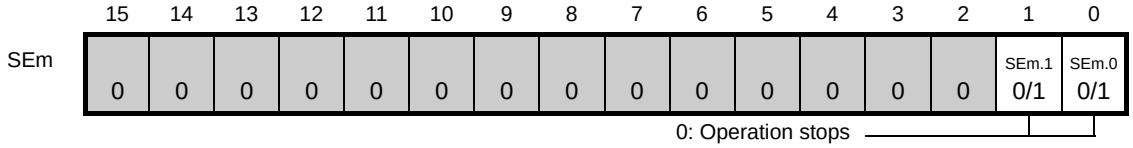
Figure 12-25. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.



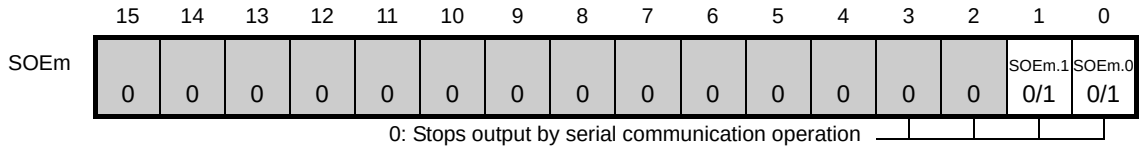
* Because the STm.n bit is a trigger bit, it is cleared immediately when SEm.n = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether serial transmission/reception operation of each channel is enabled or stopped.



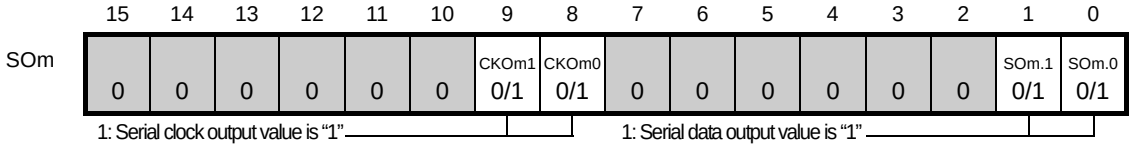
* The SEm register is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.



* For channel n, whose serial output is stopped, the SOm.n bit value of the SOm register can be set by software.

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.



* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOm.n bits to “1”.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

2. ☐ : Setting disabled (set to the initial value**), 0/1: Set to 0 or 1 depending on the usage of the user**

12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 to 16 bits (CSI00, CSI01, CSI10)
- 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
- Maximum transfer rate
  During master communication: Max. $f_{CLK}/4$[Note]

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE) and CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10) are channels 0 and 1 of SAU0, and channels 0 of SAU1.

Unit	Channel	Used as CSI	Used as Simplified I^2C
0	0	CSI00	–
	1	CSI01	–
1	0	CSI10	–
	1	–	IIC11

3-wire serial I/O (CSI00, CSI01, CIS10) performs the following six types of communication operations.

- Master transmission (See **12.5.1**.)
- Master reception (See **12.5.2**.)
- Master transmission/reception (See **12.5.3**.)
- Slave transmission (See **12.5.4**.)
- Slave reception (See **12.5.5**.)
- Slave transmission/reception (See **12.5.6**.)

12.5.1 Master transmission

Master transmission is an operation wherein the RL78/D1A outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	7 to 16 bits		
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] Note		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the serial clock operation. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Not reversed (Data output at the falling edge of SCK, data input at the rising edge of SCK) • CKPmn = 1: Reversed (Data output at the rising edge of SCK, data input at the falling edge of SCK)		
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 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10
f_{CLK}: System clock frequency

(1) Register setting

Figure 12-26. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (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	SOm.1 0/1	SOm.0 0/1

Communication starts when these bits are 1 if the data phase is not reversed (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	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Interrupt source 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
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)

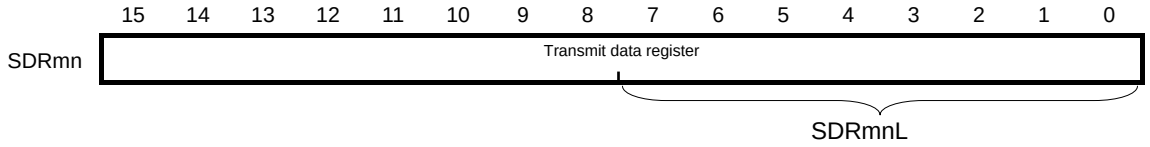
(i) When operation is stopped (SEm.n = 0)

	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 ☐: 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
n : Channel number (n = 0, 1)

Figure 12-26. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (2/2)

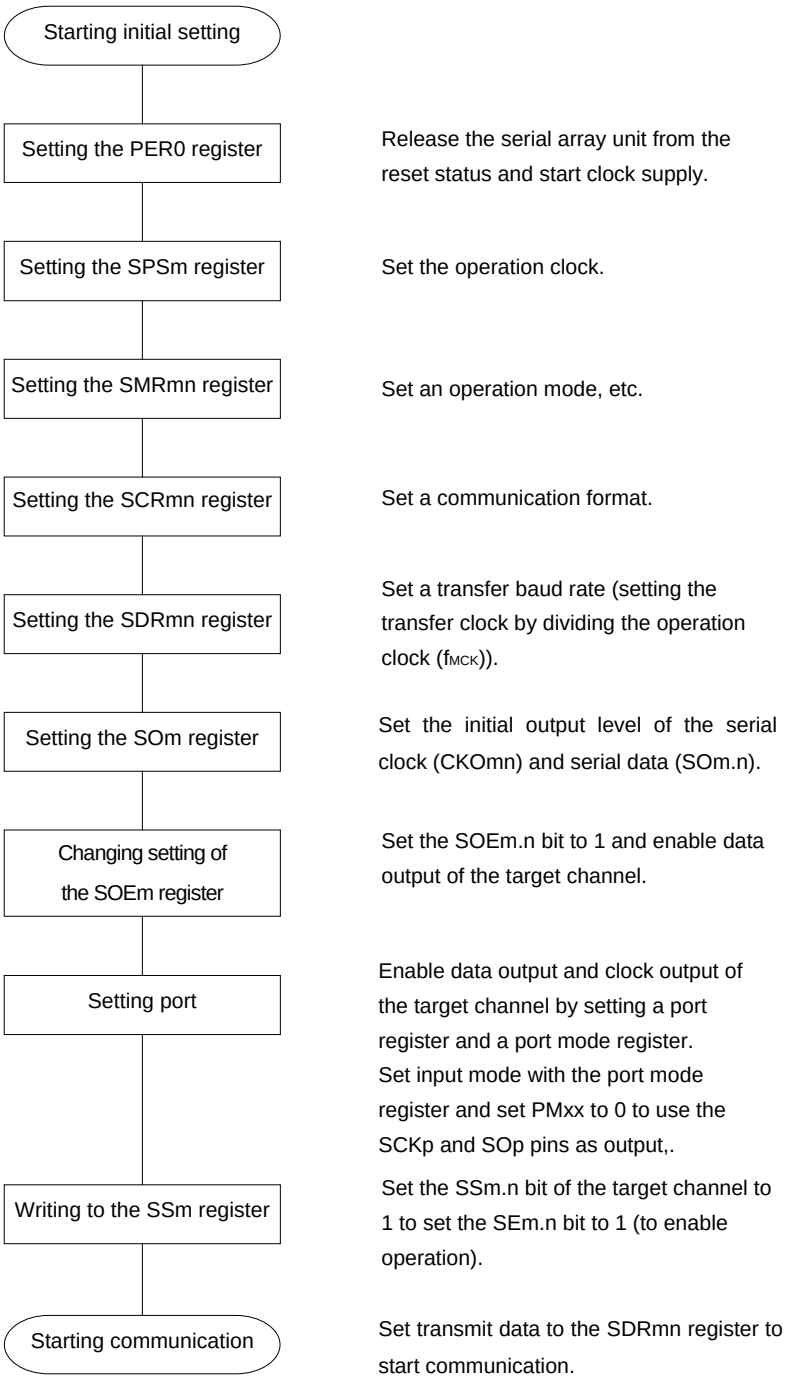
(ii) During operation (SEm.n = 1) (lower 8 bits: SDRmnL)



Remark ☐: 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
n : Channel number (n = 0, 1)

(2) Operation procedure

Figure 12-27. Initial Setting Procedure for Master Transmission

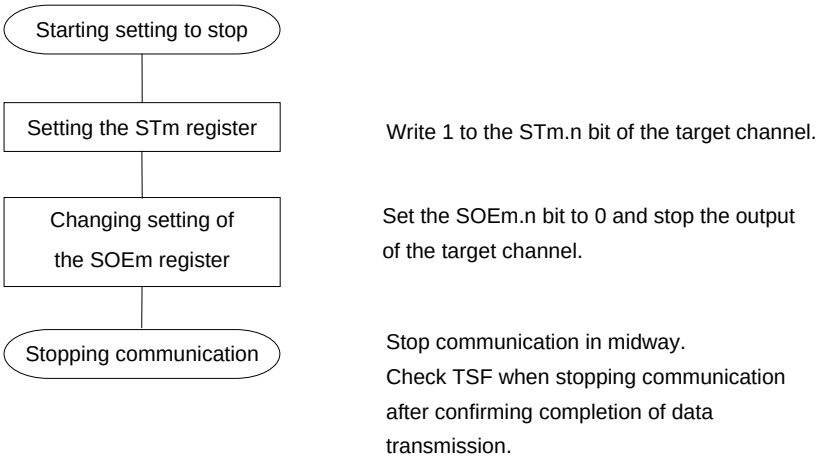


<R>

Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

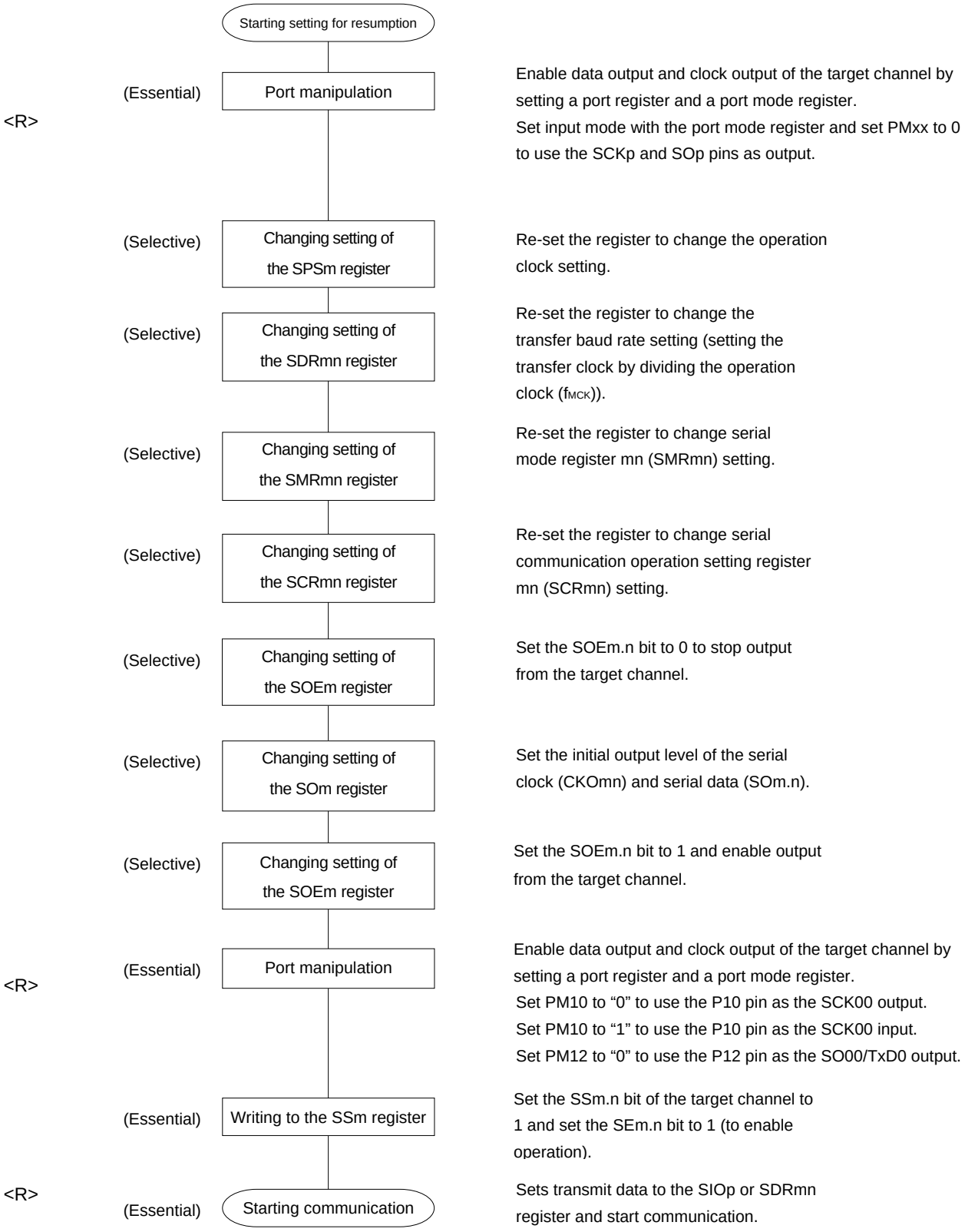
Figure 12-28. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 12-29 Procedure for Resuming Master Transmission**).

2. m: Unit number (m = 0, 1) , n: Channel number (n = 0, 1)

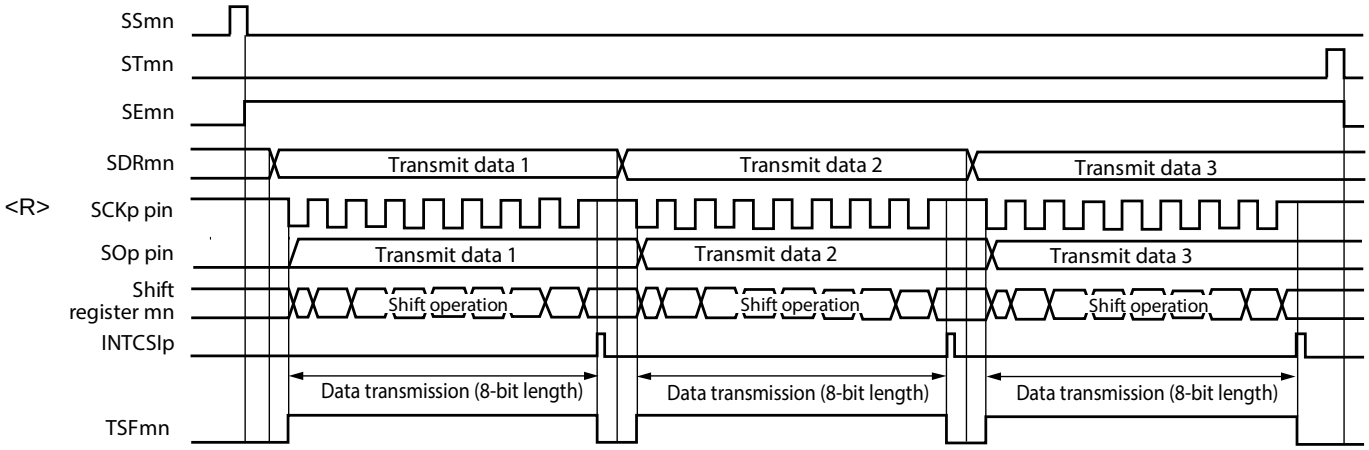
Figure 12-29. Procedure for Resuming Master Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

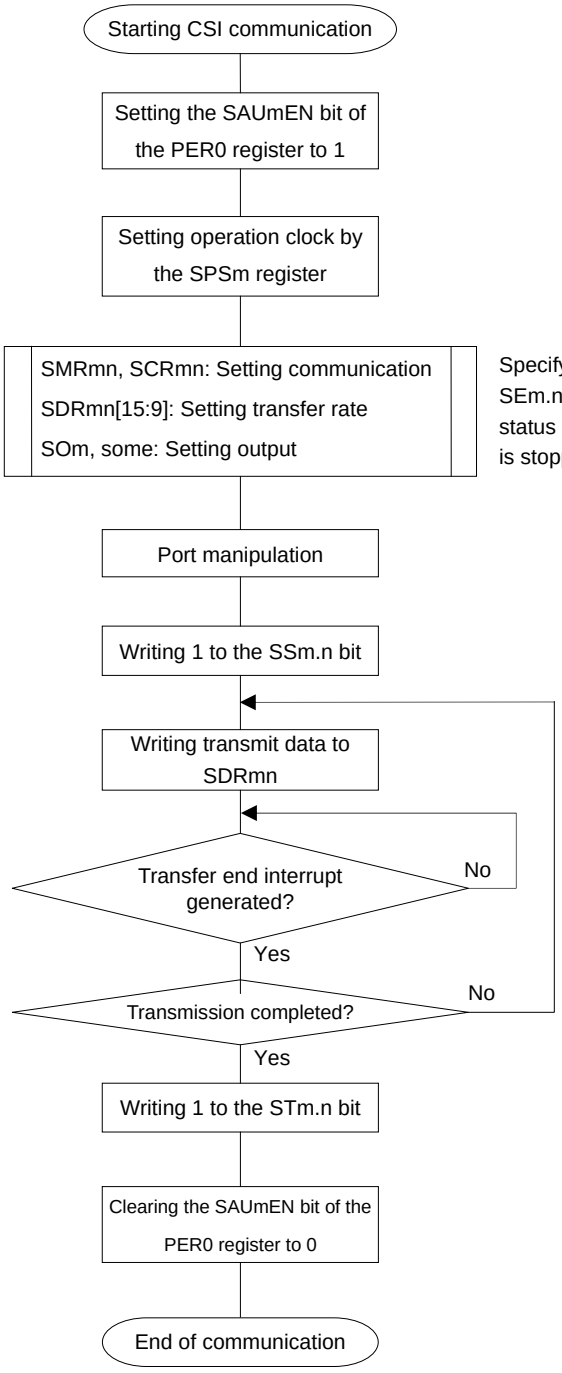
(3) Processing flow (in single-transmission mode)

Figure 12-30. 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), mn = 00, 01, 10

Figure 12-31. Flowchart of Master Transmission (in Single-Transmission Mode)

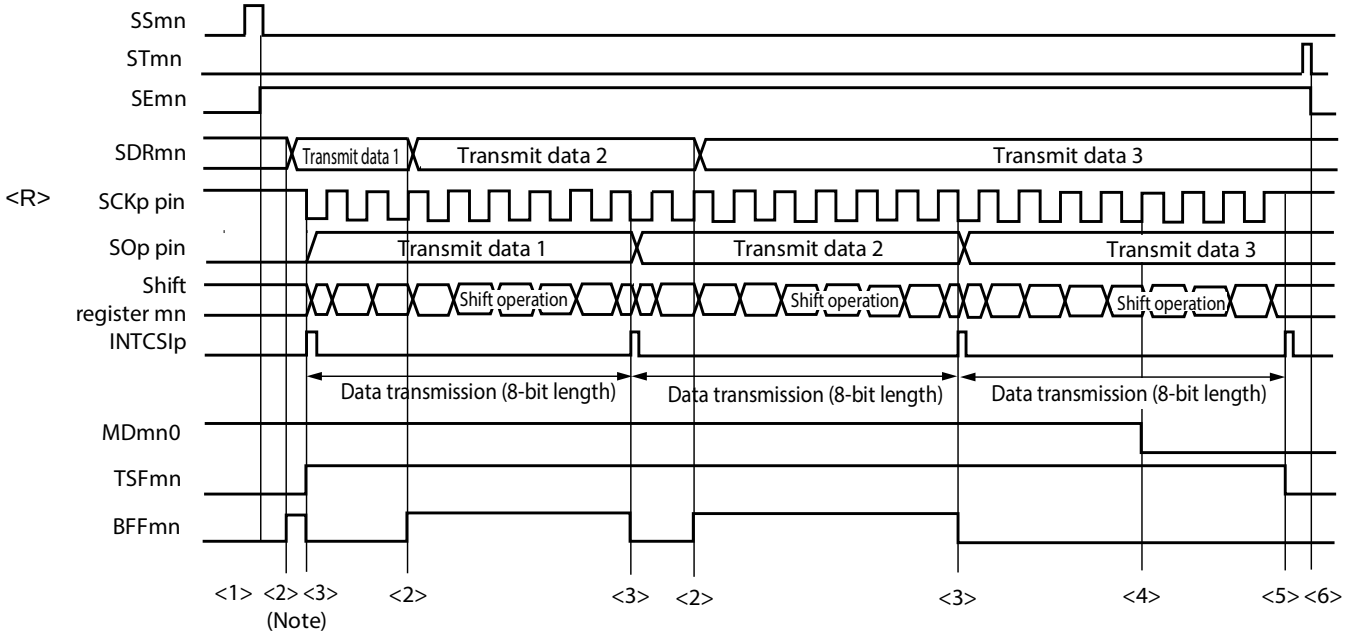


Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing flow (in continuous transmission mode)

Figure 12-32. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

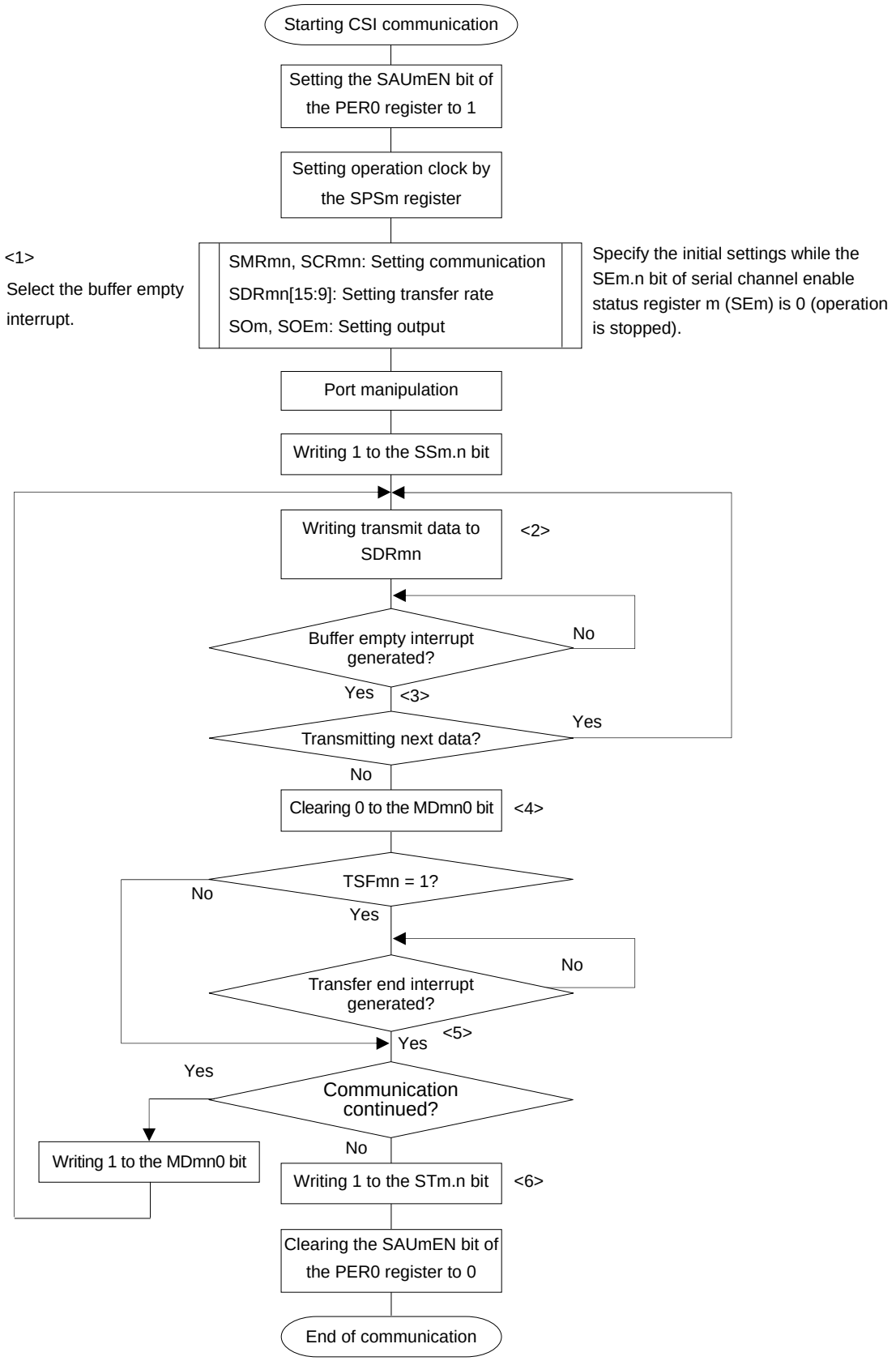


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10), mn = 00, 10, 11

Figure 12-33. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remarks 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-32 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

12.5.2 Master reception

Master reception is an operation wherein the RL78/D1A outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 to 16 bits		
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note**		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the serial clock operation. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Not reversed • CKPmn = 1: Reversed		
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 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10
f_{CLK}: System clock frequency

(1) Register setting

Figure 12-34. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (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	SOm.1 ×	SOm.0 ×

Communication starts when these bits are 1 if the data phase is not reversed (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(b) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm.1 ×	SOEm.0 ×

(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	CCSmn 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
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)

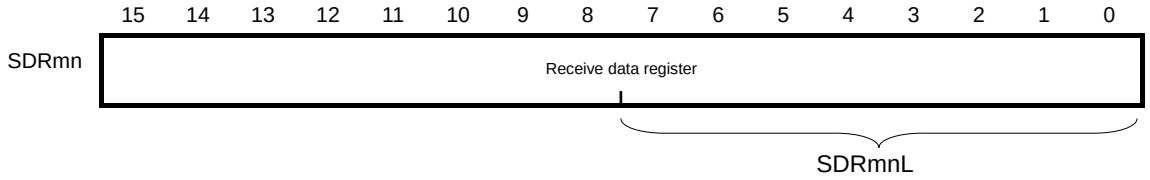
(i) When operation is stopped (SEm.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark □: Setting is fixed in the CSI master reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

Figure 12-34. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (2/2)

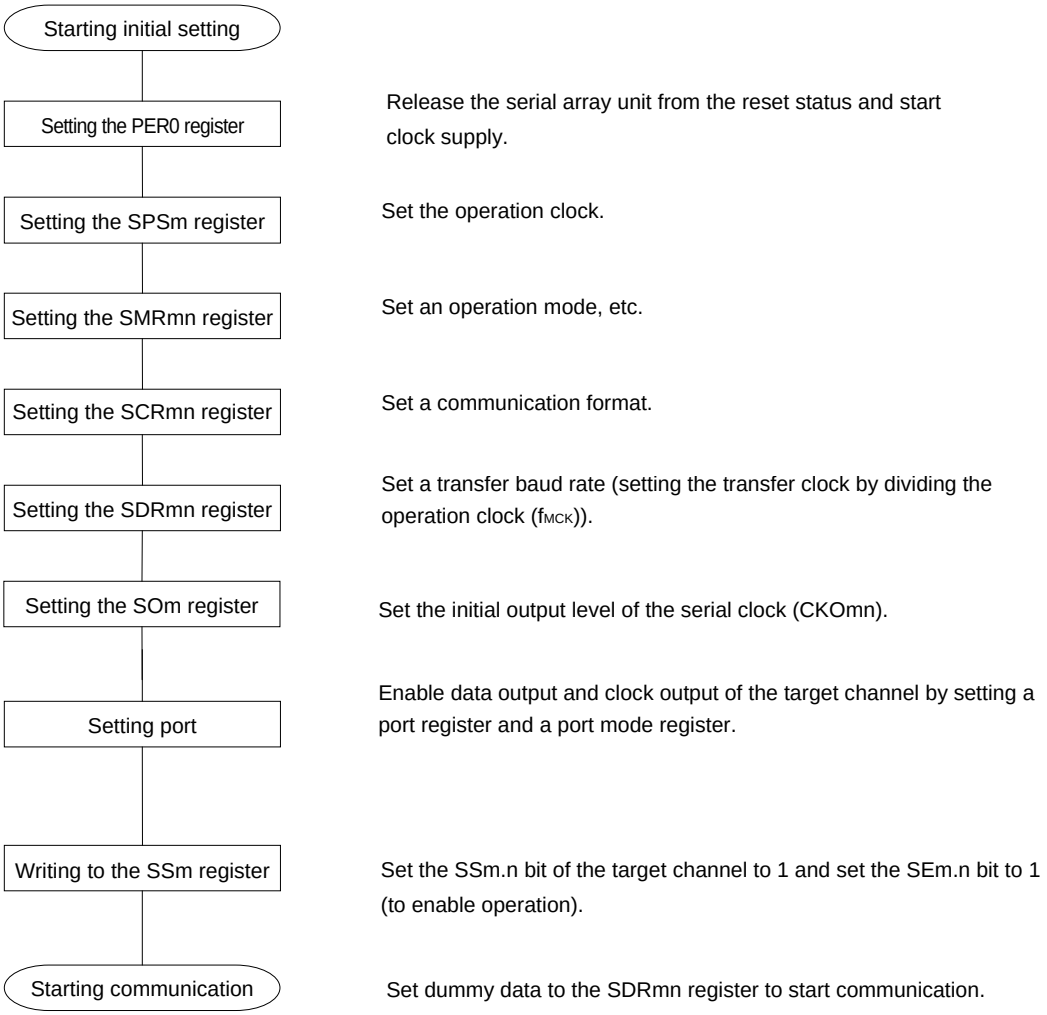
(ii) During operation (SEm.n = 1) (lower 8 bits: SDRmnL)



Remark ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

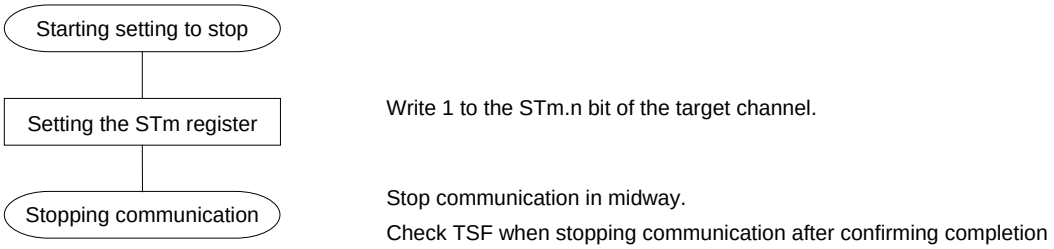
(2) Operation procedure

Figure 12-35. Initial Setting Procedure for Master Reception



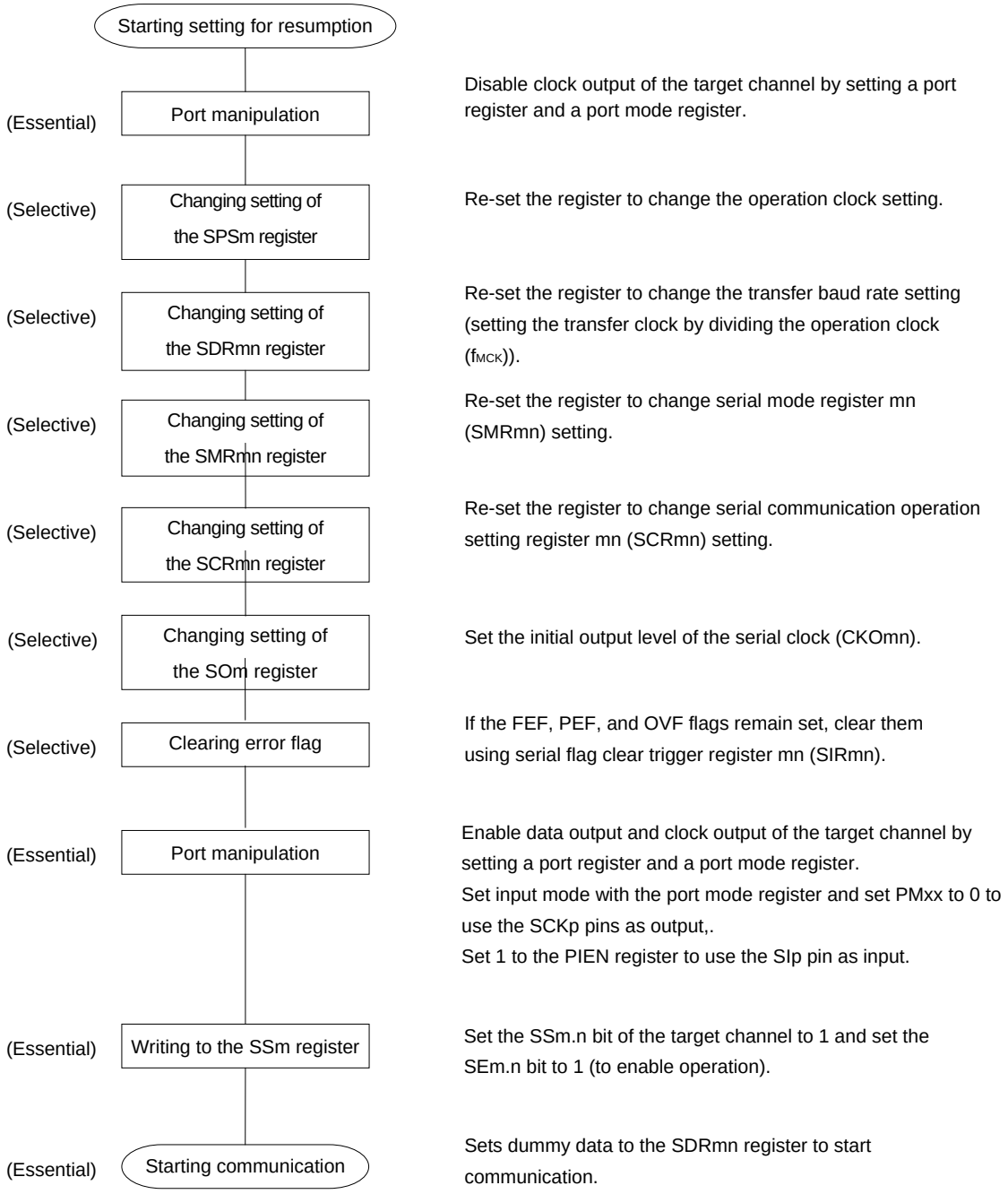
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 12-36. Procedure for Stopping Master Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 12-37 Procedure for Resuming Master Reception**).

Figure 12-37. Procedure for Resuming Master Reception

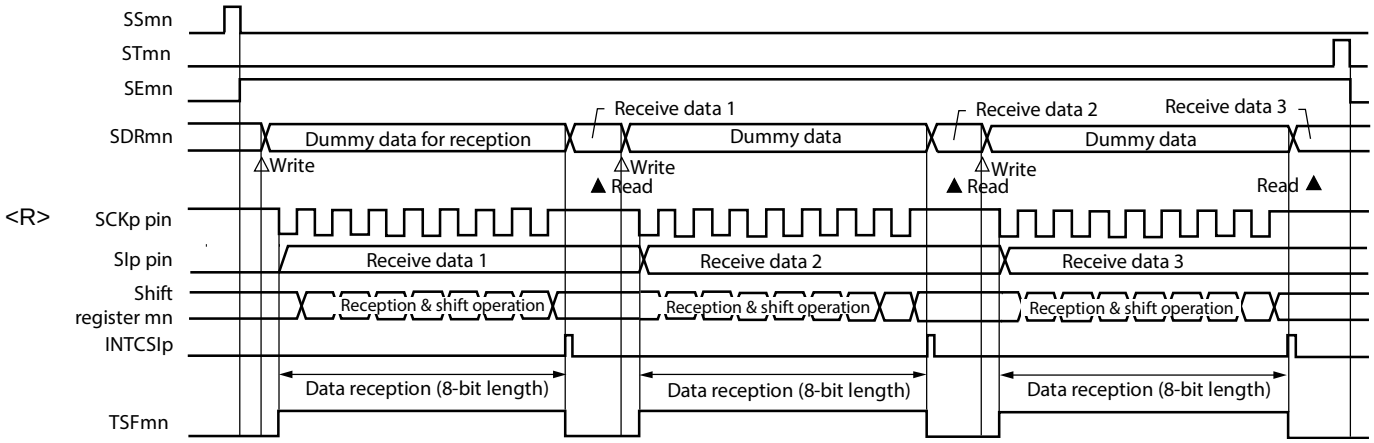


<R>

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

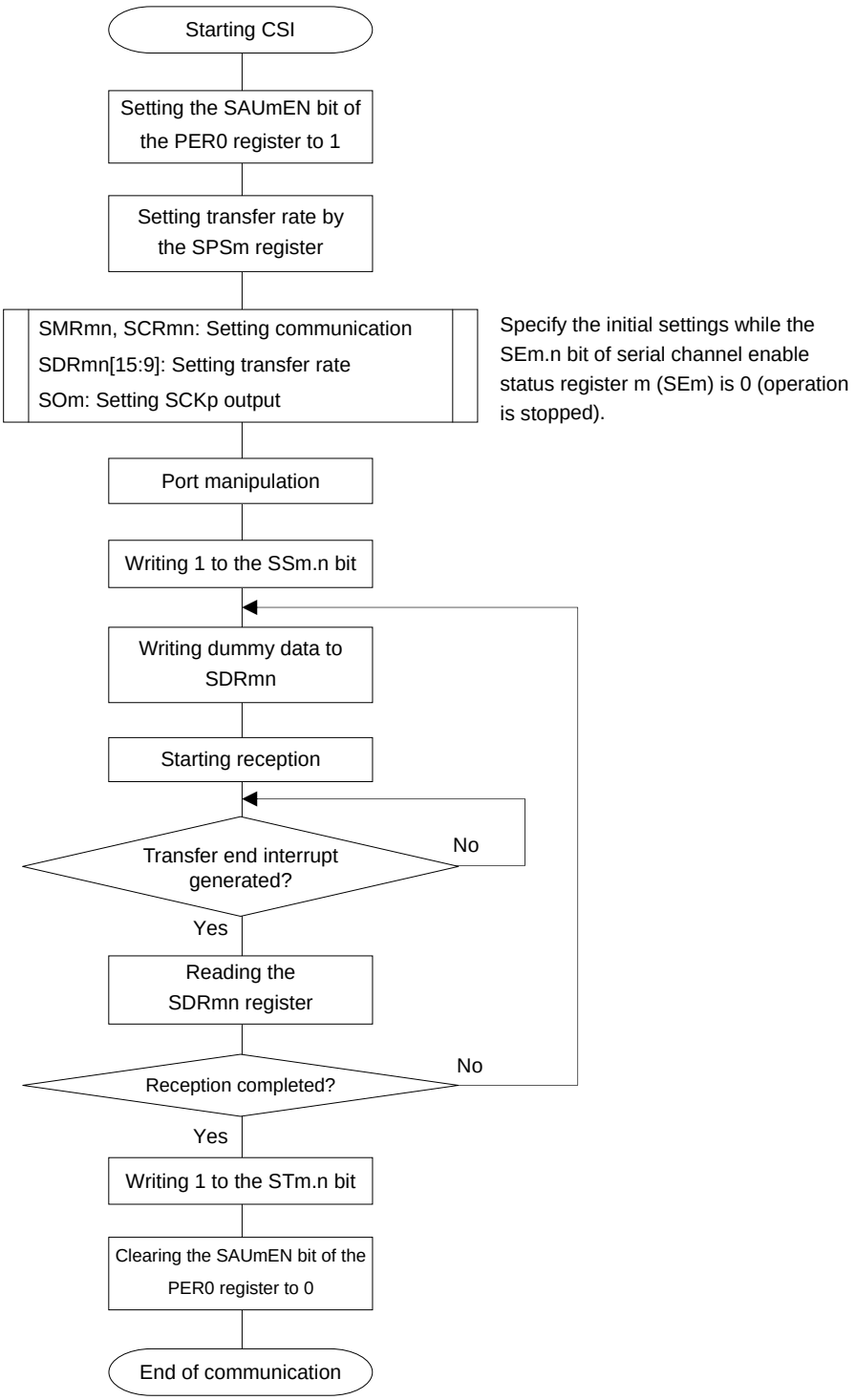
(3) Processing flow (in single-reception mode)

Figure 12-38. 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), mn = 00, 01, 10

Figure 12-39. Flowchart of Master Reception (in Single-Reception Mode)

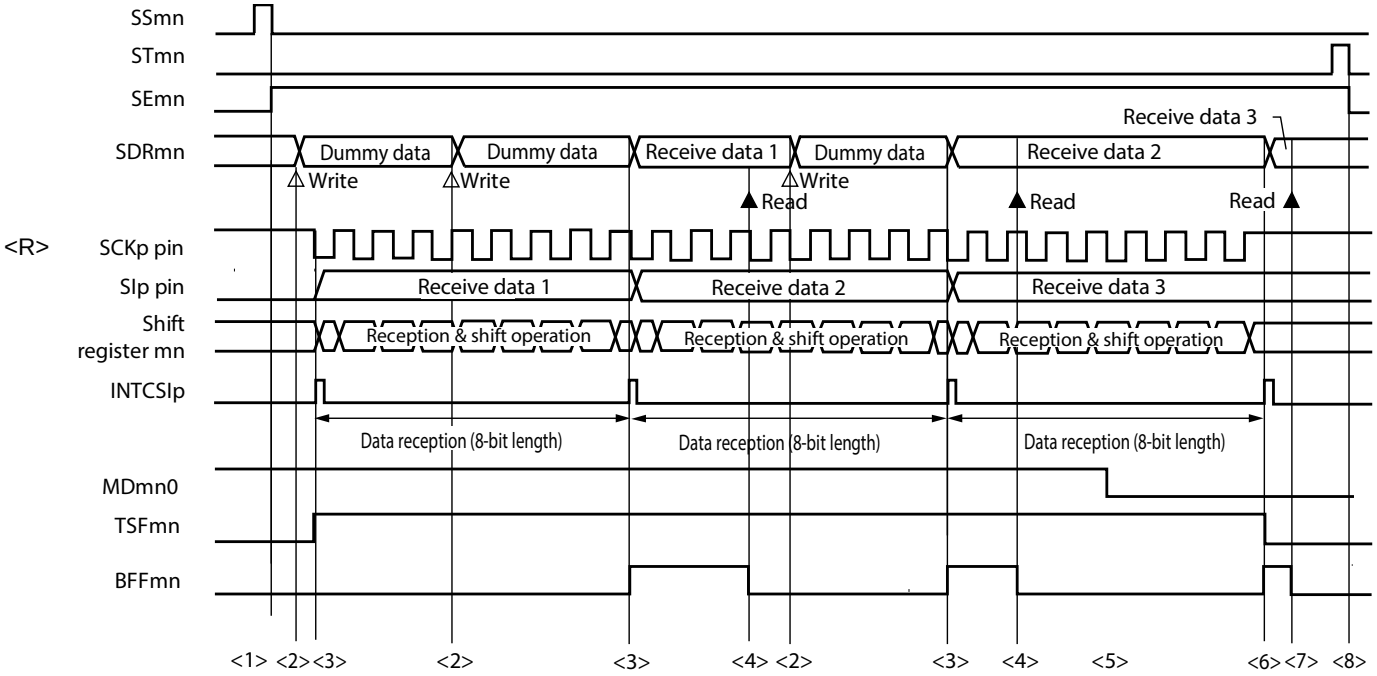


Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing flow (in continuous reception mode)

Figure 12-40. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

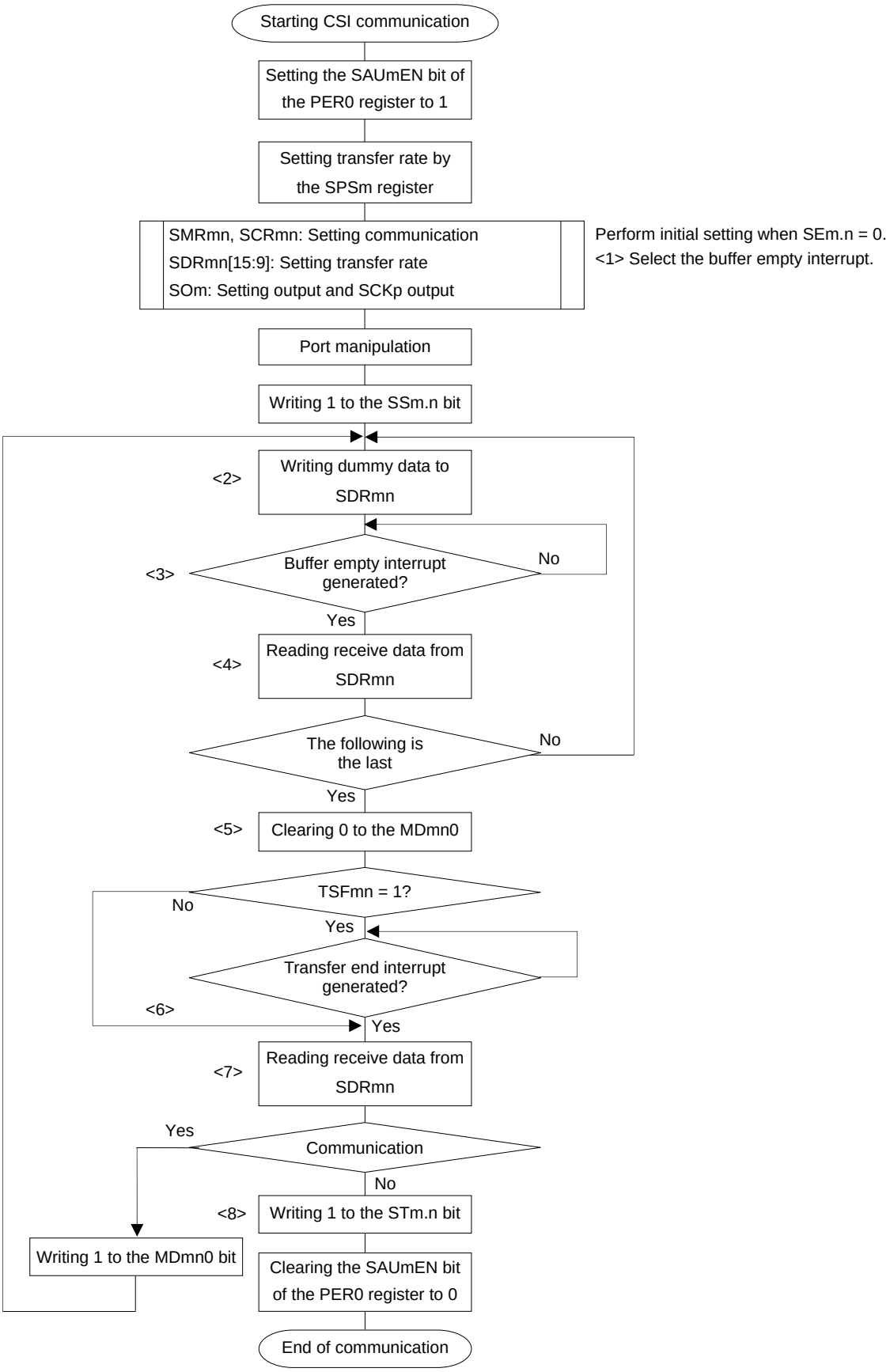


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-41 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), mn = 00, 10, 11

Figure 12-41. Flowchart of Master Reception (in Continuous Reception Mode)



Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-40 Timing Chart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

12.5.3 Master transmission/reception

Master transmission/reception is an operation wherein the RL78/D1A outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 to 16 bits		
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note**		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the serial clock operation. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Not reversed • CKPmn = 1: Reversed		
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 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10
f_{CLK}: System clock frequency

(1) Register setting

Figure 12-42. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (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	SOm.1 0/1	SOm.0 0/1

Communication starts when these bits are 1 if the data phase is not reversed (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
some	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES1 0/1	SOES0 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	SSS.1 0/1	SSS.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	CCSmn 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
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)

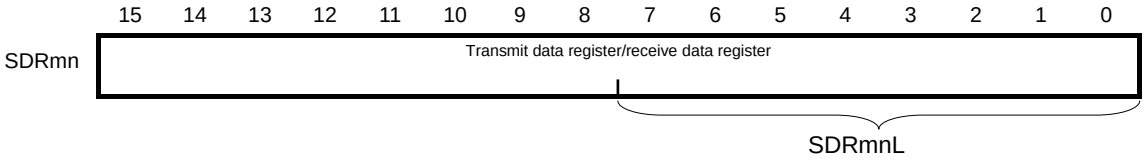
(i) When operation is stopped (SEm.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	0	0	0	0	0	0	0	0

Remark ☐: Setting is fixed in the CSI master transmission/reception mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

Figure 12-42. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (2/2)

(ii) During operation (SEm.n = 1) (lower 8 bits: SDRmnL)



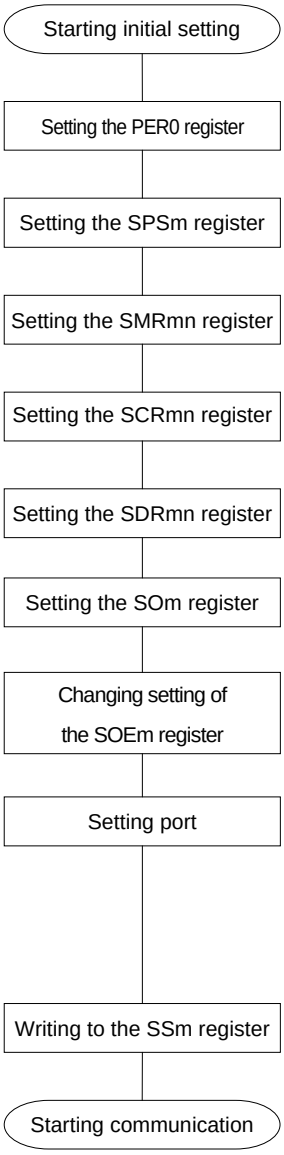
Remark ☐: Setting is fixed in the CSI master transmission/reception mode, ▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

n: Channel number (n = 0, 1)

(2) Operation procedure

Figure 12-43. 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 (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOm.n).

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.

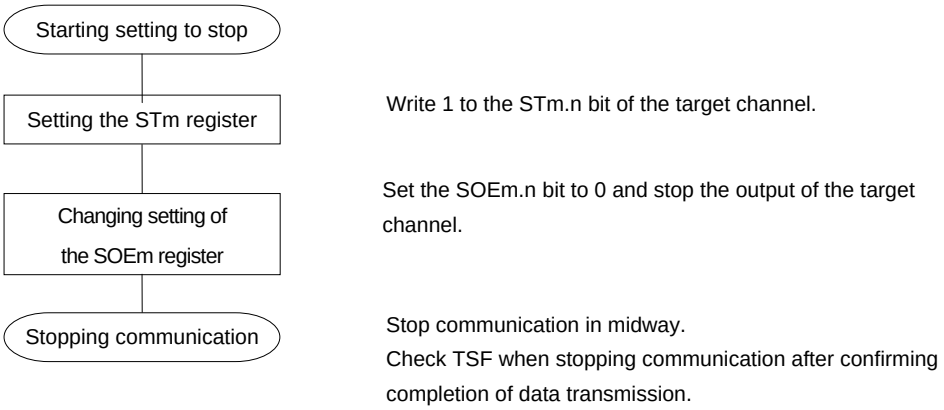
<R> Set input mode with the port mode register and set PMxx to 0 to use the SOp and SCKp pins as output.
Set 1 to the PMXx register to use the SCKp and SIp pins as input.

Set the SSm.n bit of the target channel to 1 and set the SEm.n bit to 1 (to enable operation).

Set transmit data to the SDRmn register and start communication.

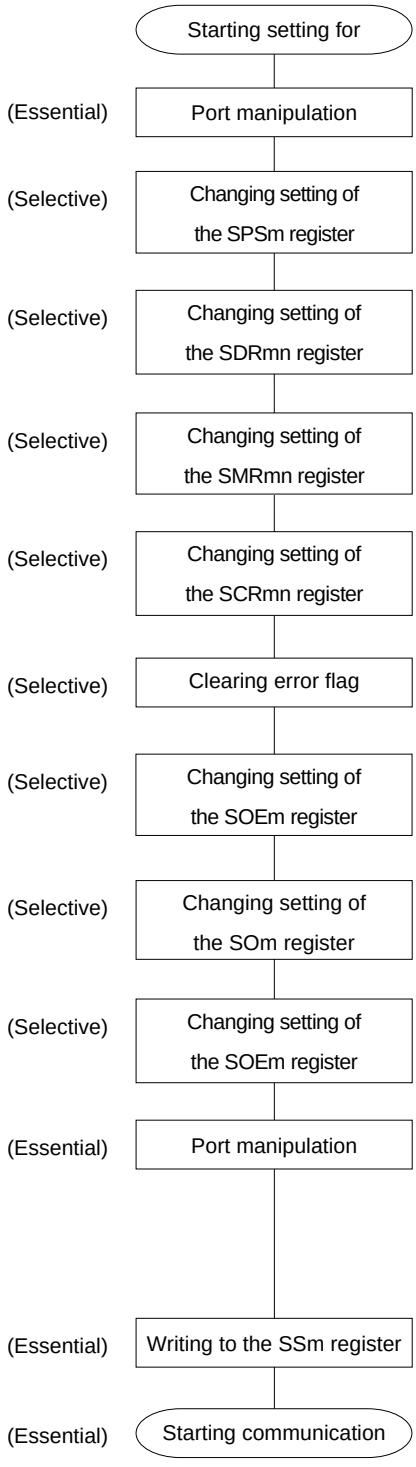
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 12-44. Procedure for Stopping Master Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 12-45 Procedure for Resuming Master Transmission/ Reception**).

Figure 12-45. Procedure for Resuming Master Transmission/Reception



Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).

Set the SOEm.n bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOm.n).

Set the SOEm.n bit to 1 and enable output from the target channel.

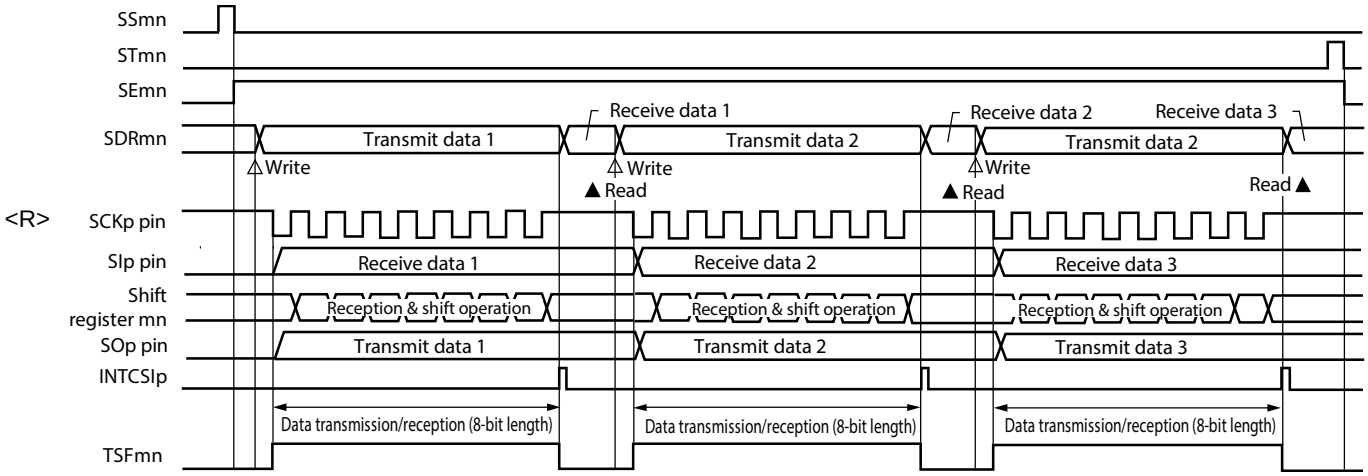
<R> Enable data output and clock output of the target channel by setting a port register and a port mode register.
Set input mode with the port mode register and set PMxx to 0 to use the SOp and SCKp pins as output.
Set 1 to the PMxx register to use the SCKp and SIp pins as input.

Set the SSm.n bit of the target channel to 1 and set the SEm.n bit to 1 (to enable operation).

Sets transmit data to the SDRmn register and start communication.

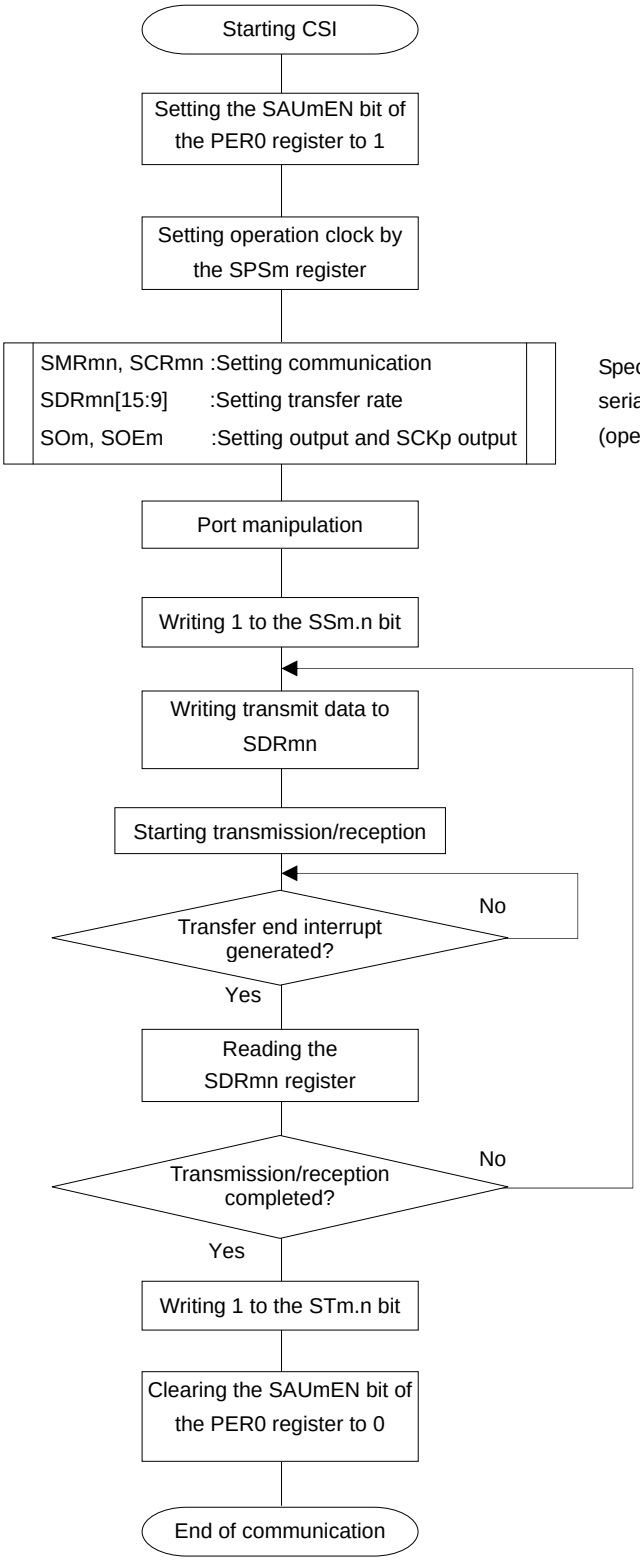
(3) Processing flow (in single-transmission/reception mode)

Figure 12-46. 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), mn = 00, 01, 10

Figure 12-47. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)

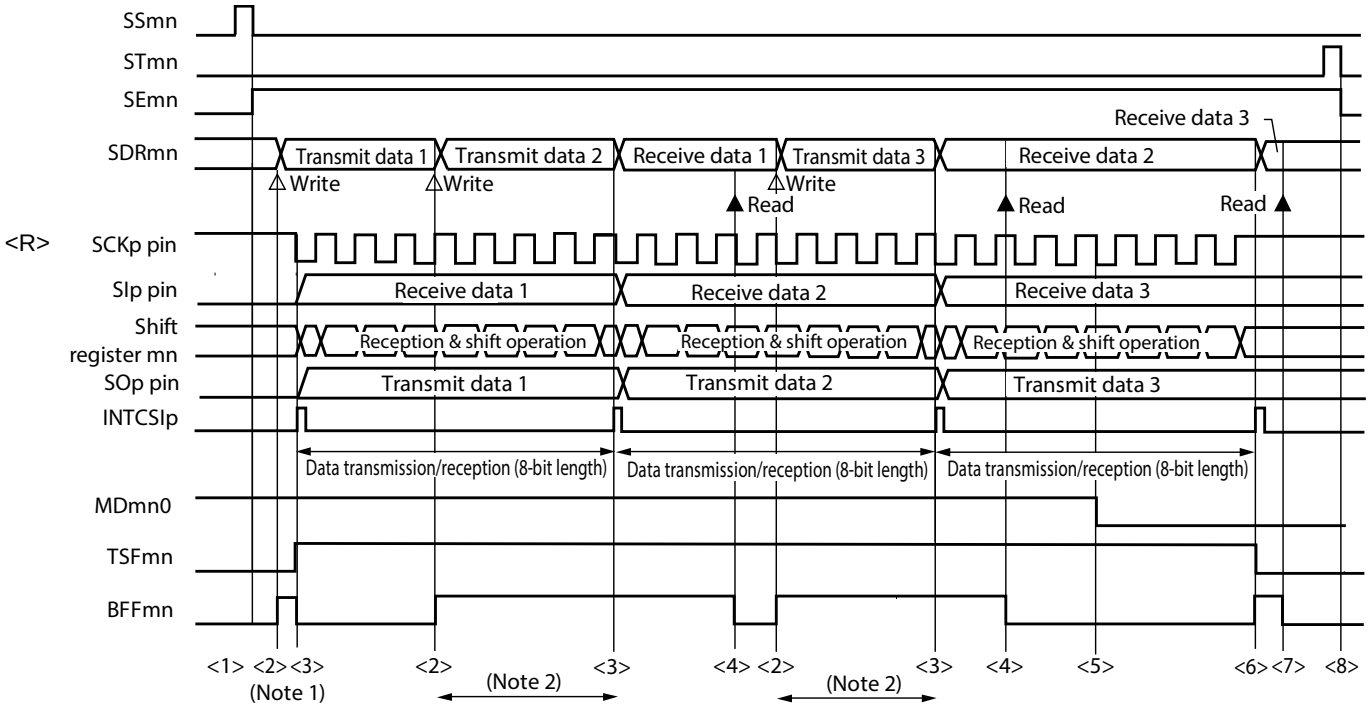


Specify the initial settings while the SEm.n bit of the serial channel enable status register m (SEm) is 0 (operation is stopped).

Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-48. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

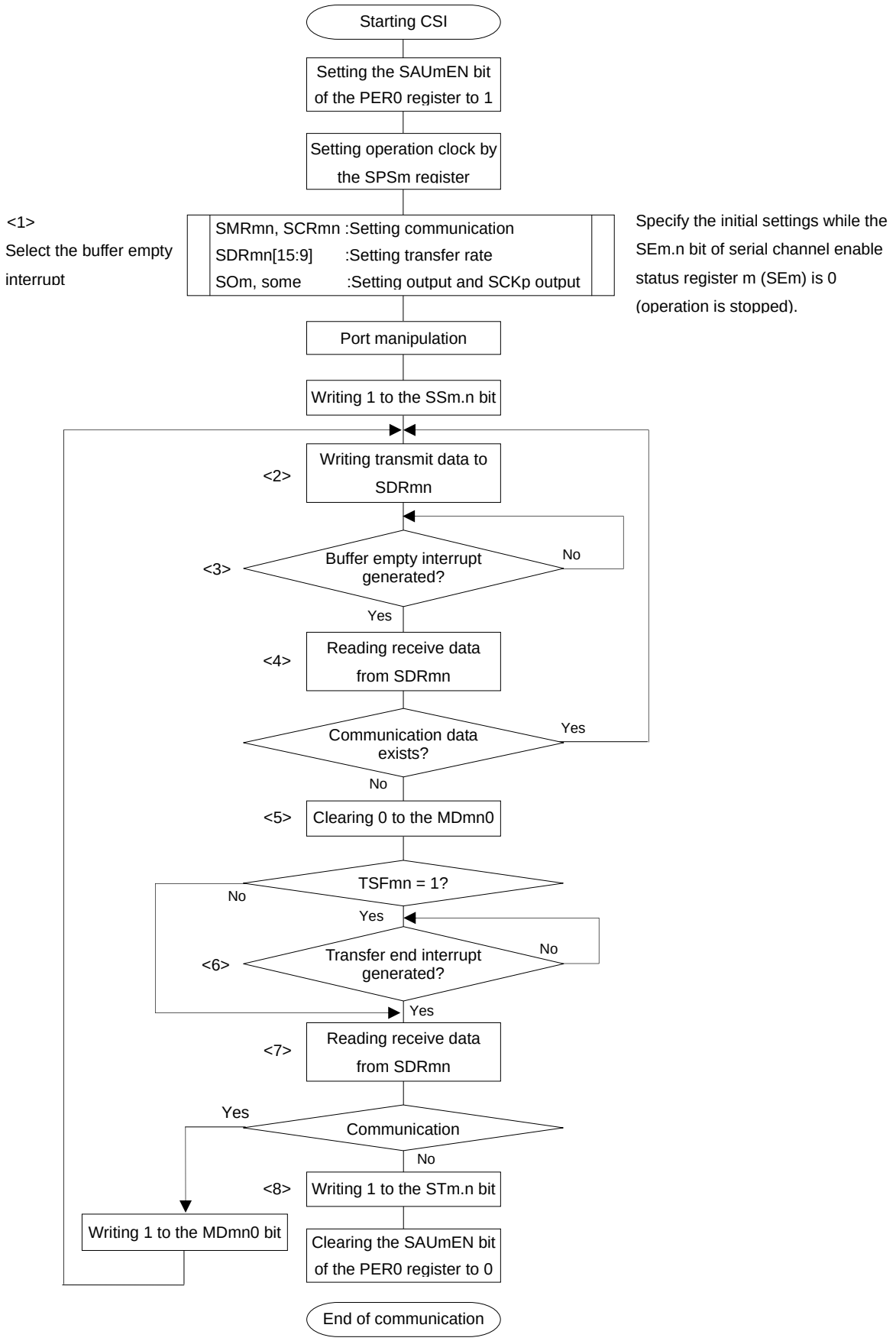
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-49 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), mn = 00, 10, 11

Figure 12-49. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.4 Slave transmission

Slave transmission is that the RL78/D1A 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
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 to 16 bits		
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2.		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the serial clock operation. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Not reversed • CKPmn = 1: Reversed		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to the SCK00, SCK01, and SCK10 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz]. Set the SPSm register so that f_{MCK}/6 [Hz] equals f_{SCK}/2 or more that is set with the SDRm register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10
f_{CLK}: System clock frequency

(1) Register setting

Figure 12-50. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (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	SOm.1 0/1	SOm.0 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	CCSmn 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
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)

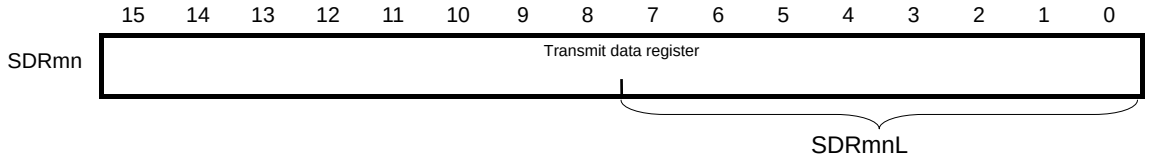
(i) When operation is stopped (SEm.n = 0)

	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 ☐: Setting is fixed in the CSI slave transmission mode, ▒ Setting disabled (set to the initial value)
× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

Figure 12-50. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (2/2)

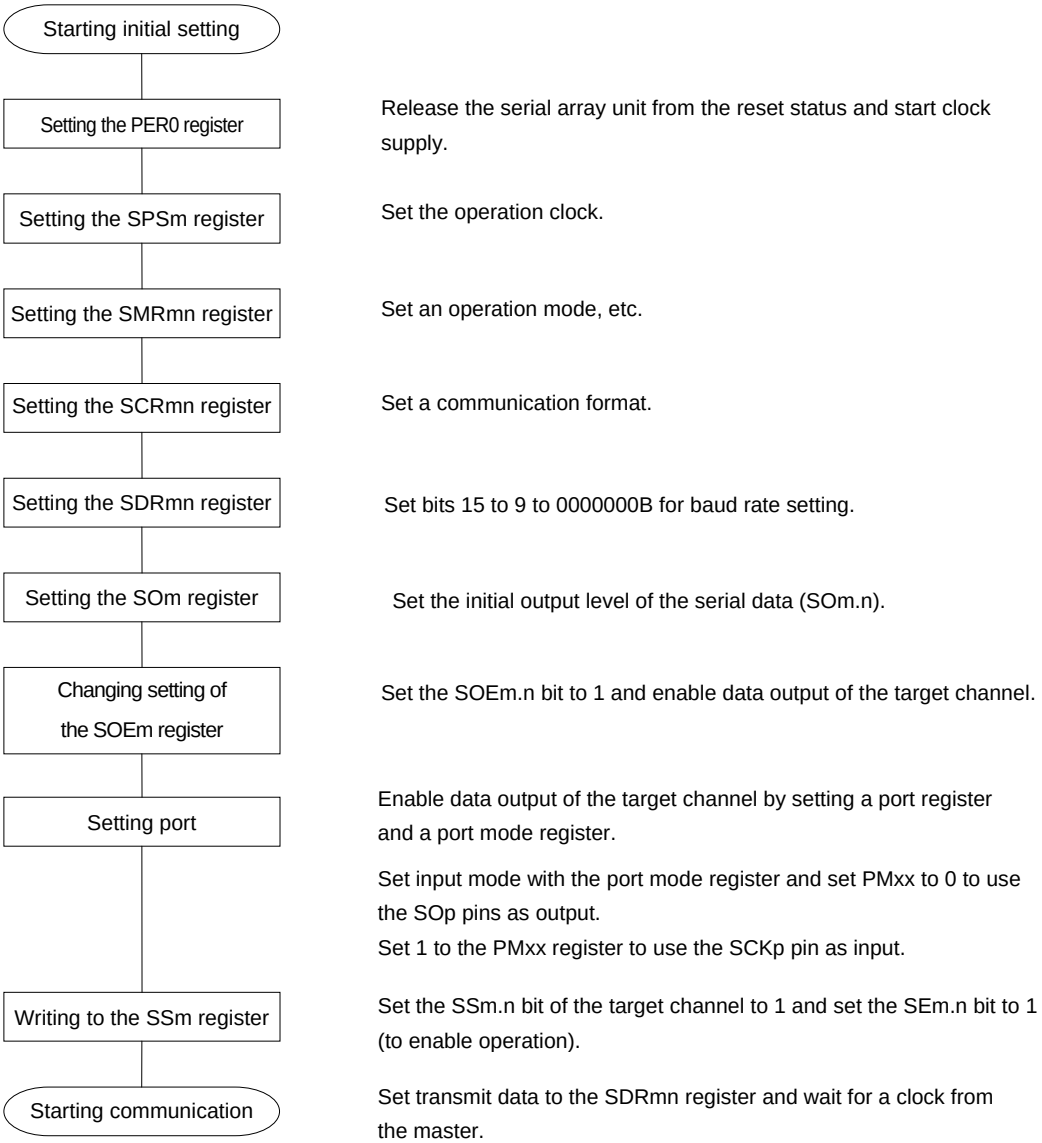
(ii) During operation (SEm.n = 1) (lower 8 bits: SDRmnL)



Remark ☐: Setting is fixed in the CSI slave transmission mode, ▒ Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

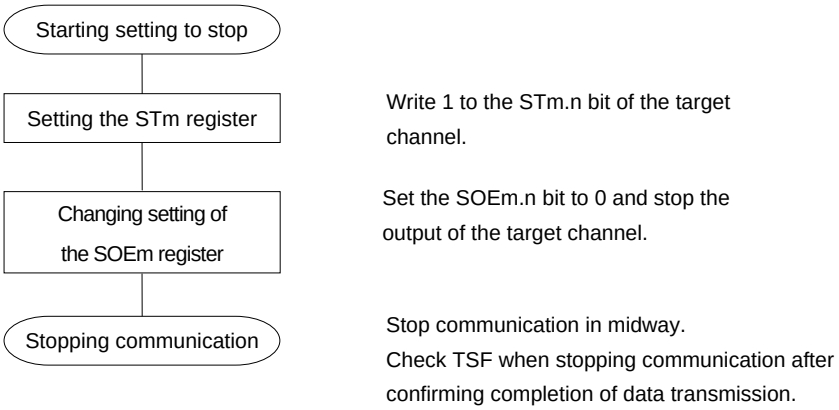
(2) Operation procedure

Figure 12-51. Initial Setting Procedure for Slave Transmission



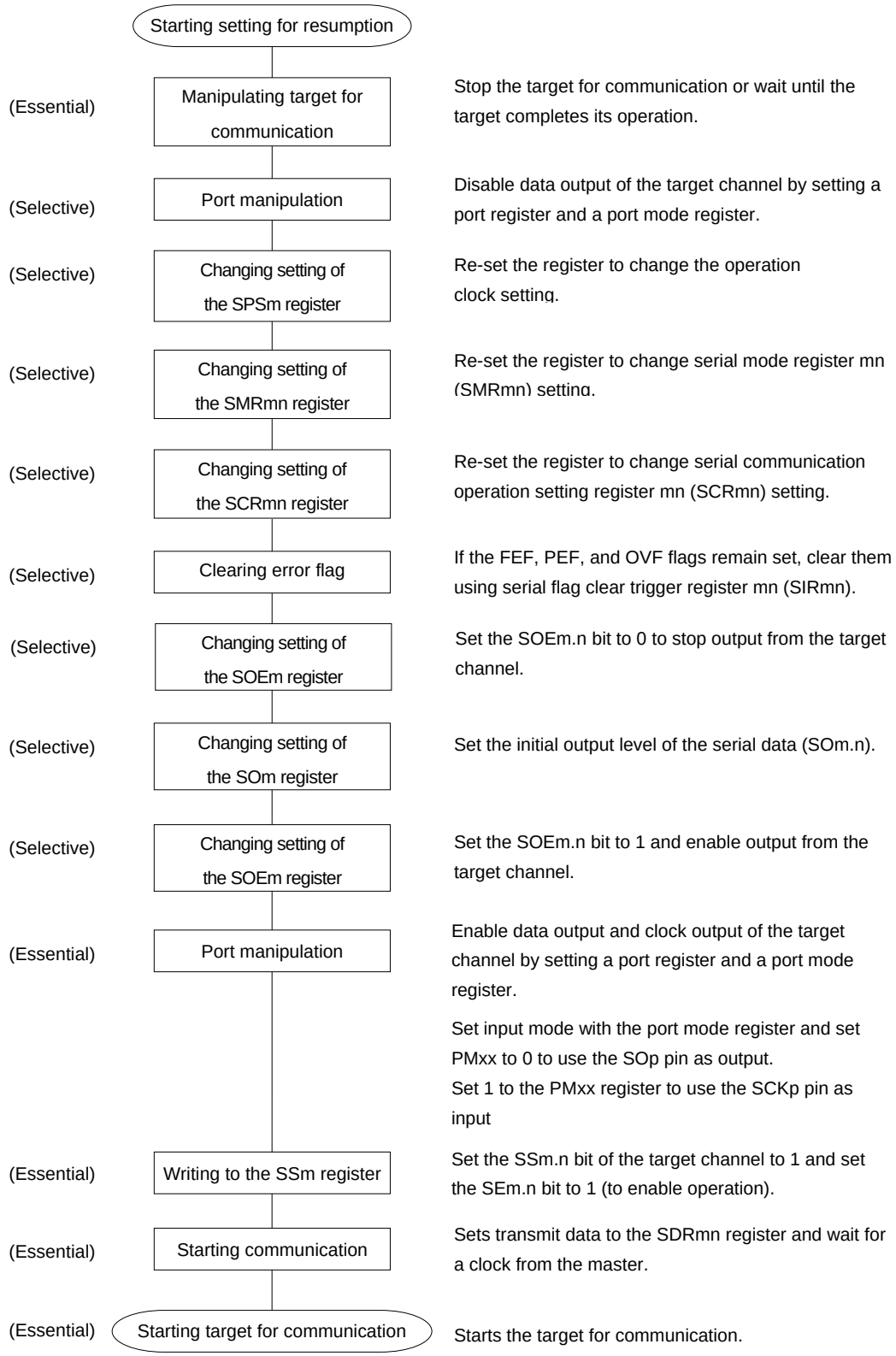
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 12-52. Procedure for Stopping Slave Transmission



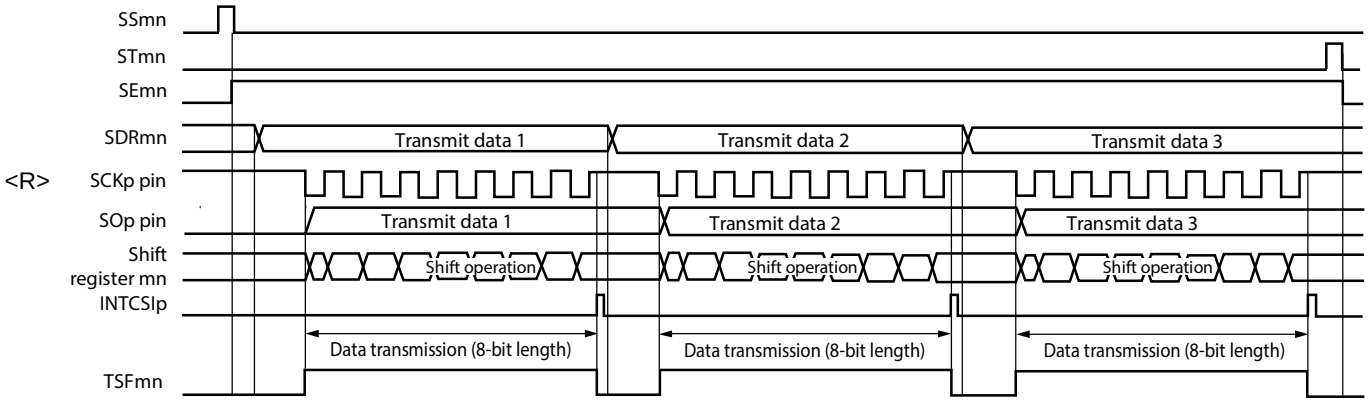
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 12-53. Procedure for Resuming Slave Transmission**).

Figure 12-53. Procedure for Resuming Slave Transmission



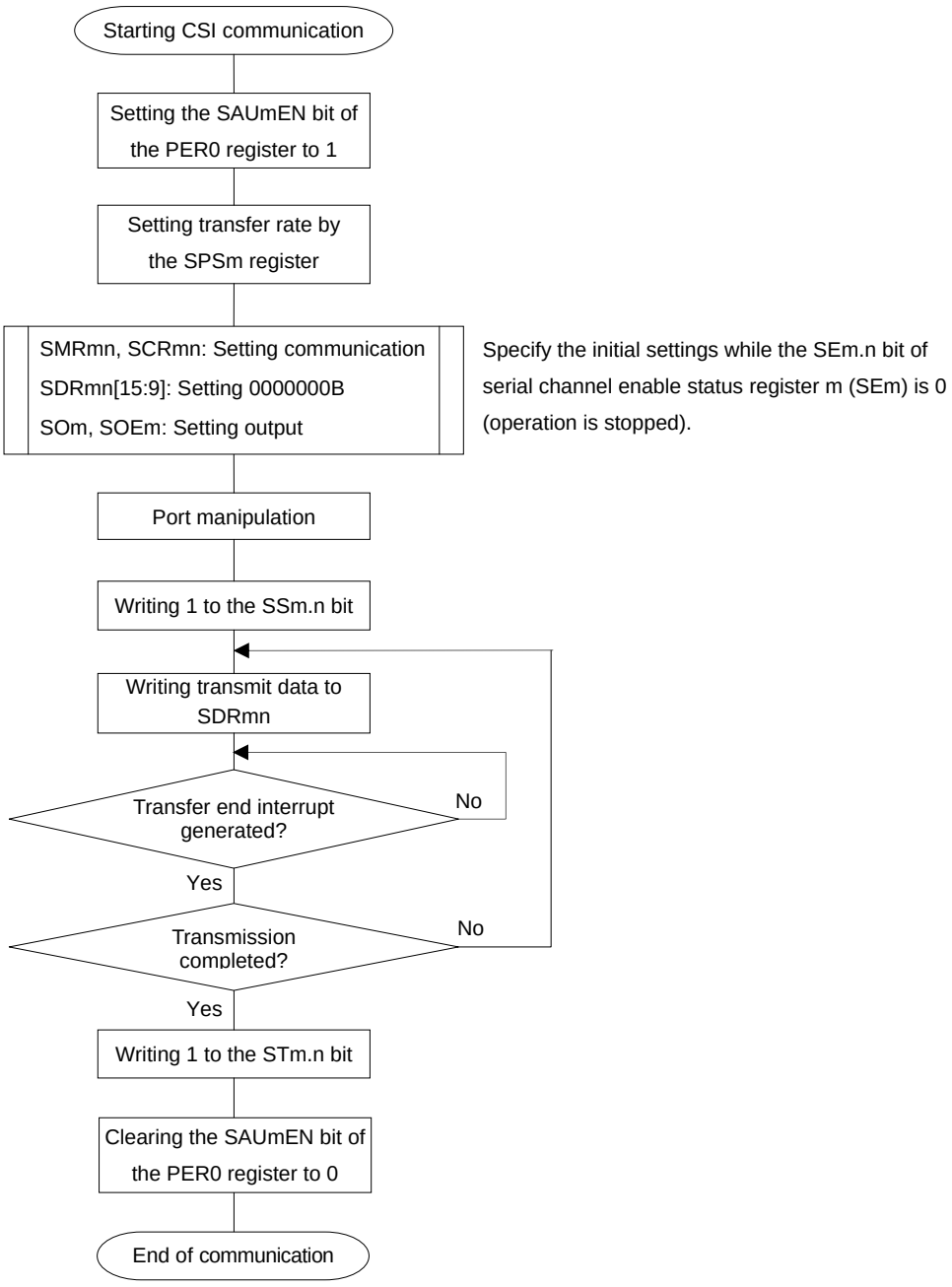
(3) Processing flow (in single-transmission mode)

Figure 12-54. 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), mn = 00, 01, 10

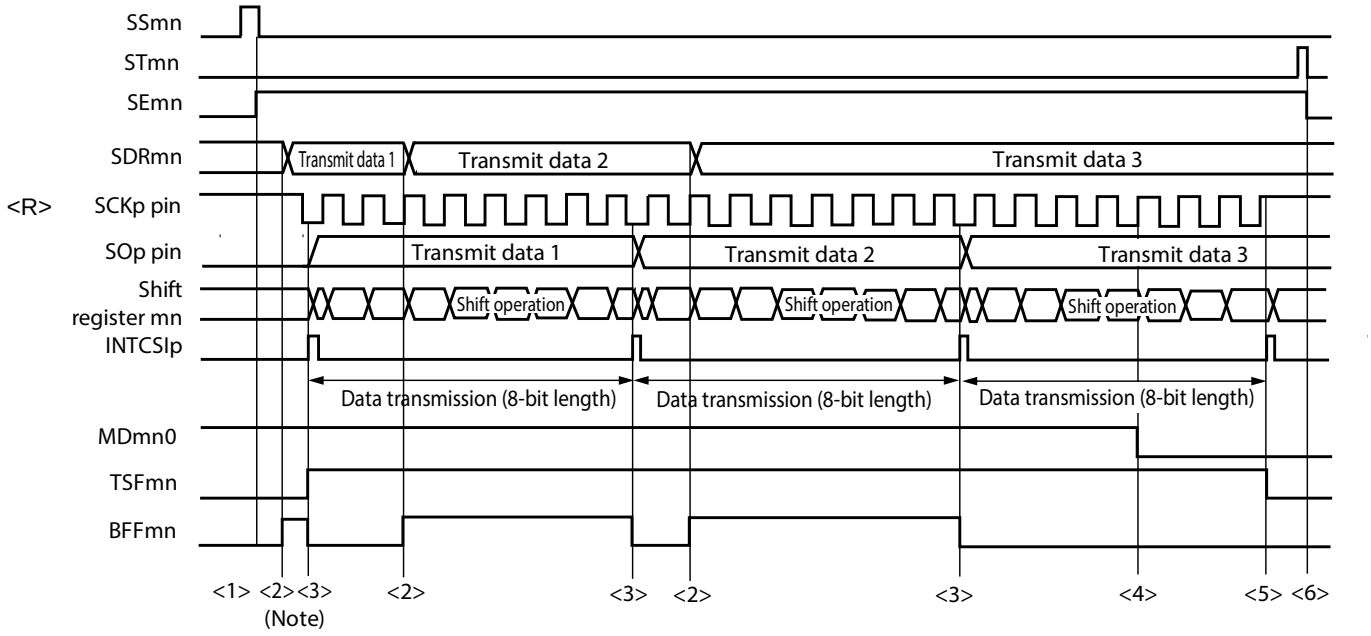
Figure 12-55. Flowchart of Slave Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 12-56. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

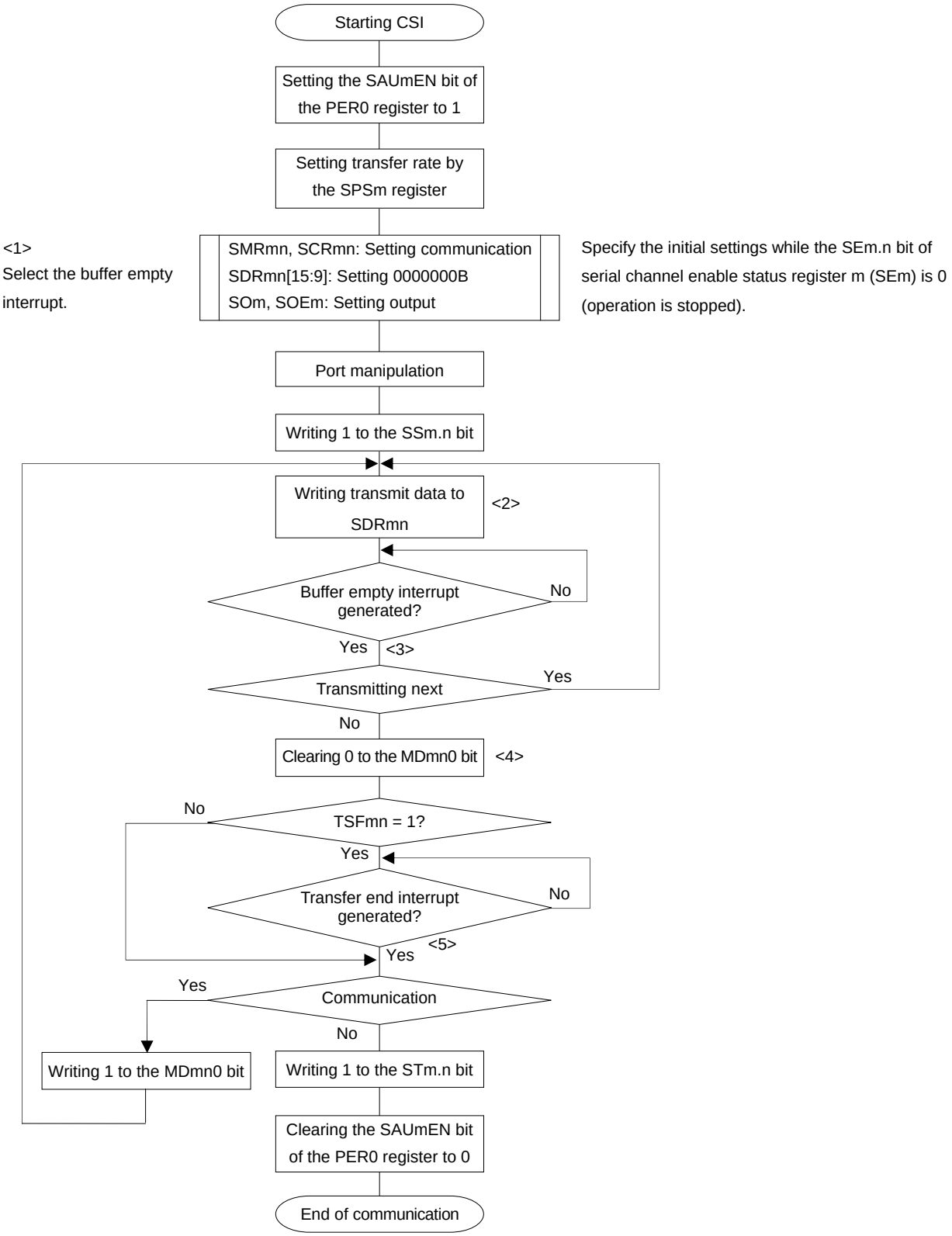


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10), mn = 00, 10, 11

Figure 12-57. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-56 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

12.5.5 Slave reception

Slave reception is that the RL78/D1A 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
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 to 16 bits		
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the serial clock operation. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Not reversed • CKPmn = 1: Reversed		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to the SCK00, SCK01, and SCK10 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz]. Set the SPSm register so that f_{MCK}/6 [Hz] equals f_{SCK} or more that is set with the SDRm register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10

(1) Register setting

Figure 12-58. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10)

(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	SOm.1 ×	SOm.0 ×

(b) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm.1 ×	SOEm.0 ×

(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	CCSmn 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
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)

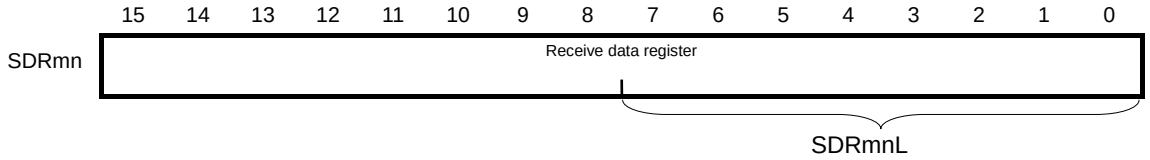
(i) When operation is stopped (SEm.n = 0)

	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 ☐: Setting is fixed in the CSI slave reception mode, ▒ : Setting disabled (set to the initial value)
× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

Figure 12-58. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (2/2)

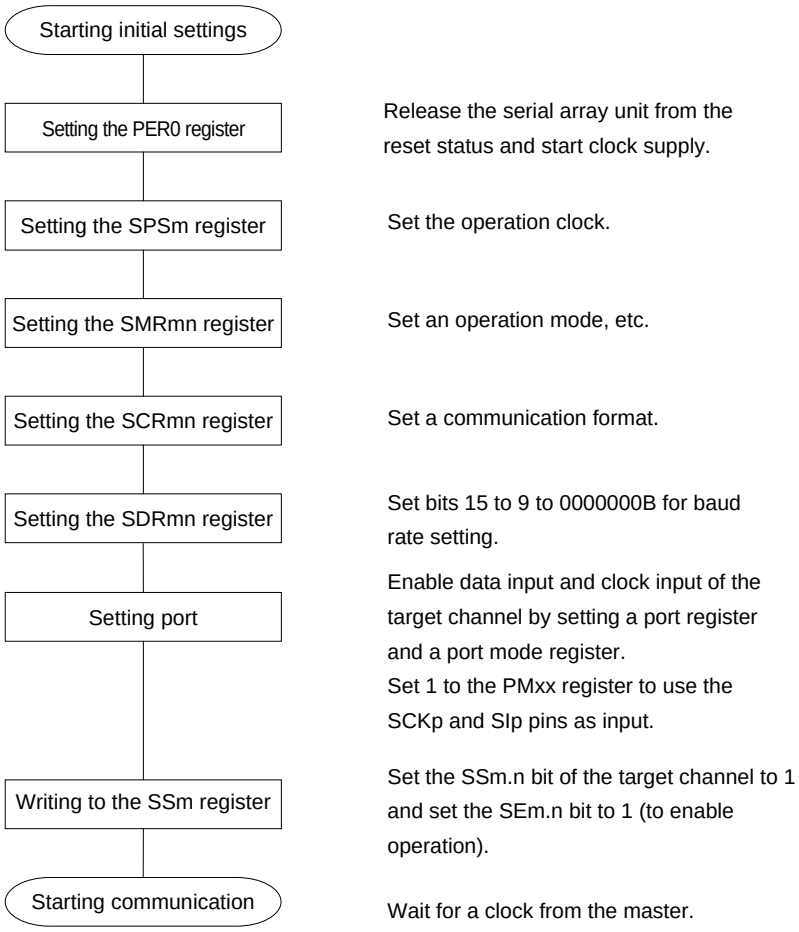
(ii) During operation (SEm.n = 1) (lower 8 bits: SDRmnL)



Remark ☐: Setting is fixed in the CSI slave reception mode, ▒ : Setting disabled (set to the initial value)
× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

(2) Operation procedure

Figure 12-59. Initial Setting Procedure for Slave Reception



<R>

Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 12-60. Procedure for Stopping Slave Reception

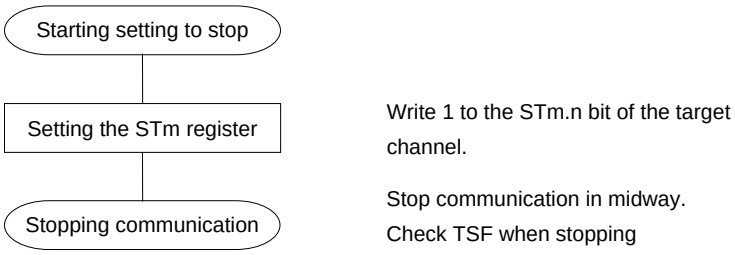
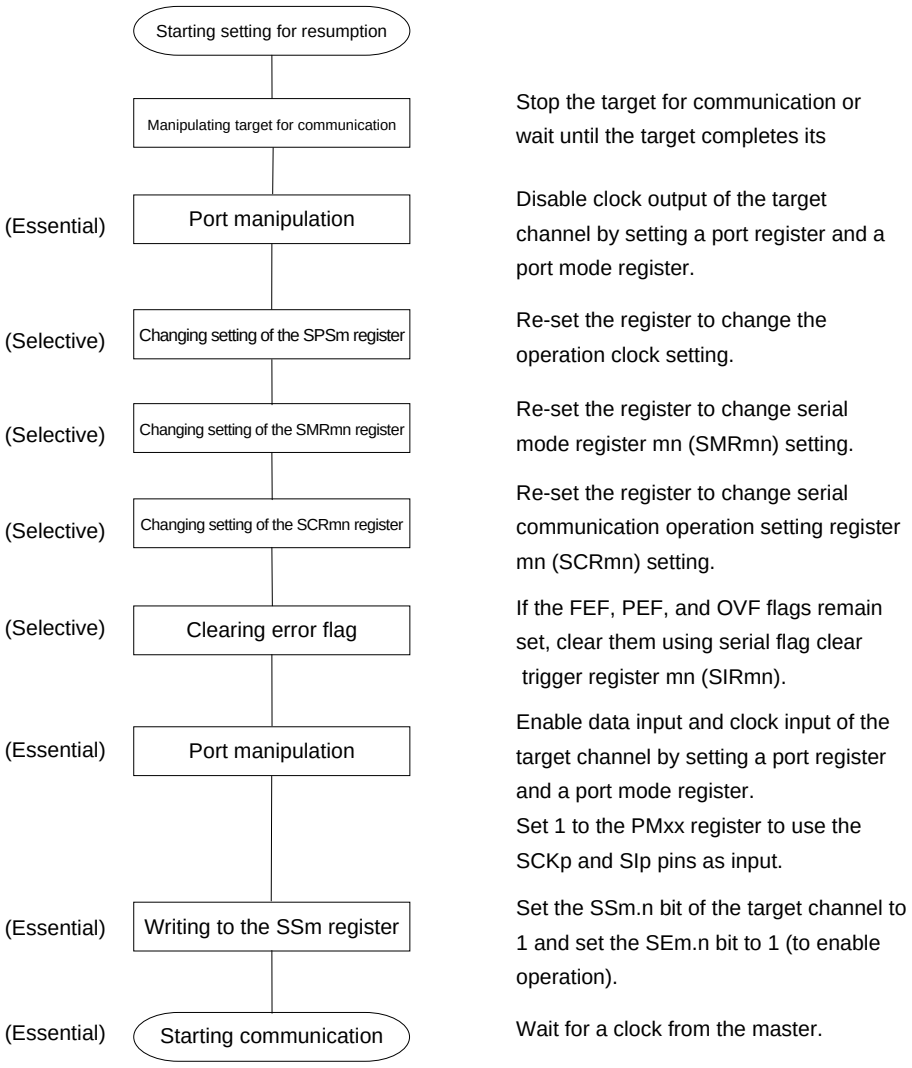


Figure 12-61. Procedure for Resuming Slave Reception

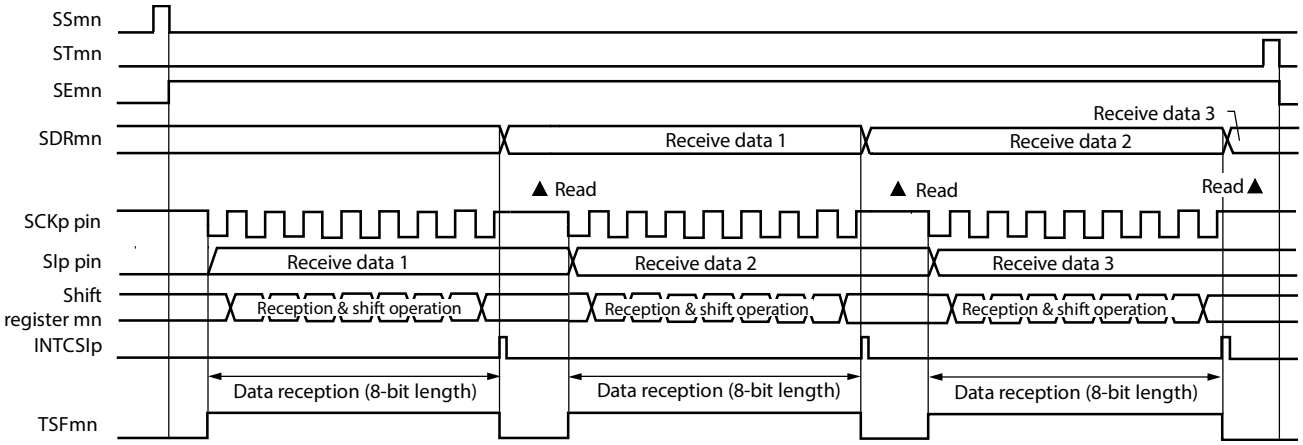
<R>



(3) Processing flow (in single-reception mode)

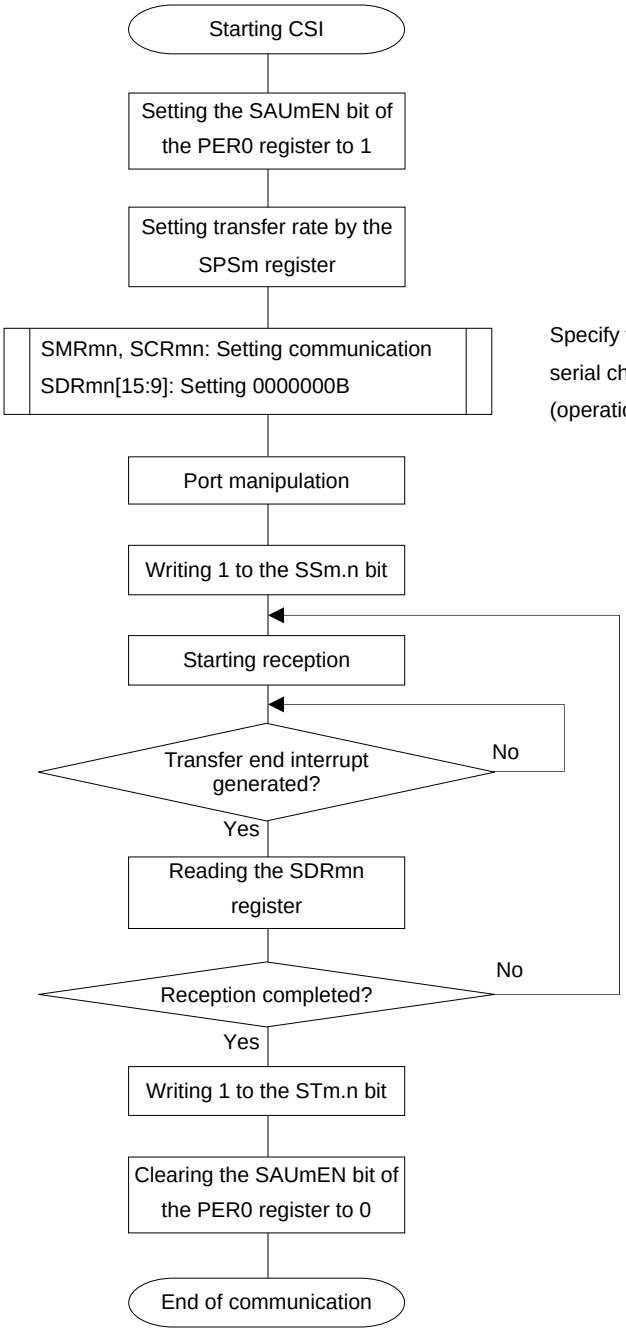
Figure 12-62. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
p: CSI number (p = 00, 01, 10), mn = 00, 01, 10

Figure 12-63. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

12.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78/D1A 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
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 to 16 bits		
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2.		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the serial clock operation. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Not reversed • CKPmn = 1: Reversed		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to the SCK00, SCK01, and SCK10 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz]. Set the SPSm register so that f_{MCK}/6 [Hz] equals f_{SCK} or more that is set with the SDRm register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10

(1) Register setting

Figure 12-64. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (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	SOm.1 0/1	SOm.0 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	CCSmn 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
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)

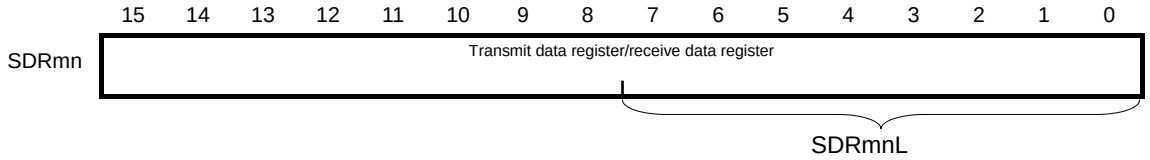
(i) When operation is stopped (SEm.n = 0)

	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 ☐: Setting is fixed in the CSI slave transmission/reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

Figure 12-64. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10) (2/2)

(ii) During operation (SEm.n = 1) (lower 8 bits: SDRmnL)



Caution Be sure to set transmit data to the SDRmnL register before the clock from the master is started.

Remark ☐: Setting is fixed in the CSI slave transmission/reception mode, ▒: Setting disabled (set to the initial value)

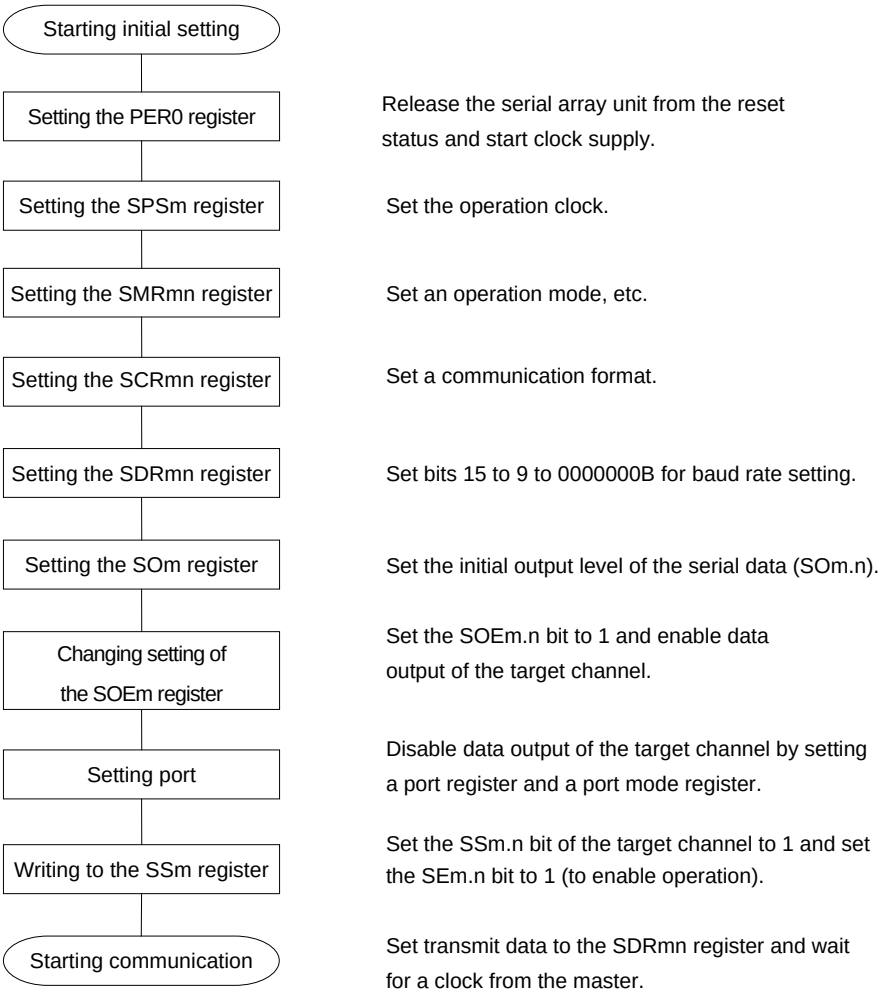
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

n: Channel number (n = 0, 1)

(2) Operation procedure

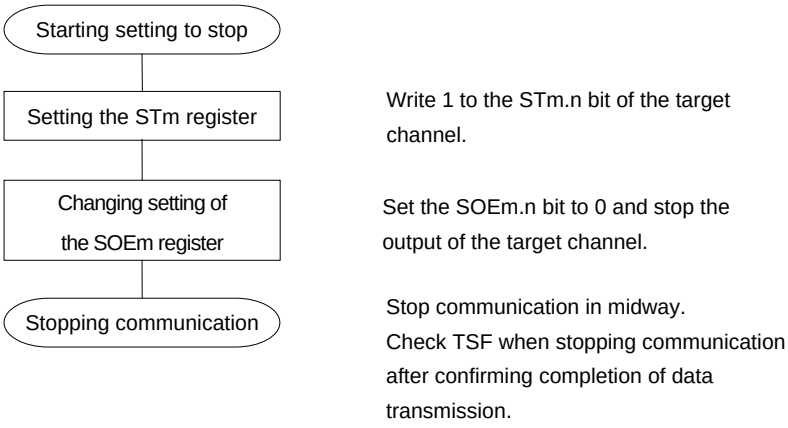
Figure 12-65. Initial Setting Procedure for Slave Transmission/Reception



Cautions 1. After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

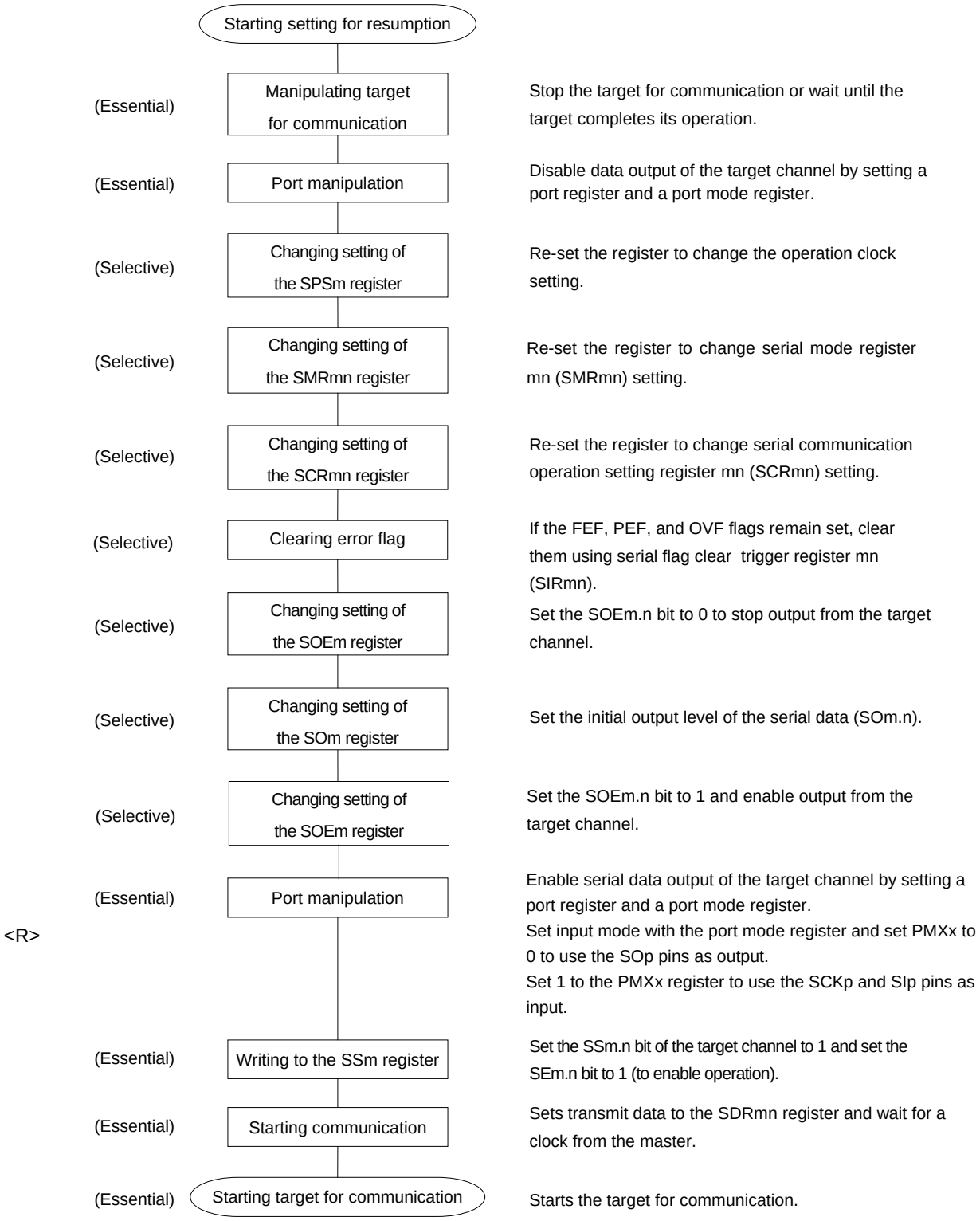
2. Be sure to set transmit data to the SDR register before the clock from the master is started.

Figure 12-66. Procedure for Stopping Slave Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 12-67 Procedure for Resuming Slave Transmission/ Reception**).

Figure 12-67. Procedure for Resuming Slave Transmission/Reception

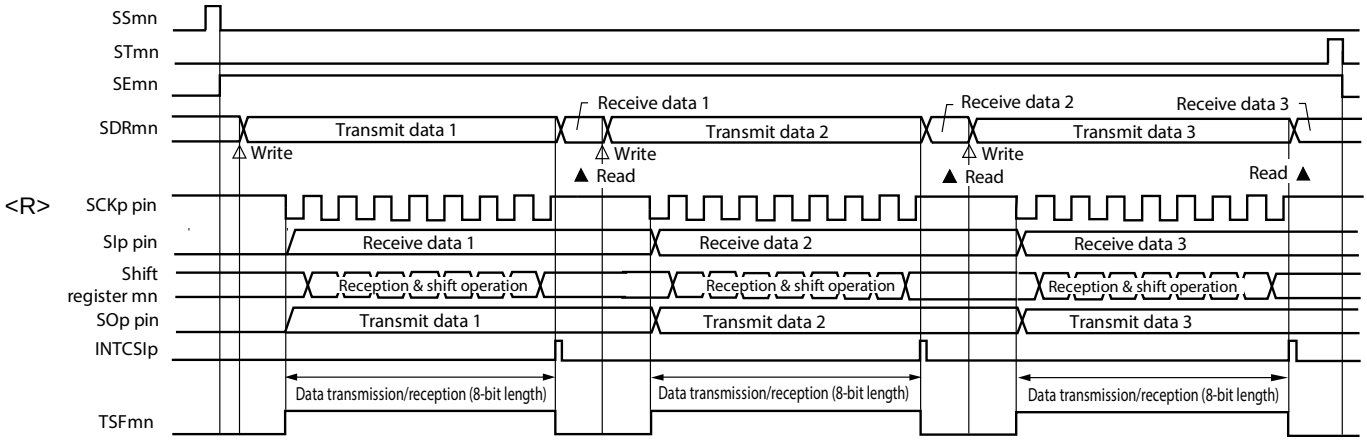


<R>

Caution Be sure to set transmit data to the SDR register before the clock from the master is started.

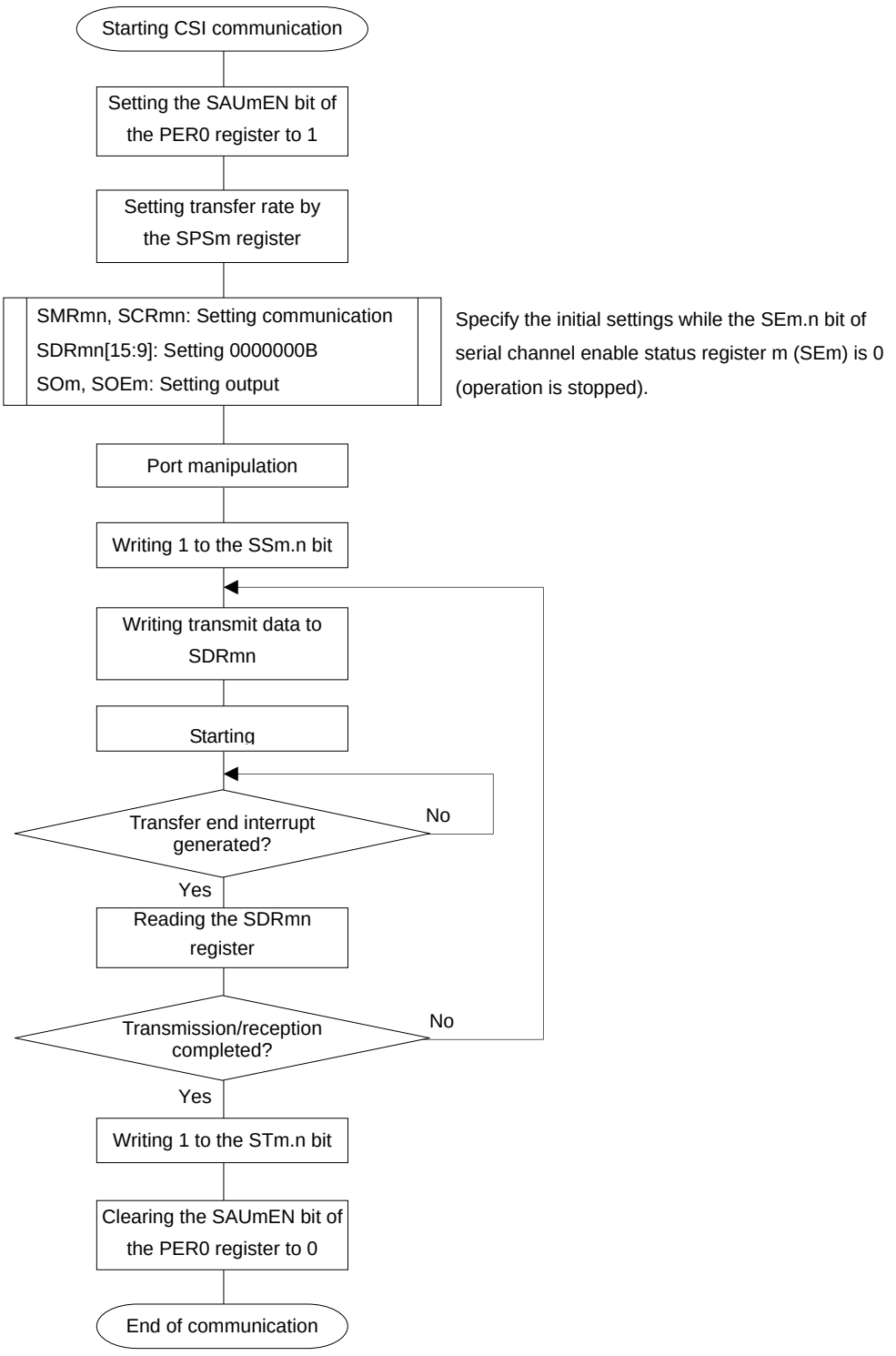
(3) Processing flow (in single-transmission/reception mode)

Figure 12-68. 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), mn = 00, 01, 10

Figure 12-69. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

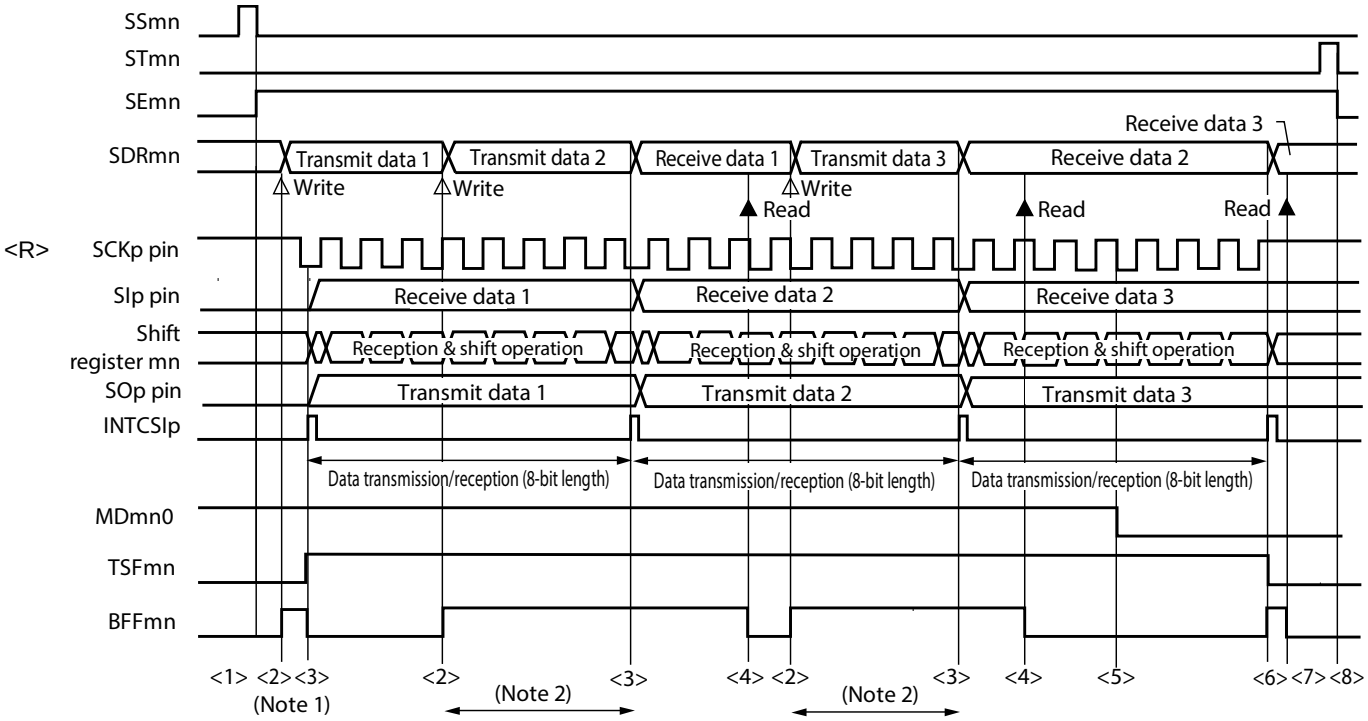


Cautions 1. After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SDR register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-70. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

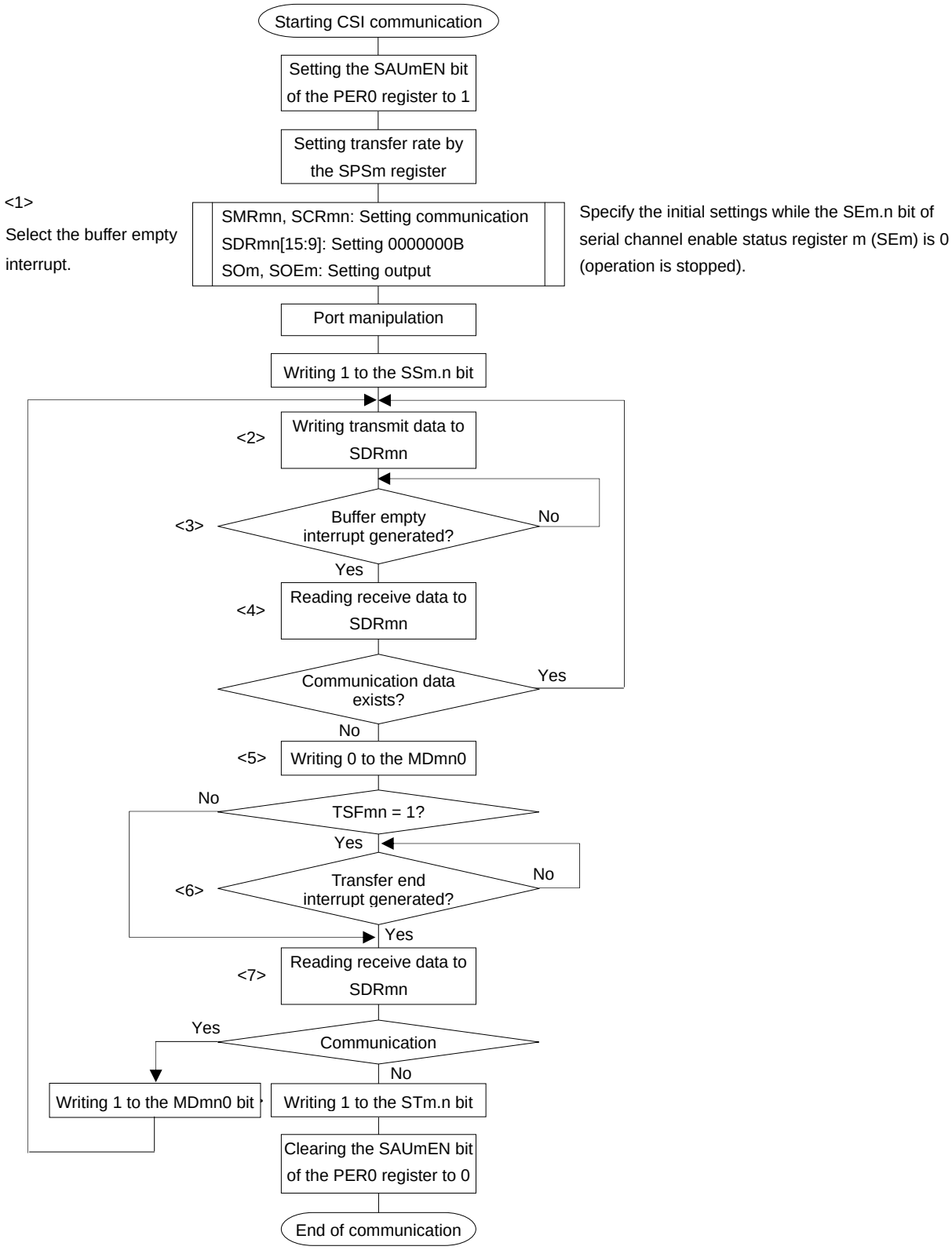
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-71 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), mn = 00, 10, 11

Figure 12-71. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Cautions 1. After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SDR register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/6.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-2. Selection of Operation Clock For 3-Wire Serial I/O

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} = 32 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
Other than the above									Setting prohibited	

Note Stop the operation of the serial array unit (SAU) (by setting bits 3 to 0 of ST0 register and bits 1 and 0 of ST1 and STS register to 1) before changing operation clock (f_{CLK}) selection (by changing the system clock control register (CKC) value).

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 10, 11

12.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10) communication is described in Figure 12-72.

Figure 12-72. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 10, 11

<R> 12.6 Operation of UART (UART0) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

[Data transmission/reception]

- Data length of 7, 8, 9 or 16 bits (UART0)
- 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

UART0 uses channels 0 and 1 of SAU0.

- 128-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	CSI01		–
1	0	CSI10	–	–
	1	–		IIC11

UART performs the following four types of communication operations.

- UART transmission (See **12.6.1**.)
- UART reception (See **12.6.2**.)

12.6.1 UART transmission

UART transmission is an operation to transmit data from the RL78/D1A to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0
Target channel	Channel 0 of SAU0
Pins used	TxD0
Interrupt	INTST0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7, 8, 9 or 16 bits
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note
Data phase	Non-reverse 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 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

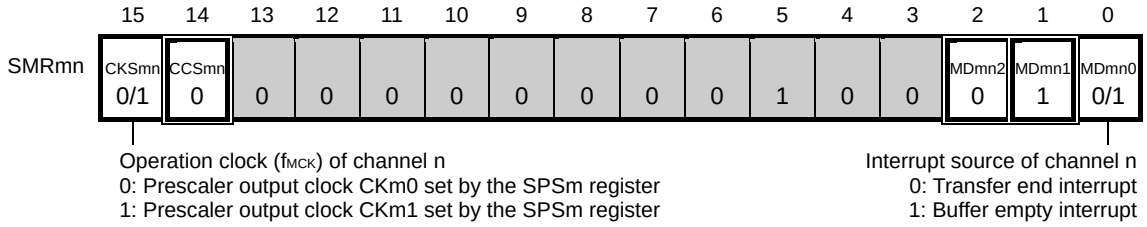
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

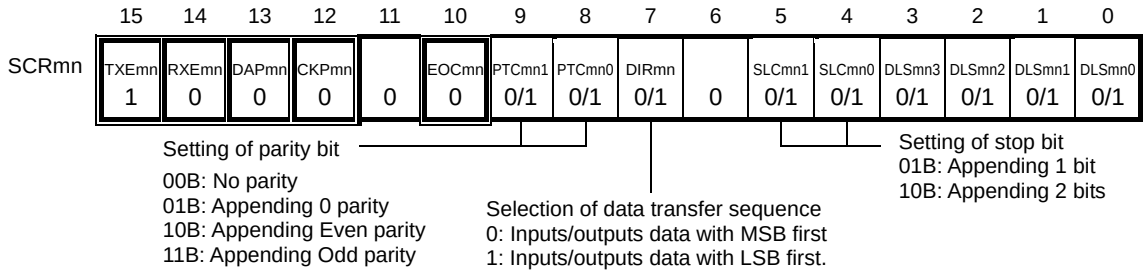
(1) Register setting

Figure 12-73. Example of Contents of Registers for UART Transmission of UART (UART0) (1/2)

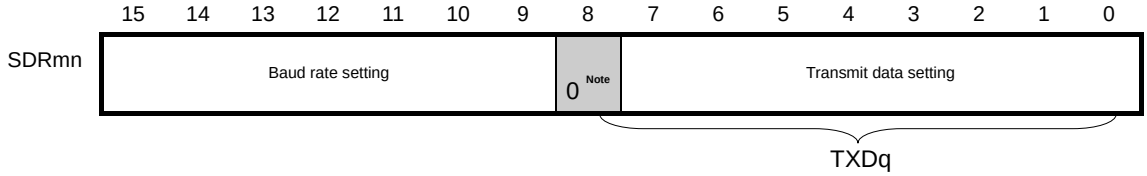
(a) Serial mode register mn (SMRmn)



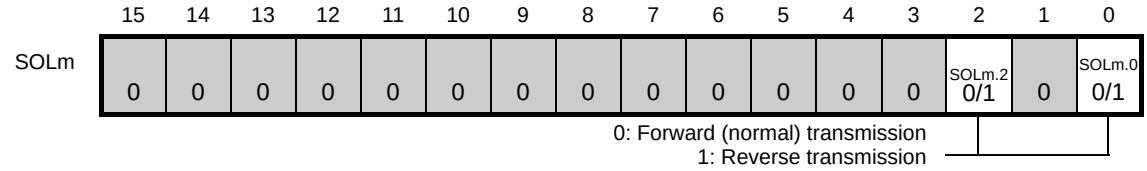
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.



Note When UART0 performs 9-bit communication (by setting the DLS001 and DLS000 bits of the SCR00 register to 0 and 1, respectively), bits 0 to 8 of the SDR00 register are used as the transmission data specification area. Only UART0 can be used to perform 9-bit communication.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

2. □: Setting is fixed in the UART transmission mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-73. Example of Contents of Registers for UART Transmission of UART (UART0) (2/2)

(e) Serial output register m (SOm) … Sets only the bits of the target channel.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

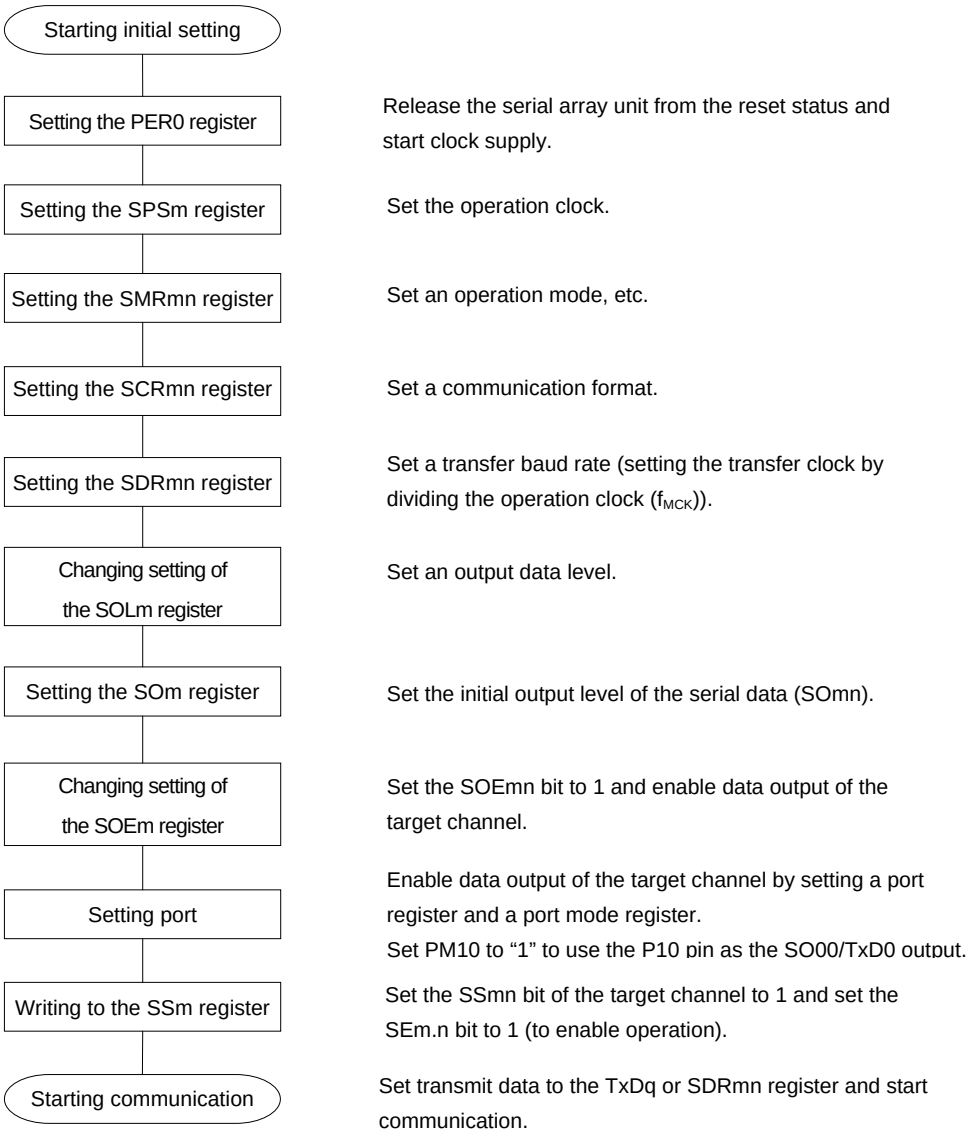
Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)
mn = 00

2. ☐ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

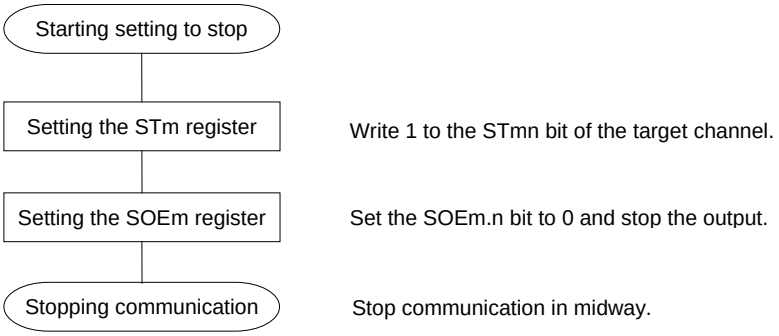
(2) Operation procedure

Figure 12-74. Initial Setting Procedure for UART Transmission



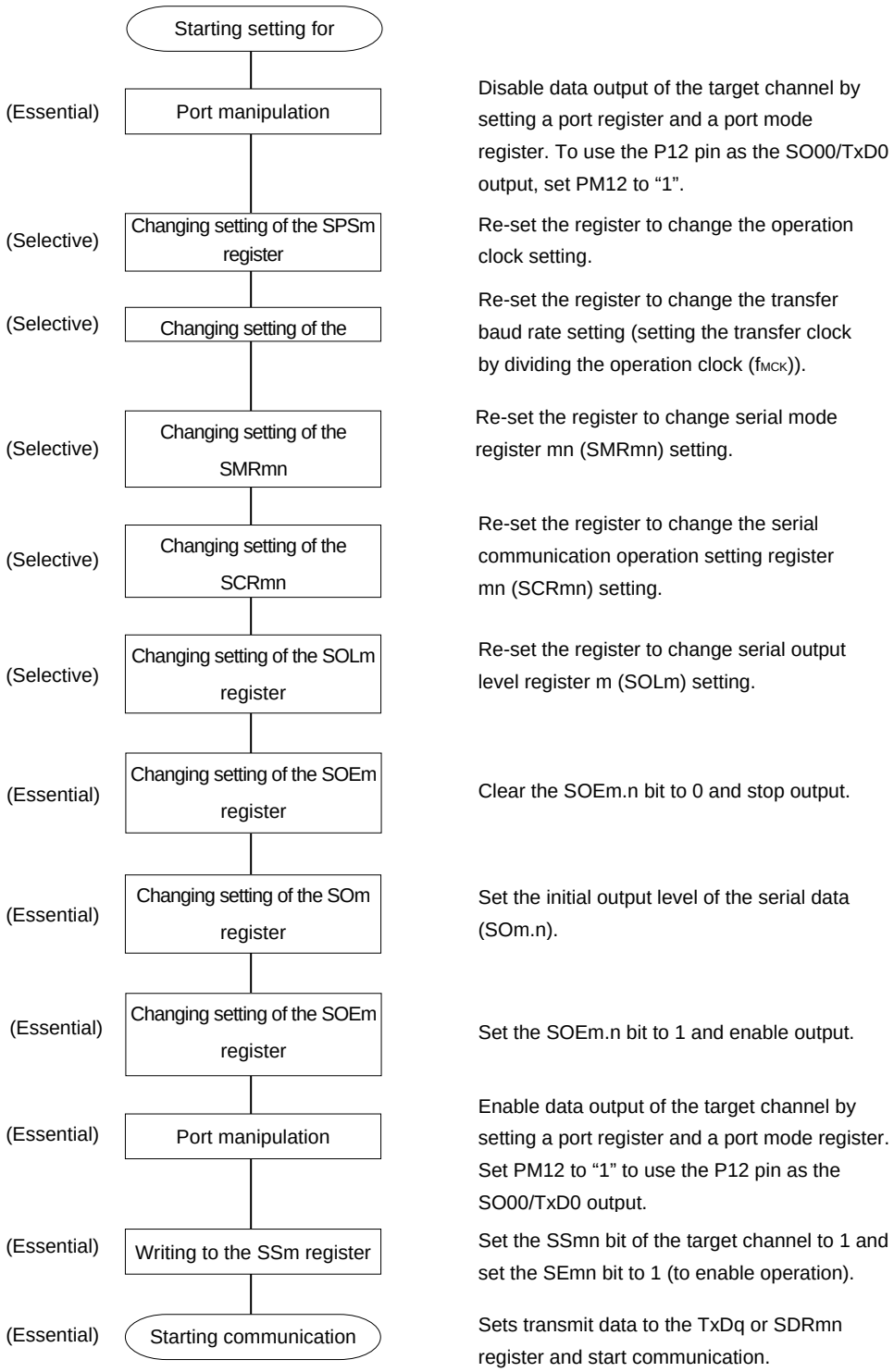
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 12-75. Procedure for Stopping UART Transmission



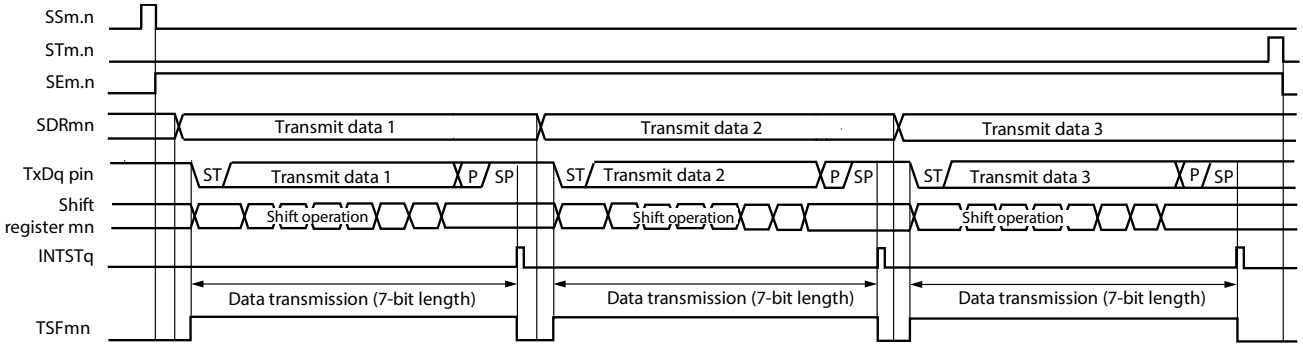
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 12-76 Procedure for Resuming UART Transmission**).

Figure 12-76. Procedure for Resuming UART Transmission



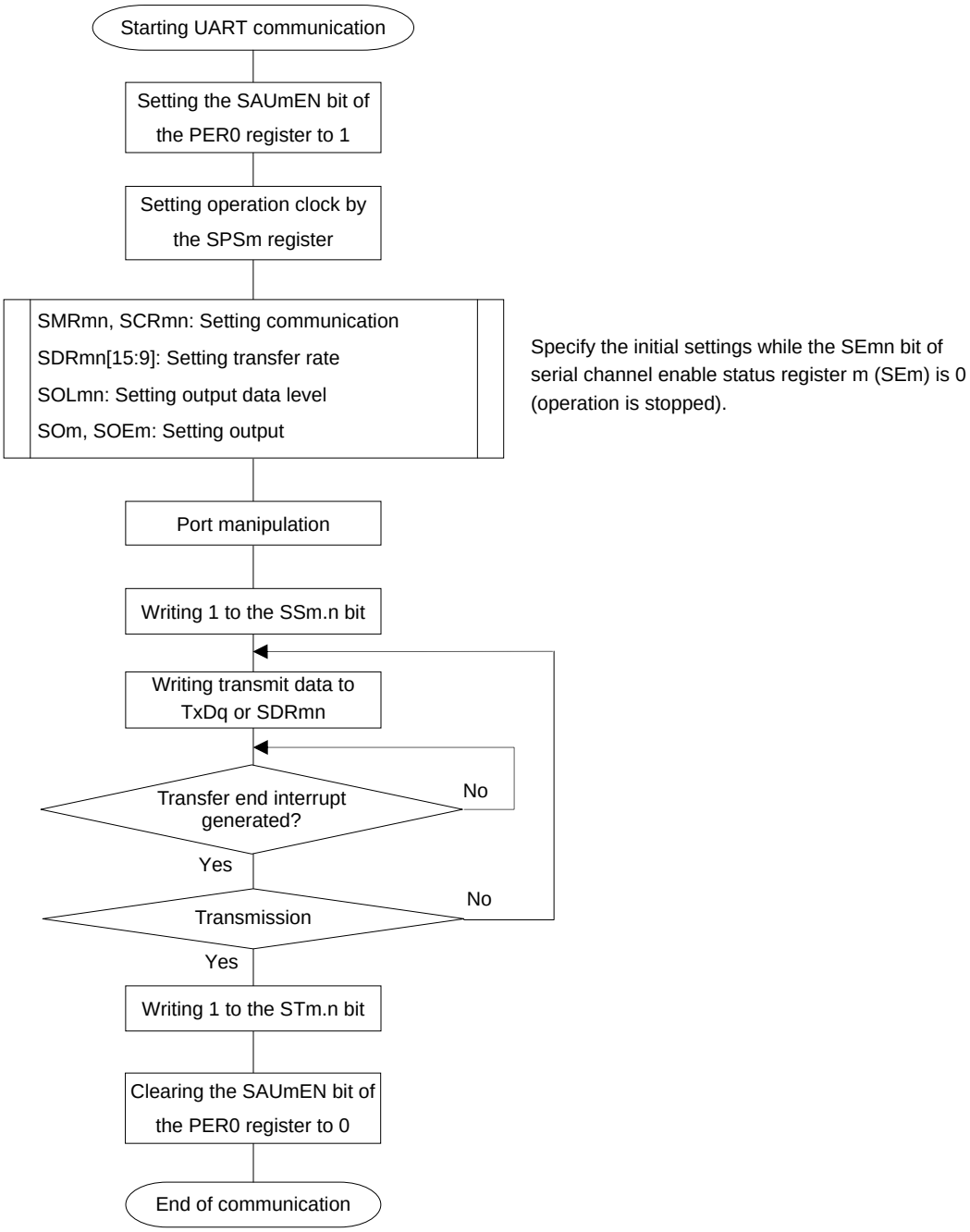
(3) Processing flow (in single-transmission mode)

Figure 12-77. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)
mn = 00

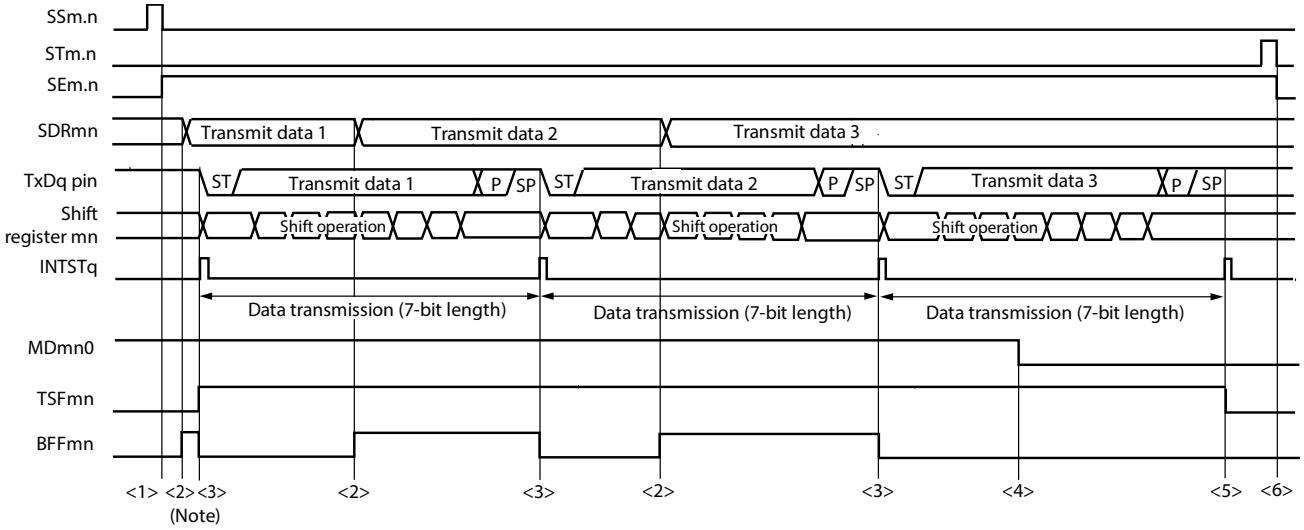
Figure 12-78. Flowchart of UART Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 12-79. Timing Chart of UART Transmission (in Continuous Transmission Mode)

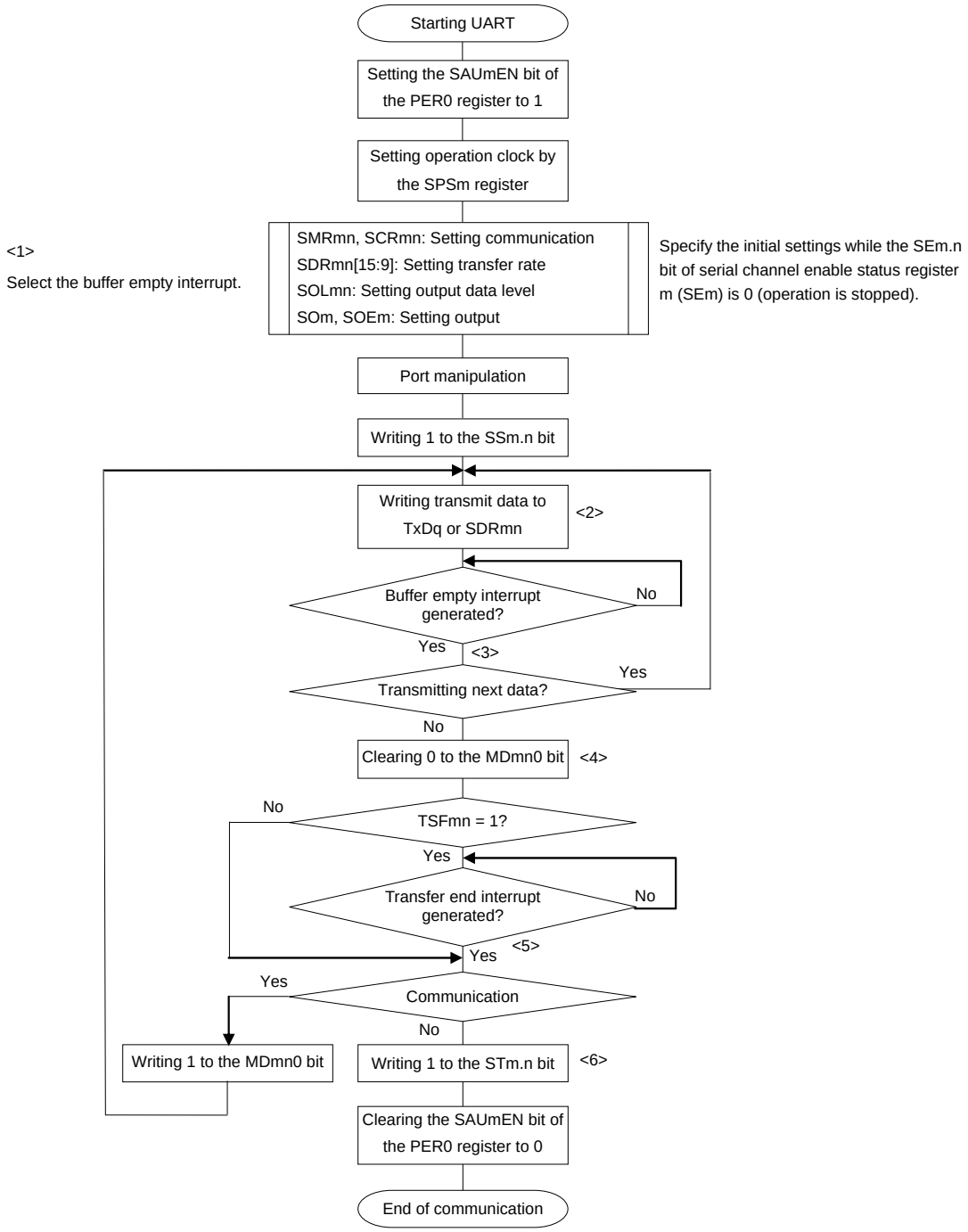


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)
mn = 00

Figure 12-80. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure G-79 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

12.6.2 UART reception

UART reception is an operation wherein the RL78/D1A asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0
Target channel	Channel 1 of SAU0
Pins used	RxD0
Interrupt	INTSR0
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	None
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)
Transfer data length	7, 8, 9 or 16 bits
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note
Data phase	Forward output (default: high level) Reverse output (default: low level)
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity
Stop bit	Appending 1 bit
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

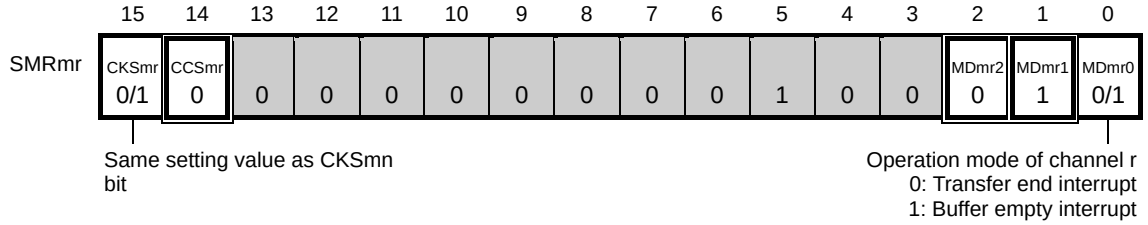
2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

(1) Register setting

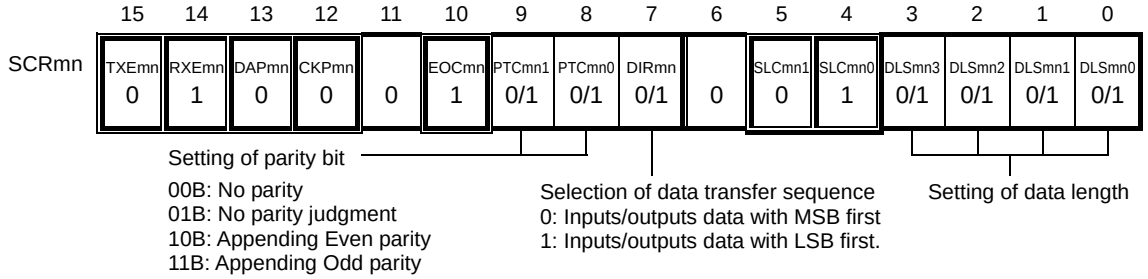
Figure 12-81. Example of Contents of Registers for UART Reception of UART (UART0) (1/2)

(a) Serial mode register mn (SMRmn)

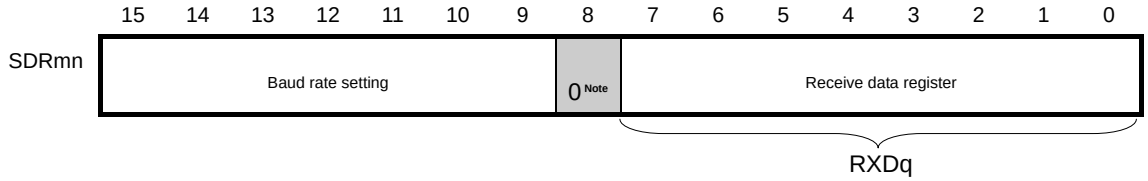
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Notes When UART0 performs 9-bit communication (by setting the DLS011 and DLS010 bits of the SMR01 register to 1), bits 0 to 8 of the SDR01 register are used as the transmission data specification area. Only UART0 can be used to perform 9-bit communication.

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01
r: Channel number (r = n – 1), q: UART number (q = 0)

2. ☐: Setting is fixed in the UART reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-81. Example of Contents of Registers for UART Reception of UART (UART0) (2/2)

(e) Serial output register m (SOm) … The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(f) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 ×

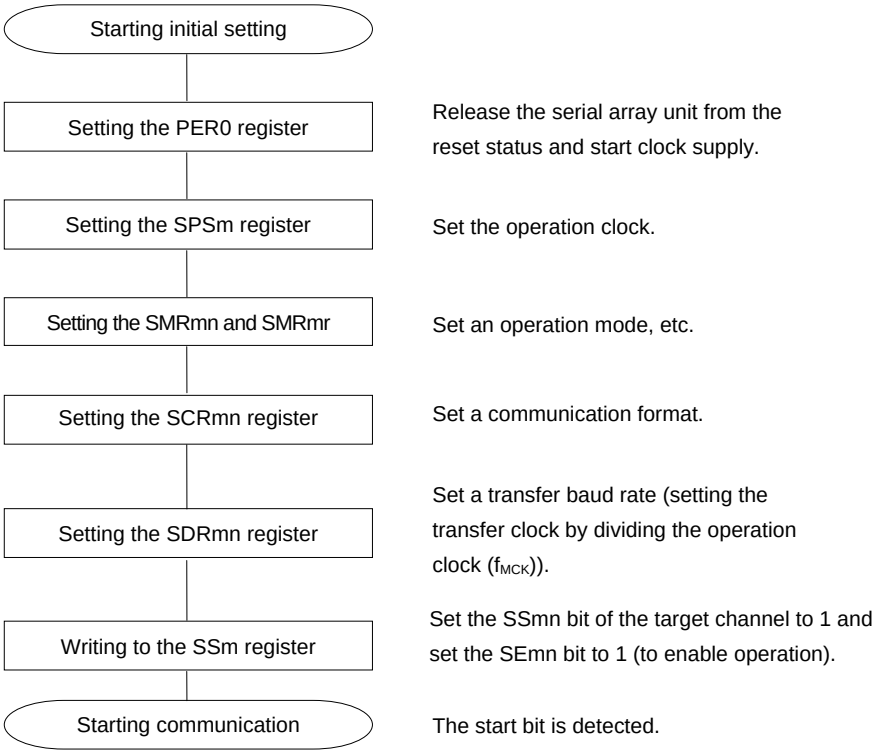
Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01
r: Channel number (r = n – 1), q: UART number (q = 0)

2. ☐: Setting is fixed in the UART reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-82. Initial Setting Procedure for UART Reception



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 12-83. Procedure for Stopping UART Reception

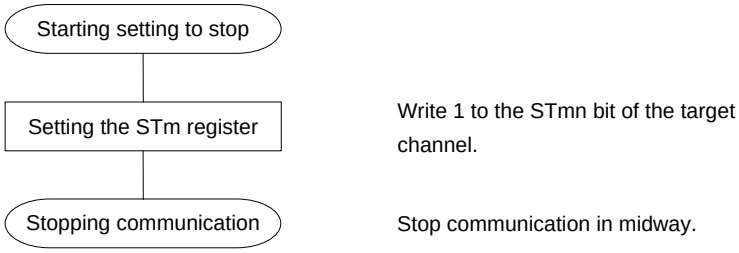
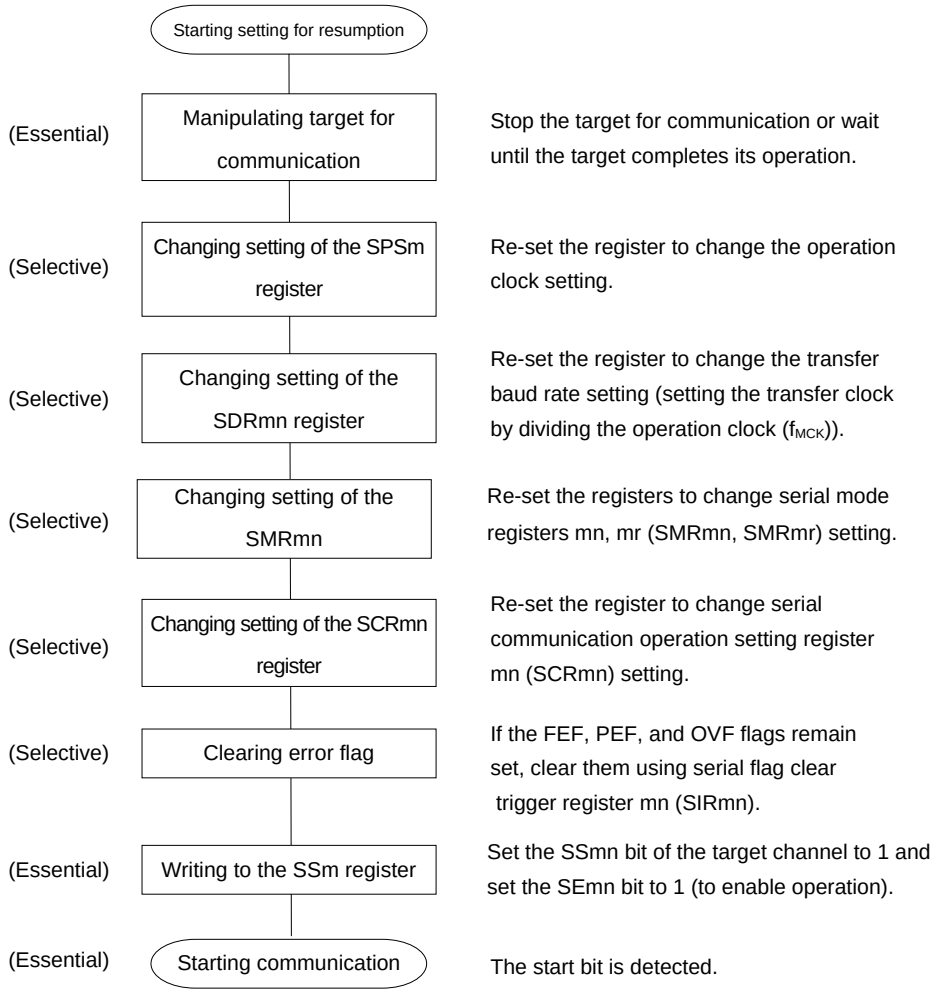
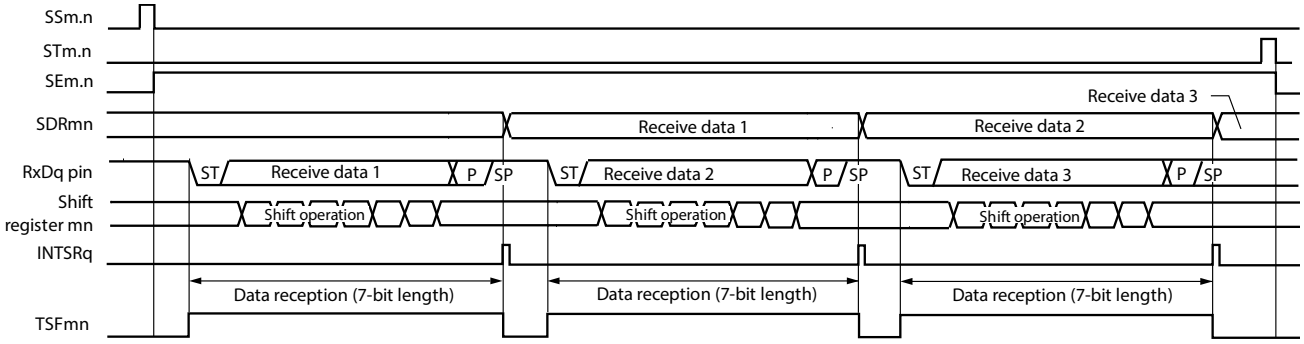


Figure 12-84. Procedure for Resuming UART Reception



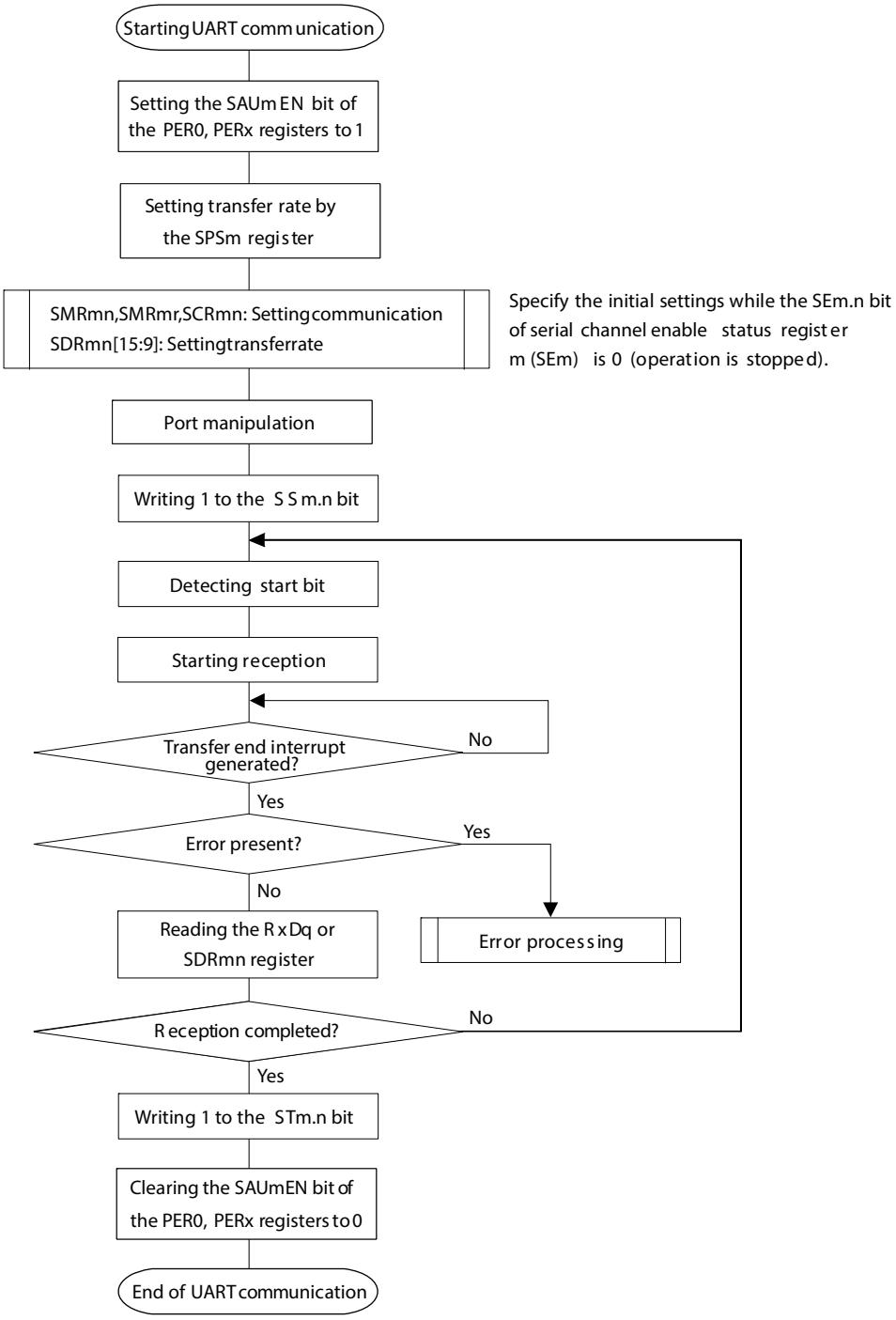
(3) Processing flow

Figure 12-85. Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1), mn = 01
r: Channel number (r = n – 1), q: UART number (q = 0)

Figure 12-86. Flowchart of UART Reception



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

12.6.3 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B, 0000010B) is prohibited in UART0.
Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited in UARTS0.

Remarks 1. When UART0 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.
When UARTS0 is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-3. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
Other than the above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: Don't care

2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(2) Baud rate error during transmission

The baud rate error of UART (UART0) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 32 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	103	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	103	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	103	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	103	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	103	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	103	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	103	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	63	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	103	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	103	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	103	153846 bps	+0.16 %
312500 bps	f_{CLK}	50	312500 bps	±0.39 %

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

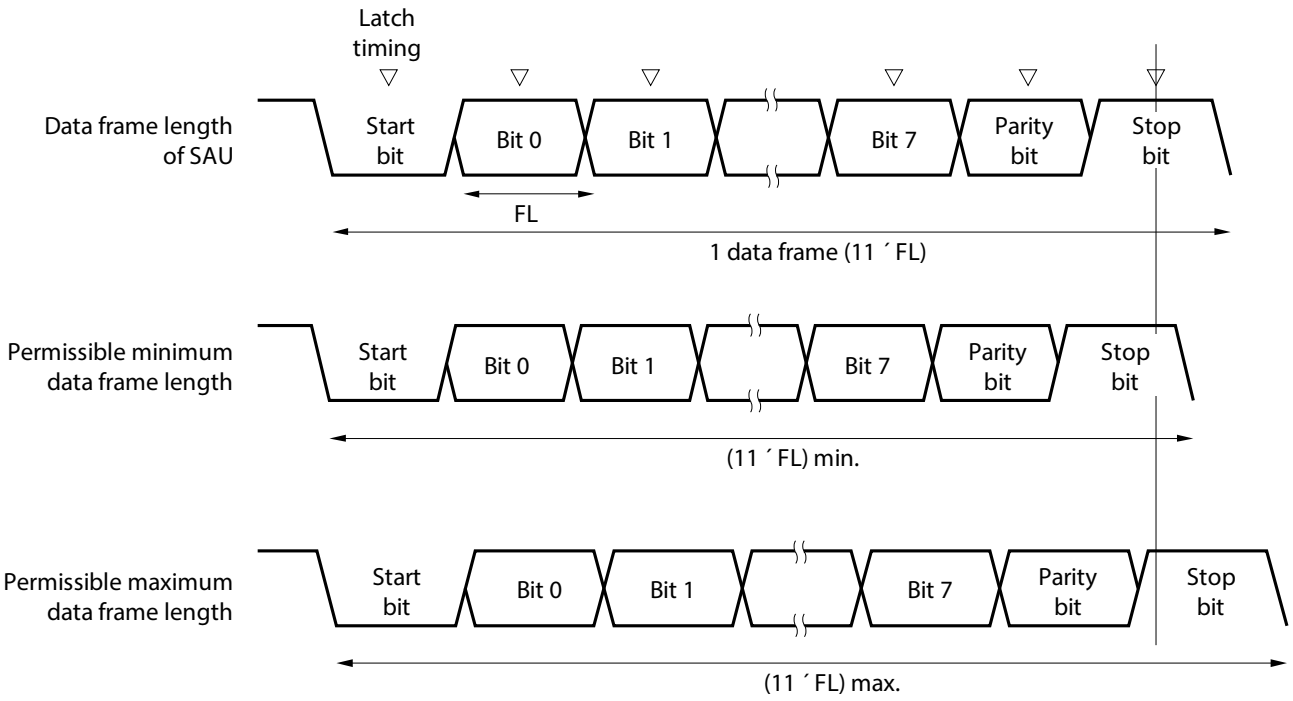
Brate: Calculated baud rate value at the reception side (See **12.6.3 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

Figure 12-87. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 12-87, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

12.6.4 Procedure for processing errors that occurred during UART (UART0) communication

The procedure for processing errors that occurred during UART (UART0) communication is described in Figures 12-88 and 12-89.

Figure 12-88. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 12-89. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STm.n bit of serial channel stop register m (STm) to 1. →	The SEm.n bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSm.n bit of serial channel start register m (SSm) to 1. →	The SEm.n bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

12.7 Operation of Simplified I^{2}C (IIC11) 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.

To generate the start and stop conditions, manipulate the control registers through software, considering the IIC bus line characteristics.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition by software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error detection flag

* [Functions not supported by simplified I^{2}C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEm.n (SOEm register) bit and serial communication data output is stopped. See the processing flow in **12.6.3 (2)** for details.

Remark m: Unit number (m = 1), n: Channel number (n = 1), mn = 11

The channels supporting simplified I^{2}C (IIC11) are channel 1 of SAU1.

Unit	Channel	Used as CSI	Used as Simplified I^{2}C
0	0	CSI00	–
	1	CSI01	–
1	0	CSI10	–
	1	–	IIC11

Simplified I^{2}C (IIC11) performs the following four types of communication operations.

- Address field transmission (See **12.6.1**.)
- Data transmission (See **12.6.2**.)
- Data reception (See **12.6.3**.)
- Stop condition generation (See **12.6.4**.)

12.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	IIC11
Target channel	Channel 1 of SAU1
Pins used	SCL11, SDA11 Note
Interrupt	INTIIC11
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	ACK error detection flag (PEFmn)
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)
Transfer rate	Max. $f_{MCK}/2$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)
Data level	Non-reverse output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK reception timing)
Data direction	MSB first

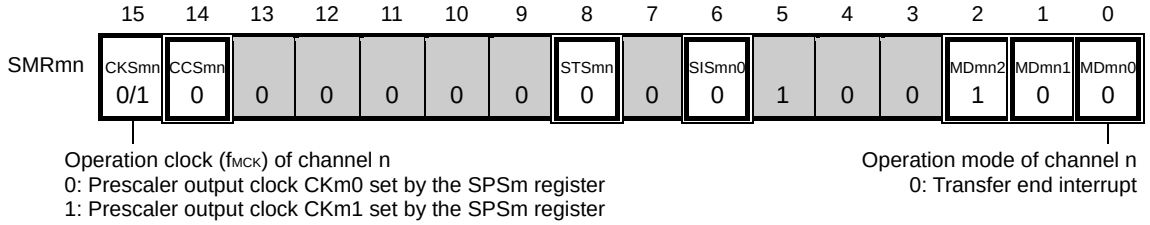
Note To perform communication via simplified I^2C, set the N-ch open-drain output (EV_{DD} tolerance) mode (POMxx = 1) for the port output mode registers (POMx) (see **4.3 Registers Controlling Port Function** for details).

Remark m: Unit number (m = 1), n: Channel number (n = 1), mn = 11

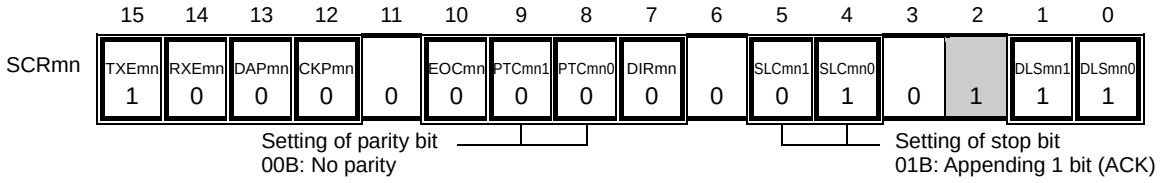
(1) Register setting

Figure 12-90. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC11)(1/2)

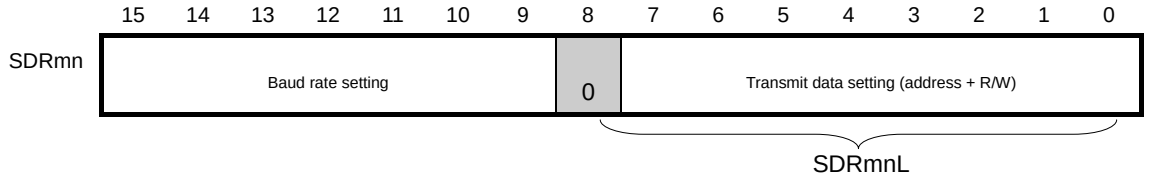
(a) Serial mode register mn (SMRmn)



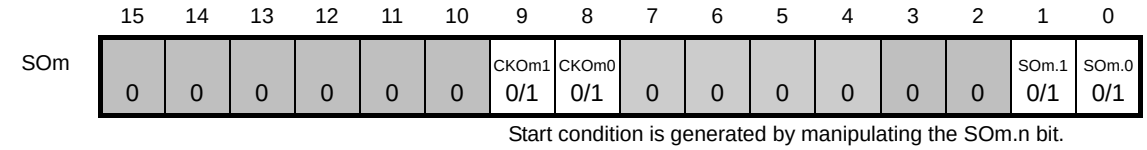
(b) Serial communication operation setting register mn (SCRmn)



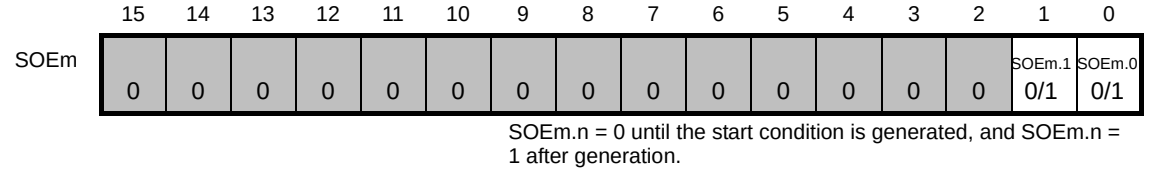
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



(d) Serial output register m (SOm)



(e) Serial output enable register m (SOEm)



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-90. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC00, IIC01, IIC11, IIC20, IIC21)(2/2)

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm.1 0/1	SSm.0 0/1

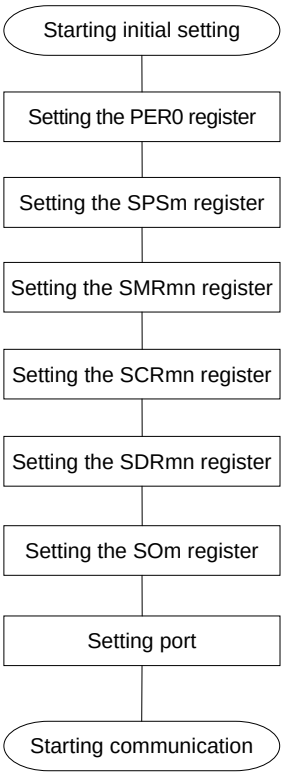
SSEm.n = 0 until the start condition is generated, and SSEm.n = 1 after generation.

Remarks 1. m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

2. ☐ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-91. 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 (setting the transfer clock by dividing the operation clock (f_{MCK})).

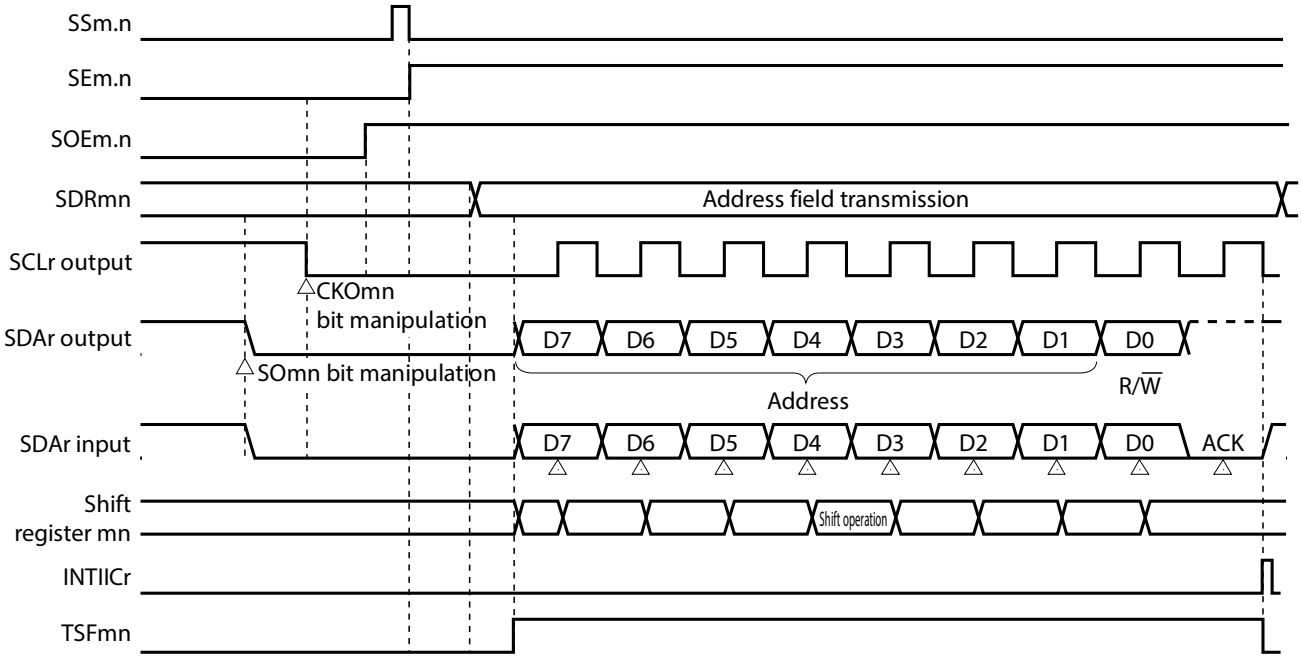
Set the initial output level of the serial data (SOmn) and serial clock (CKOmn).

<R> Enable data output, clock output, and N-ch open-drain output (EV_{DD} tolerance[2]) mode of the target channel by setting the port register, port mode register, and port output mode register. Set PM10 to “1” to use the P10 pin as the SCK00/SCL00.

Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

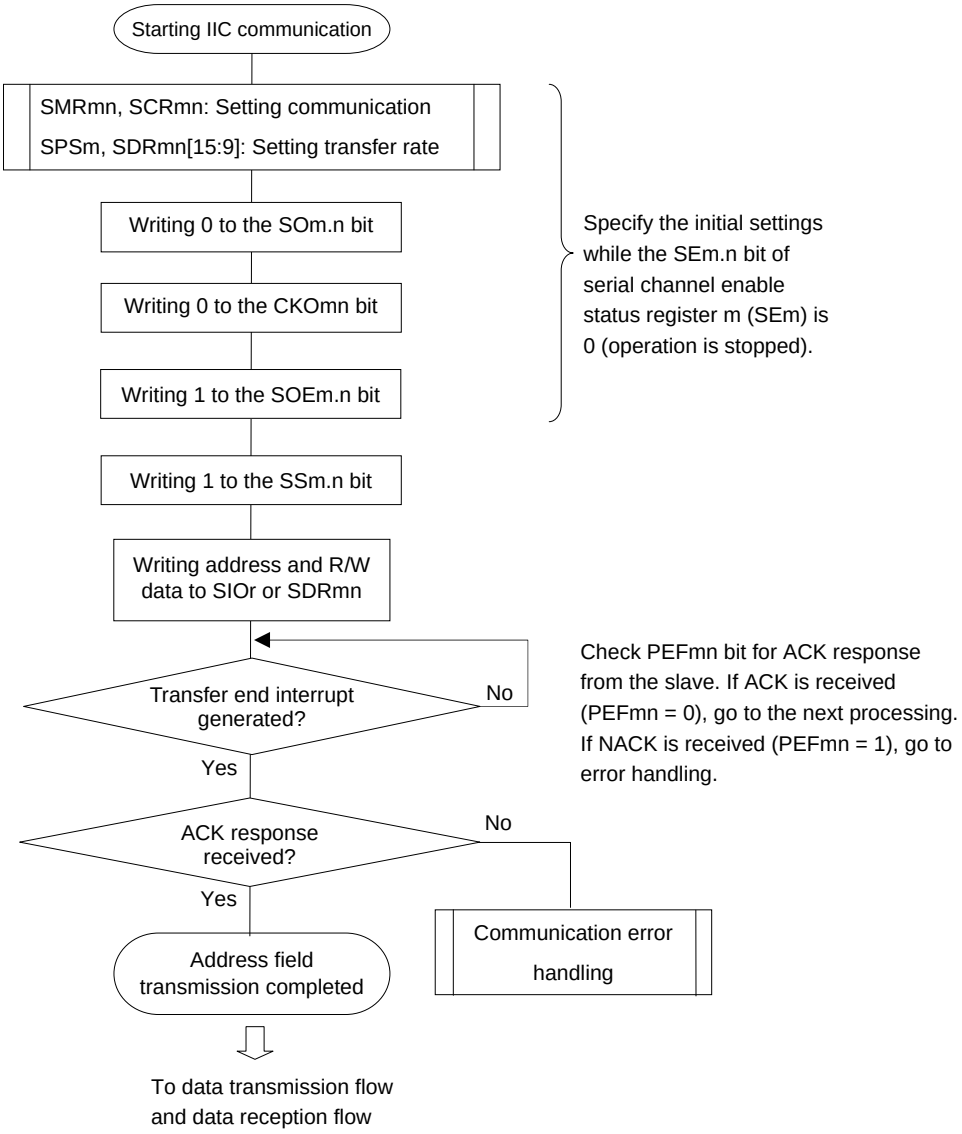
(3) Processing flow

Figure 12-92. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

Figure 12-93. Flowchart of Address Field Transmission



12.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC11
Target channel	Channel 3 of SAU0
Pins used	SCL11, SDA11 Note
Interrupt	INTIIC11
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	ACK error detection flag (PEFmn)
Transfer data length	8 bits
Transfer rate	Max. f_{MCK}/2 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)
Data level	Non-reverse 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 (EV_{DD} tolerance) mode (POMxx = 1) for the port output mode registers (POMx) (see **4.3 Registers Controlling Port Function** for details).

Remark m: Unit number (m = 1), n: Channel number (n = 1), mn = 11

(1) Register setting

Figure 12-94. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC11) (1/2)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

SDRmnL

(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1	CKOm0	0	0	0	0	0	0	SOm.1	SOm.0
							0/1Note	0/1Note							0/1Note	0/1Note

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm.1	SOEm.0
															1	1

Note The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-94. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC11) (2/2)

(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm.1 0/1	SSm.0 0/1

Remarks 1. m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

2. ☐ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 12-95. Timing Chart of Data Transmission

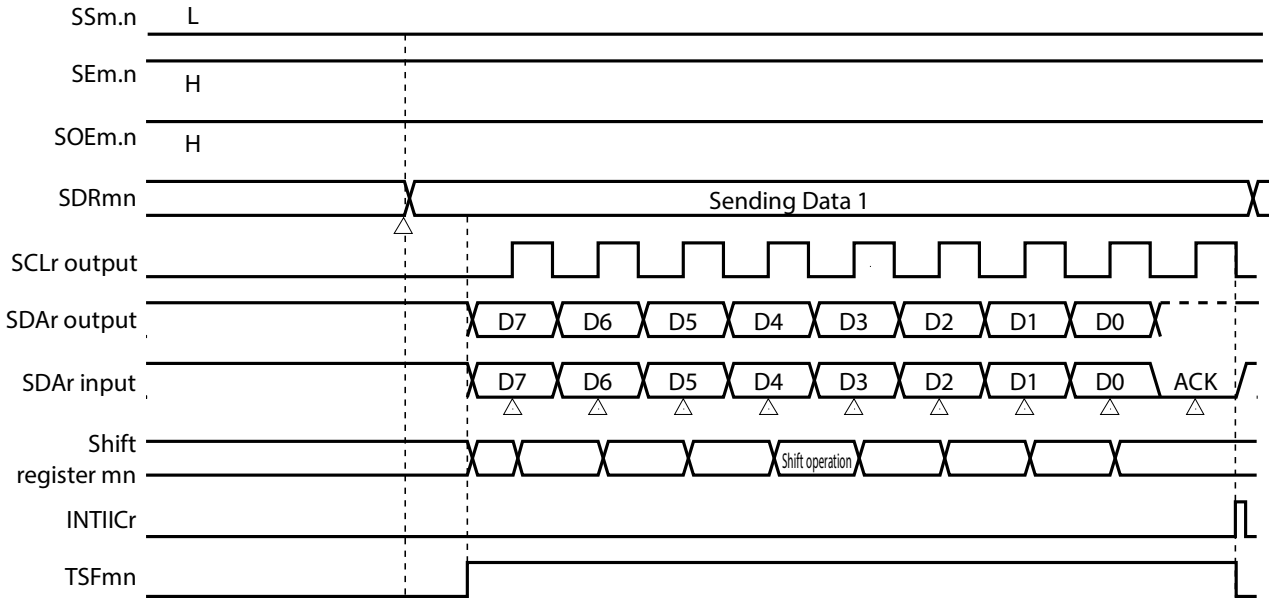
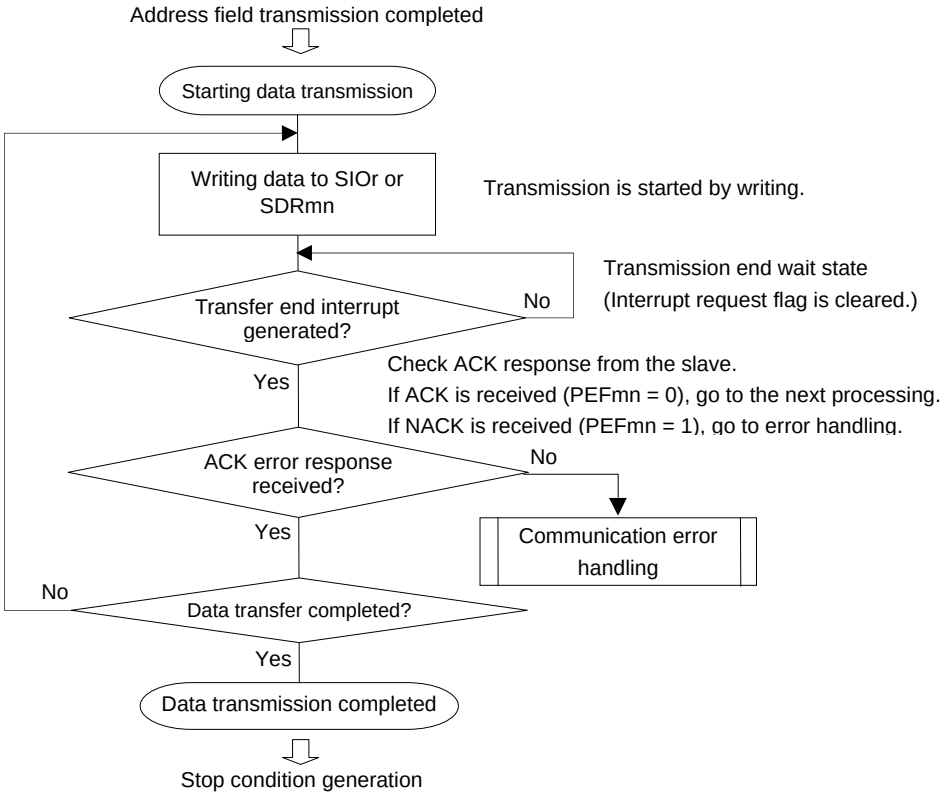


Figure 12-96. Flowchart of Data Transmission



12.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC11
Target channel	Channel 3 of SAU0
Pins used	SCL11, SDA11 Note
Interrupt	INTIIC11
	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}/2 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)
Data level	Non-reverse output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (ACK transmission)
Data direction	MSB first

Note To perform communication via simplified I^2C, set the N-ch open-drain output (EV_{DD} tolerance) mode (POMxx = 1) for the port output mode registers (POMx) (see **4.3 Registers Controlling Port Function** for details).

Remark m: Unit number (m = 1), n: Channel number (n = 1), mn = 11

(1) Register setting

Figure 12-97. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC11) (1/2)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Dummy transmit data setting (FFH)							

SDRmnL

(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm.1	SOm.0
	0	0	0	0	0	0	0/1Note	0/1Note2	0	0	0	0	0	0	0/1Note	0/1Note2

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm.1	SOEm.0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Notes The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-97. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC11) (2/2)

(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

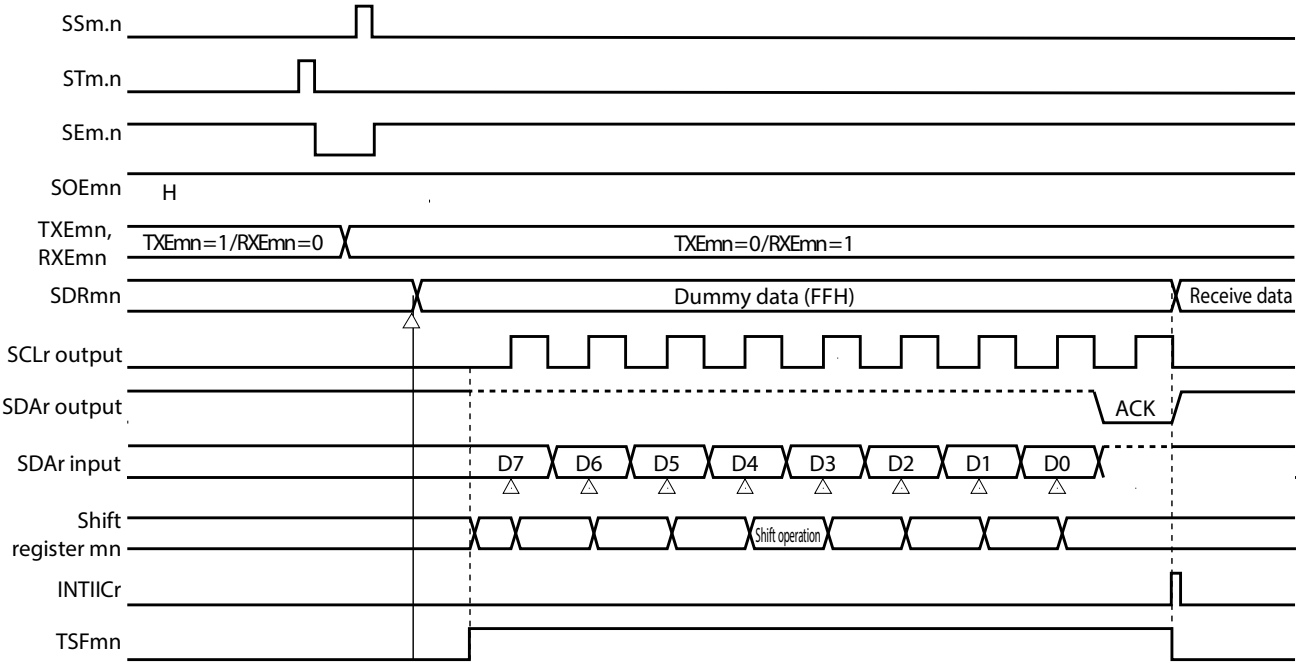
Remarks 1. m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

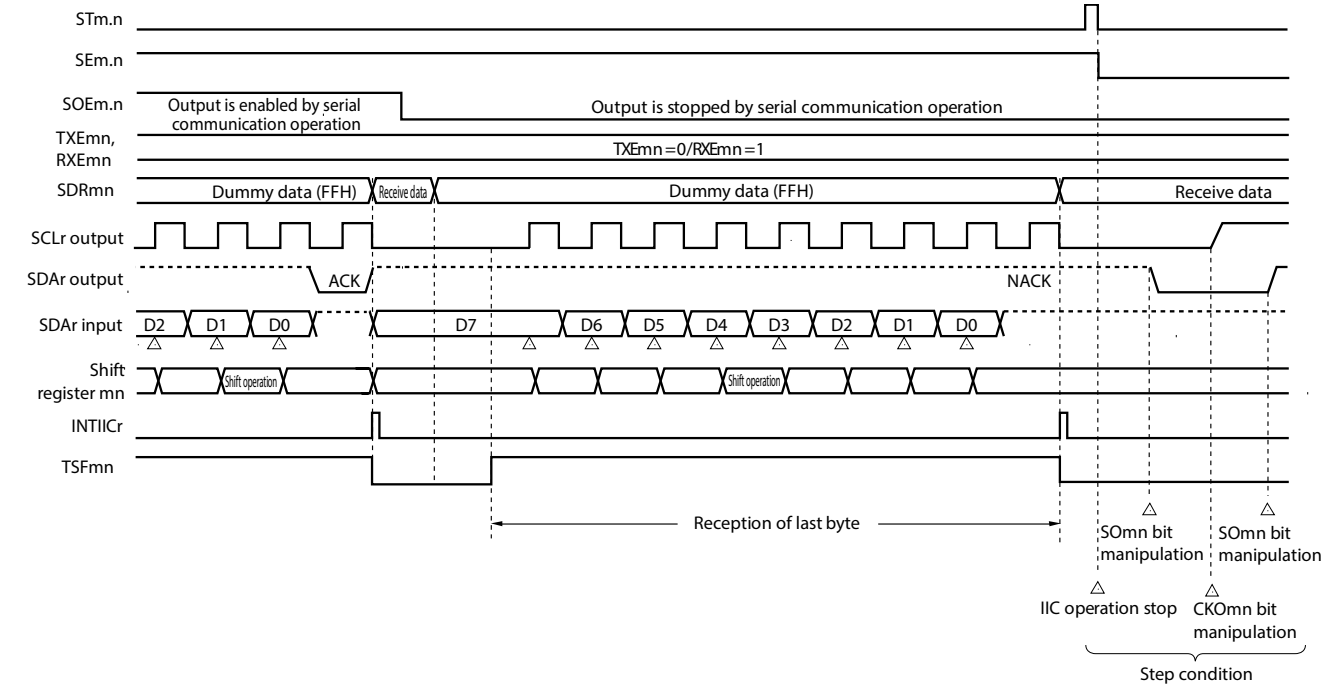
(2) Processing flow

Figure 12-98. Timing Chart of Data Reception

(a) When starting data reception

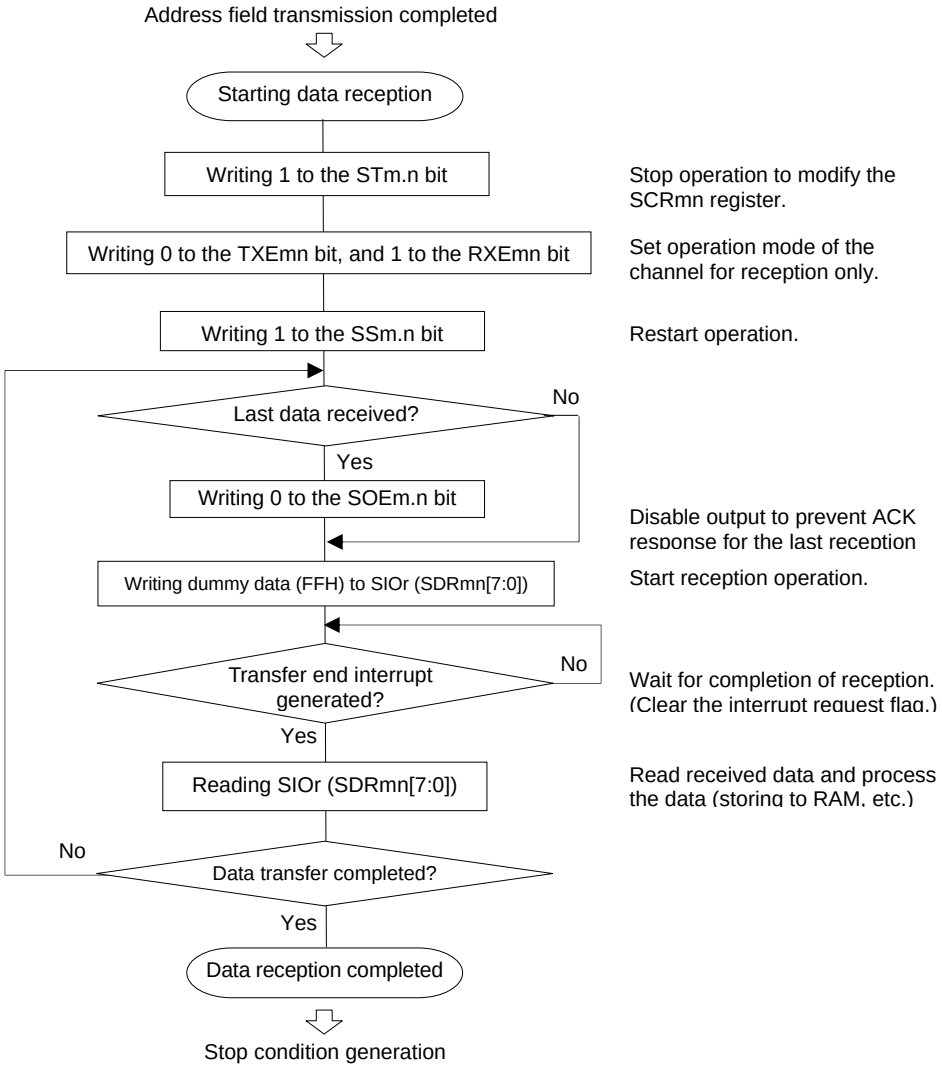


(b) When receiving last data



Remark m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

Figure 12-99. Flowchart of Data Reception



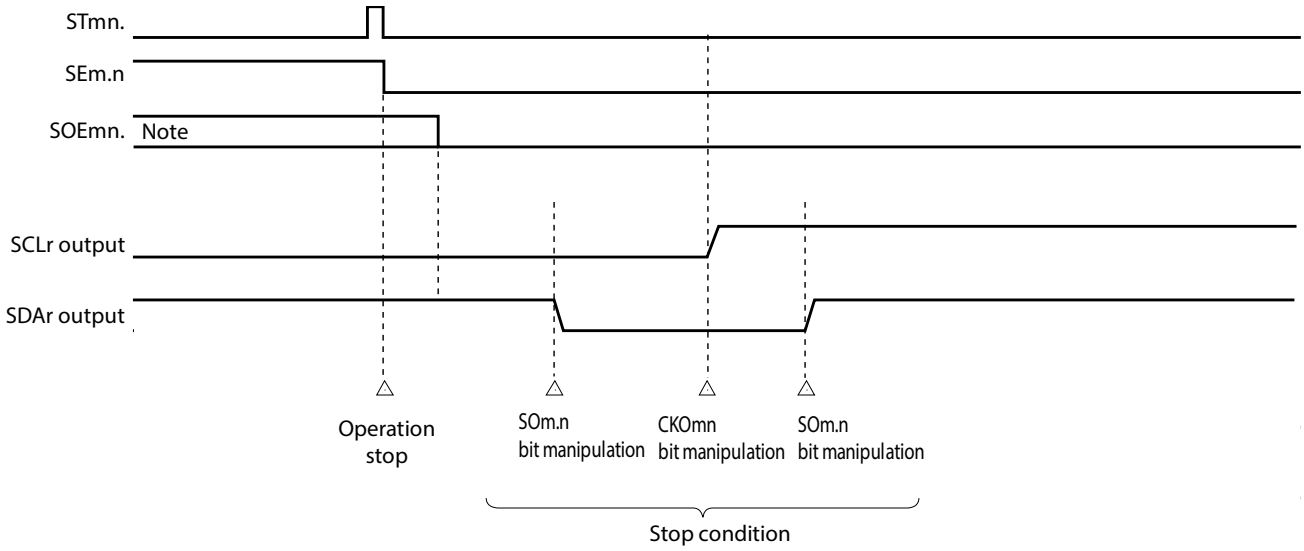
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STm.n bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

12.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

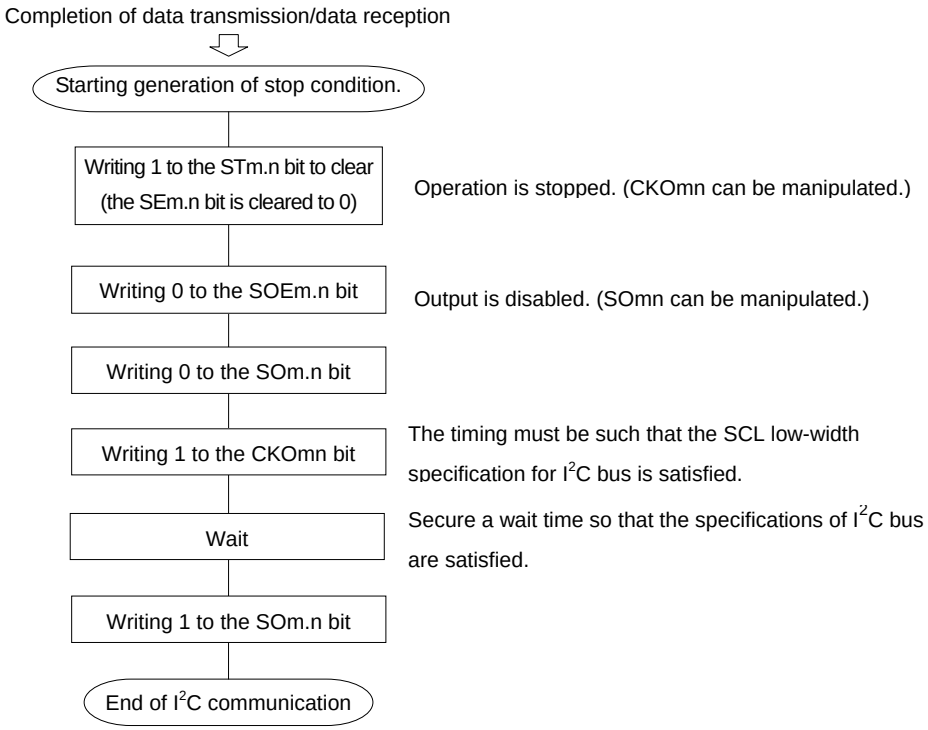
(1) Processing flow

Figure 12-100. Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEm.n bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 12-101. Flowchart of Stop Condition Generation



12.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC11) 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. Set SDRmn[15:9] to 0000001B or more.

The duty ratio of the SCL signal output from the simplified I^2C is 50%.

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), n: Channel number (n = 1), mn = 11

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-4. Selection of Operation Clock For Simplified I^2C

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} = 32 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
Other than the above									Setting prohibited	

Note Stop the operation of the serial array unit (SAU) (by setting bits 3 to 0 of ST0 register and bits 1 and 0 of ST1 and STS register to 1) before changing operation clock (f_{CLK}) selection (by changing the system clock control register (CKC) value).

Remarks 1. X: Don't care

2. m: Unit number (m = 1), n: Channel number (n = 1), mn = 11

Here is an example of setting an IIC transfer rate where $f_{MCK} = f_{CLK}$ = 32 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	79	100 kHz	0.0%
400 kHz	f_{CLK}	41	380 kHz	5.0% Note

Note The error cannot be controlled to about 0%, because the duty ratio of the SCL signal is 50%.

12.7.6 Procedure for processing errors that occurred during simplified I^2C (IIC11) communication

The procedure for processing errors that occurred during simplified I^2C (IIC11) communication is described in Figure 12-102.

Figure 12-102. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^2C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STm.n bit of serial channel stop register m (STm) to 1. →	The SEm.n bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSm.n bit of serial channel start register m (SSm) to 1. →	The SEm.n bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 1), n: Channel number (n = 1), r: IIC number (r = 11)
mn = 11

12.8 Relationship Between Register Settings and Pins

Tables 12-5 to **12-14** show the relationship between register settings and pins for each channel of serial array units 0, 1.

Table 12-5. Relationship between register settings and pins (Channel 0 of unit 0: CSI00, SCSI0001 = 0, SCSI000 = 0)

SE00 Note1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM10	P10	PM11	P11	PM12	P12	Operation mode	Pin Function		
															P10/LTxD1/ SCK00/TI10/ TO10/INTP4	P11/LRxD1/ INTPLR1/SI00/TI11/TO11	P12/SO00/ TI12/TO12/ INTP2
0	0	0	0	1	1	0	0	× Note2	× Note2	× Note2	× Note2	× Note2	× Note2	Operation stop mode	P10/LTxD1/ TI10/TO10/ INTP4	P11/LRxD1/ INTPLR1/ TI11/TO11	P12/TI12/ TO12/ INTP2
1	0	0	0	1	1	0	1	1	×	1	×	× Note2	× Note2	Slave CSI00 reception	SCK00 (input)	SI00	P12/TI12/ TO12/ INTP2
			1	0/1 Note3	1	1	0	1	×	×	×	0	1	Slave CSI00 transmission	SCK00 (input)	P11/TI11/ TO11	SO00
			1	0/1 Note3	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/ reception	SCK00 (input)	SI00	SO00
			0	1	0/1	0	1	0	1	1	×	× Note2	× Note2	Master CSI00 reception	SCK00 (output)	SI00	P12/SO00/ TI12/TO12/ INTP2
			1	0/1 Note3	0/1 Note3	1	0	0	1	×	×	0	1	Master CSI00 transmission	SCK00 (output)	P11/TI11/ TO11	SO00
			1	0/1 Note3	0/1 Note3	1	1	0	1	1	×	0	1	Master CSI00 transmission/ reception	SCK00 (output)	SI00	SO00

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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

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

Table 12-6. Relationship between register settings and pins (Channel 0 of unit 0: CSI00, SCSI0001 = 0, SCSI000 = 1)

SE00 Note1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM04	P04	PM03	P03	PM02	P02	Operation mode	Pin Function		
															P04/SCK00/ TI04/TO04/ TI14/TO14	P03/SI00/ TI03/TO03/ TI13/TO13	P02/ SO00/ TI02/TO02/ TI12/TO12
0	0	0	0	1	1	0	0	× Note2	× Note2	× Note2	× Note2	× Note2	× Note2	Operation stop mode	P04/TI04/ TO04/ TI14/TO14	P03/TI03/ TO03/ TI13/TO13	P02/TI02/ TO02/ TI12/TO12
1	0	0	0	1	1	0	1	1	×	1	×	× Note2	× Note2	Slave CSI00 reception	SCK00 (input)	SI00	P02/TI02/ TO02/ TI12/TO12
			1	0/1 Note3	1	1	0	1	×	×	×	0	1	Slave CSI00 transmission	SCK00 (input)	P03/TI03/ TO03/ TI13/TO13	SO00
			1	0/1 Note3	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/ reception	SCK00 (input)	SI00	SO00
			0	1	0/1	0	1	0	1	1	×	× Note2	× Note2	Master CSI00 reception	SCK00 (output)	SI00	P02/TI02/ TO02/ TI12/TO12
			1	0/1 Note3	0/1 Note3	1	0	0	1	×	×	0	1	Master CSI00 transmission	SCK00 (output)	P03/TI03/ TO03/ TI13/TO13	SO00
			1	0/1 Note3	0/1 Note3	1	1	0	1	1	×	0	1	Master CSI00 transmission/ reception	SCK00 (output)	SI00	SO00

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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

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

Table 12-7. Relationship between register settings and pins (Channel 0 of unit 0: CSI00, SCSI001 = 1, SCSI000 = 0)

SE00 **Note1**	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM34	P34	PM33	P33	PM32	P32	Operation mode	Pin Function		
															P34/TI24/TO24/SCK00	P33/TI23/TO23/SI00	P32/TI22/TO22/SO00
0	0	0	0	1	1	0	0	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	Operation stop mode	P34/TI24/TO24	P33/TI23/TO23	P32/TI22/TO22
1	0	0	0	1	1	0	1	1	×	1	×	× **Note2**	× **Note2**	Slave CSI00 reception	SCK00 (input)	SI00	P32/TI22/TO22
			1	0/1 **Note3**	1	1	0	1	×	× **Note2**	× **Note2**	0	1	Slave CSI00 transmission	SCK00 (input)	P33/TI23/TO23	SO00
			1	0/1 **Note3**	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/reception	SCK00 (input)	SI00	SO00
			0	1	0/1	0	1	0	1	1	×	× **Note2**	× **Note2**	Master CSI00 reception	SCK00 (output)	SI00	P32/TI22/TO22
			1	0/1 **Note3**	0/1 **Note3**	1	0	0	1	× **Note2**	× **Note2**	0	1	Master CSI00 transmission	SCK00 (output)	P33/TI23/TO23	SO00
			1	0/1 **Note3**	0/1 **Note3**	1	1	0	1	1	×	0	1	Master CSI00 transmission/reception	SCK00 (output)	SI00	SO00

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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

Table 12-8. Relationship between register settings and pins (Channel 1 of unit 0: CSI01, SCSI010 = 0)

SE 01 **Note1**	MD 012	MD 011	SOE 01	SO 01	CKO 01	TXE 01	RXE 01	PM74	P74	PM 75	P75	PM13	P13	Operation mode	Pin Function		
															P74/SCK01/ TI23/TO123	P75/PCL/ SI01/TI22/ TO22	P13/SO01/ TI13/TO13
0	0	0	0	1	1	0	0	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	Operation stop mode	P74/TI23/ TO123	P75/PCL/ TI22/TO22	P13/TI13/ TO13
1	0	0	0	1	1	0	1	1	×	1	×	× **Note2**	× **Note2**	Slave CSI01 reception	SCK01 (input)	SI01	P13/TI13/ TO13
			1	0/1 **Note3**	1	1	0	1	×	× **Note2**	× **Note2**	0	1	Slave CSI01 transmission	SCK01 (input)	P75/PCL/ TI22/TO22	SO01
			1	0/1 **Note3**	1	1	1	1	×	1	×	0	1	Slave CSI01 transmission/ reception	SCK01 (input)	SI01	SO01
			0	1	0/1	0	1	0	1	1	×	× **Note2**	× **Note2**	Master CSI01 reception	SCK01 (output)	SI01	P13/TI13/ TO13
			1	0/1 **Note3**	0/1 **Note3**	1	0	0	1	× **Note2**	× **Note2**	0	1	Master CSI01 transmission	SCK01 (output)	P75/PCL/ TI22/TO22	SO01
			1	0/1 **Note3**	0/1 **Note3**	1	1	0	1	1	×	0	1	Master CSI01 transmission/ reception	SCK01 (output)	SI01	SO01

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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

Table 12-9. Relationship between register settings and pins (Channel 1 of unit 0: CSI01, SCSI010 = 1)

SE 01 **Note1**	MD 012	MD 011	SOE 01	SO 01	CKO 01	TXE 01	RXE 01	PM56	P56	PM55	P55	PM54	P54	Operation mode	Pin Function		
															P56/TI16/ TO16/ SCK01	P55/TI15/ TO15/SI01	P54/TI14/ TO14/SO0 1
0	0	0	0	1	1	0	0	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	Operation stop mode	P56/TI16/ TO16/	P55/TI15/ TO15	P54/TI14/ TO14
1	0	0	0	1	1	0	1	1	×	1	×	× **Note2**	× **Note2**	Slave CSI01 reception	SCK01 (input)	SI01	P54/TI14/ TO14
			1	0/1 **Note3**	1	1	0	1	×	× **Note2**	× **Note2**	0	1	Slave CSI01 transmission	SCK01 (input)	P55/TI15/ TO15	SO01
			1	0/1 **Note3**	1	1	1	1	×	1	×	0	1	Slave CSI01 transmission/ reception	SCK01 (input)	SI01	SO01
			0	1	0/1	0	1	0	1	1	×	× **Note2**	× **Note2**	Master CSI01 reception	SCK01 (output)	SI01	P54/TI14/ TO14
			1	0/1 **Note3**	0/1 **Note3**	1	0	0	1	× **Note2**	× **Note2**	0	1	Master CSI01 transmission	SCK01 (output)	P55/TI15/ TO15	SO01
			1	0/1 **Note3**	0/1 **Note3**	1	1	0	1	1	×	0	1	Master CSI01 transmission/ reception	SCK01 (output)	SI01	SO01

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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

Table 12-10. Relationship between register settings and pins (Channel 0 of unit 1: CSI10, SCSI100 = 0)

SE 10 **Note1**	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 10	P133	PM 132	P132	PM 131	P131	Operation mode	Pin Function		
															P133/SCK10/ TI22/TO22	P132/SI10/ LRxD1/INTPLR1 /TI20/TO20	P131/SO10/ LTxD1/TI21/ TO21
0	0	0	0	1	1	0	0	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	× **Note2**	Operation stop mode	P132/INTP4/ CTxD/LTxD1/ TI00/P133/ TI22/TO22	P132/INTP5/ LRxD1/ INTPLR1/TI20/ TO20	P131/TI21/ TO21
1	0	0	0	1	1	0	1	1	×	1	×	× **Note2**	× **Note2**	Slave CSI10 reception	SCK10 (input)	SI10	P131/TI21/ TO21
			1	0/1 **Note3**	1	1	1	1	×	× **Note2**	× **Note2**	0	1	Slave CSI10 transmission	SCK10 (input)	P132/INTP5/ TI20/TO20	SO10
			1	0/1 **Note3**	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission/ reception	SCK10 (input)	SI10	SO10
			0	1	0/1	0	1	0	1	1	×	× **Note2**	× **Note2**	Master CSI10 reception	SCK10 (output)	SI10	P131/TI21/ TO21
			1	0/1 **Note3**	0/1 **Note3**	1	0	0	1	× **Note2**	× **Note2**	0	1	Master CSI10 transmission	SCK10 (output)	P132/INTP5/ TI20/TO20	SO10
			1	0/1 **Note3**	0/1 **Note3**	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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

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

Table 12-11. Relationship between register settings and pins (Channel 0 of unit 1: CSI10, SCSI100 = 1)

SE 10 Note1	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 10	P51	PM 52	P52	PM 53	P53	Operation mode	Pin Function		
															P51/TI04/ TO04/SCK10	P52/TI06/ TO06/SI10	P53/TI13/ TO13/SO10
0	0	0	0	1	1	0	0	× Note2	× Note2	× Note2	× Note2	× Note2	× Note2	Operation stop mode	P51/TI04/ TO04	P52/TI06/ TO06	P53/TI13/ TO13
1	0	0	0	1	1	0	1	1	×	1	×	× Note2	× Note2	Slave CSI10 reception	SCK10 (input)	SI10	P53/TI13/ TO13
			1	0/1 Note3	1	1	1	1	×	× Note2	× Note2	0	1	Slave CSI10 transmission	SCK10 (input)	P52/TI06/ TO06	SO10
			1	0/1 Note3	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission/ reception	SCK10 (input)	SI10	SO10
			0	1	0/1	0	1	0	1	1	×	× Note2	× Note2	Master CSI10 reception	SCK10 (output)	SI10	P53/TI13/ TO13
			1	0/1 Note3	0/1 Note3	1	0	0	1	× Note2	× Note2	0	1	Master CSI10 transmission	SCK10 (output)	P52/TI06/ TO06	SO10
			1	0/1 Note3	0/1 Note3	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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

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

Table 12-12. Relationship between register settings and pins (Channel 1 of unit 1: IIC11, SIIC1: 0, SIIC0: 0)

SE11 Note1	MD112	MD111	SOE11	SO11	CKO11	TXE11	RXE11	PM60	P60	PM61	P61	POM1	Operation mode	Pin Function	
														P60/TI20/TO20/ SCL11/INTP1	P61/TI21/TO21/ SDA11/INTP3
0	1	0	0	0/1 Note2	0/1 Note2	0	0	0	1	0	1	1	IIC11 start condition	SCL11	SDA11
						1	0								
						0	1								
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	1	IIC11 address field transmission	SCL11	SDA11
			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	1	IIC11 data transmission	SCL11	SDA11
			1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	1	IIC11 data reception	SCL11	SDA11
0			0	0/1 Note5	0/1 Note5	0	1	0	1	0	1	1	IIC11 stop condition	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, refer to **12.3 (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.

Table 12-13. Relationship between register settings and pins (Channel 1 of unit 1: IIC11, SIIC1 = 0, SIIC0 = 1)

SE11 **Note1**	MD112	MD111	SOE11	SO11	CKO11	TXE11	RXE11	PM30	P30	PM31	P31	POM2	Operation mode	Pin Function	
														P30/TI20/TO20/ SCL11	P31/TI21/TO21/ SDA11
0	1	0	0	0/1 **Note2**	0/1 **Note2**	0	0	0	1	0	1	1	IIC11 start condition	SCL11	SDA11
						1	0								
						0	1								
1			1	0/1 **Note4**	0/1 **Note4**	1	0	0	1	0	1	1	IIC11 address field transmission	SCL11	SDA11
			1	0/1 **Note4**	0/1 **Note4**	1	0	0	1	0	1	1	IIC11 data transmission	SCL11	SDA11
			1	0/1 **Note4**	0/1 **Note4**	0	1	0	1	0	1	1	IIC11 data reception	SCL11	SDA11
0			0	0/1 **Note5**	0/1 **Note5**	0	1	0	1	0	1	1	IIC11 stop condition	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, refer to **12.3 (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.

Table 12-14. Relationship between register settings and pins (Channel 1 of unit 1: IIC11, SIIC1 = 1, SIIC0 = 0)

SE11 Note1	MD112	MD111	SOE11	SO11	CKO11	TXE11	RXE11	PM136	P136	PM50	P50	POM5	Operation mode	Pin Function	
														P136/TI00/TO00/SCL11	P50/TI21/TO21/SDA11
0	1	0	0	0/1 Note2	0/1 Note2	0	0	0	1	0	1	1	IIC11 start condition	SCL11	SDA11
						1	0								
						0	1								
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	1	IIC11 address field transmission	SCL11	SDA11
			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	1	IIC11 data transmission	SCL11	SDA11
			1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	1	IIC11 data reception	SCL11	SDA11
0			0	0/1 Note5	0/1 Note5	0	1	0	1	0	1	1	IIC11 stop condition	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 or other alternate function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **12.3 (12) Serial output register m (SOm)**.

5. 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.

<R>

Table 12-15. Relationship between register settings and pins (Channel 0 and 1 of unit 0: UART0)

SE 00 Note1	SE 01 Note1	MD 002	MD 001	MD 012	MD 011	SOE 00	SO 00	TXE 00	RXE 01	PM11	P11	PM12	P12	Operation mode	Pin Function	
															P11/LRxD1/ INTPLR1/ SI00/RxD0/ TI11/TO11	P12/ SO00/TxD0/ TI12/TO12/ INTP2
0	0	×	×	×	×	0	1	0	0	× Note2	× Note2	× Note2	× Note2	Operation stop mode	P11/LRxD1/ INTPLR1/ SI00/RxD0/ TI11/TO11	P12/ SO00/TxD0/ TI12/TO12/ INTP2
0	1	×	×	0	1	0	1	0	1	1	×	× Note2	× Note2	UART0 reception	RxD0	P12/ TI12/TO12/ INTP2
1	0	0	1	×	×	1	0/1 Note3	1	0	×	×	0	1	UART0 transmission	P11/LRxD1/ INTPLR1/ TI11/TO11	TxD0
1	1	0	1	0	1	1	0/1 Note3	1	1	1	×	0	1	UART0 transmission/ reception	RxD0	TxD0

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 or other alternate function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to 12.3 (12) Serial output register m (SOm).

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

CHAPTER 13 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)

In the RL78/D1A, two asynchronous serial interface LIN-UART (UARTF) are provided.

13.1 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
- ○ Reception data/reception error detection function
  - 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)

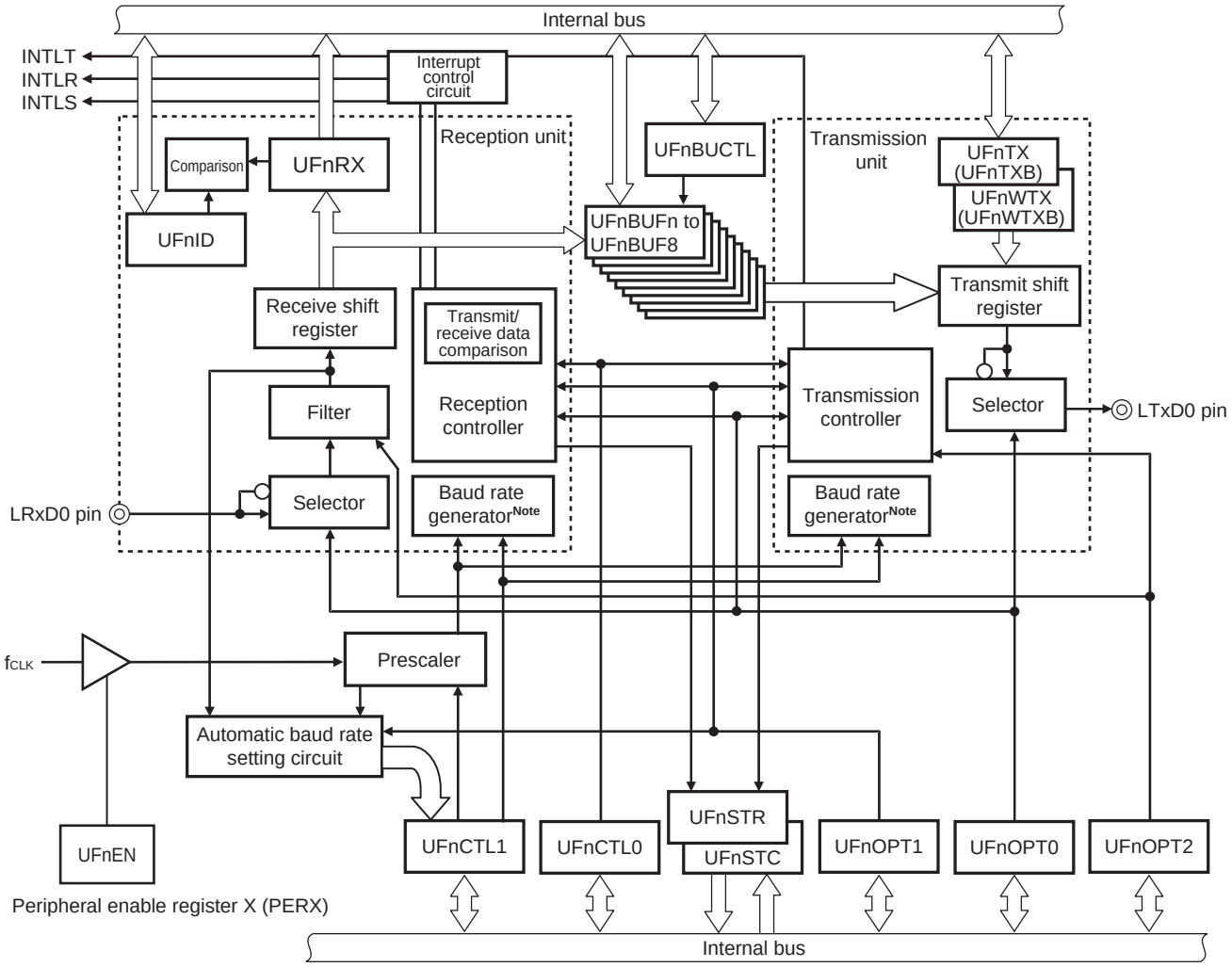
Remark n = 0, 1

○ Transmission/reception function in the 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.

13.2 Configuration

Figure 13-1. Block Diagram of Asynchronous Serial Interface LIN-UART



Note For the configuration of the baud rate generator, see **Figure 13-72 Configuration of Baud Rate Generator**.

Remark n = 0, 1

LIN-UART consists of the following hardware units.

Table 13-1. 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 0, 1 (STSEL0, STSEL1) Port mode register 1, 7, 13 (PM1, PM7, PM13)

Remark n = 0, 1

13.3 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.

Set PER0 by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 13-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	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.

(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.

Figure 13-3. 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 setting of the UFnTDL bit in the UFnOPT0 register is reflected in the LTxDn pin level, irrespective of the value of the UFnTXE bit. • When clearing the transmission enable bit (UFnCTL0.UFnTXE) after transmission completion, set (UFnOPT2.UFnITS = 1) a transmission interrupt upon transmission completion and confirm that the transmission interrupt has been generated, or clear the bit after having confirmed that the transmission status flag (UFnSTR.UFnTSF) has been cleared to “0” and communication has been completed.	

UFnRXE	Reception operation enable
0	Stops reception operation. An interrupt is not generated and received data is not stored.
1	Enables reception operation.

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 13-3. 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 **13.5.7 Parity types and operations**.

(3) LIN-UARTn control register 1 (UFnCTL1)

See **13.10 (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 13H.

Figure 13-4. 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.

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.
When in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), normal completion of BF reception can be confirmed by checking that the successful BF reception flag (UFnBSF) is set to 1 when a status interrupt is detected.
In BF reception enable mode during communication, a reception complete interrupt is not generated even by setting the BF reception trigger bit. However, the normal completion of BF reception can be confirmed also by checking that the UFnBRF flag is 0 when a status interrupt is detected after setting the bit.

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 13-4. Format of LIN-UARTn Option Register 0 (UFnOPT0) (2/3)

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. Setting both the next transmit data and the UFnBTT bit during data transmission is prohibited.
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).

Remark Before setting UFnOPT0.UFnBTT to “1” and starting BF communication, check that no data transfer is being processed (UFnSTR.UFnTSF = 0).

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”.

Remark n = 0, 1

Figure 13-4. 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 13.8.3 Expansion bit mode reception (with data comparison) for details.

Figure 13-5. 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".
3. The expansion bit is included in 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 **13.8.2 Expansion bit mode reception (no data comparison)** and **13.8.3 Expansion bit mode reception (with data comparison)** for details.

Figure 13-5. 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 **13.8.2 Expansion bit mode reception (no data comparison)** and **13.8.3 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 **13.7.3 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 **13.7.4 Automatic checksum function** for details.

Figure 13-5. 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 = 0B)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 **13.5.8 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 13-6. 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	0	0	0	0	0	0	UFnRXFL	UFnITS

(n = 0, 1)

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 **13.9 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 13-7. 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 **13.7.3 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 13-7. 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 **13.7.4 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 **13.7.2 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 **13.7.1 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 13-7. 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 **13.6.1 UART buffer mode transmission** and **13.7 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 **13.8.3 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 **13.8.2 Expansion bit mode reception (no data comparison)** and **13.8.3 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 13-7. Format of LIN-UARTn Status Register (UFnSTR) (4/6)

UFnTSF	Transmission status flag
0	A transmit operation is not performed. <Transmission stop condition> • When UFnCTL0.UFnTXE has been cleared to “0” • When there was no next transmit data after transmission completion, and at the same time, BF transmit trigger bit (UFnBTT) has not been set • 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
1	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
• 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. **Note** Only during BF period	

UFnRSF	Reception status flag
0	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
1	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)
To initialize the reception unit, check that UFnRSF is “0” before performing initialization. If initialization is performed while UFnRSF = 1, the reception will be aborted midway.	

Figure 13-7. 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 13.5.8 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 depends on the settings of the UFnPS1 and UFnPS0 bits.
- 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 13-7. 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 13-8. 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 13-8. 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 **13.5.1 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 13-9. 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	UFnTX.8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnTX.7	UFnTX.6	UFnTX.5	UFnTX.4	UFnTX.3	UFnTX.2	UFnTX.1	UFnTX.0

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 UFnTX.8 bit.

3. Writing to the UFnTX register is prohibited when using the UFnBUF0 to UFnBUF8 registers.

4. When using automatic checksum function, set 0000H in the UFnTX register before starting communication.

Remarks 1. When UFnOPT2.UFnITS is “0”, successive transmission can be performed by writing the next transmit data before transmission is completed, after a transmission interrupt request signal (INTLTn) has been generated.

2. 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. For detail of the transmit data format, see **13.5.1 Data format**.

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.

Figure 13-10. 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	UFnTX.7	UFnTX.6	UFnTX.5	UFnTX.4	UFnTX.3	UFnTX.2	UFnTX.1	UFnTX.0

(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 UFnTX.8 bit of UFnTX register.

3. Writing to the UFnTXB register is prohibited when using the UFnBUF0 to UFnBUF8 registers.

4. When using automatic checksum function, set 00H in the UFnTXB register before starting communication.

Remark When UFnOPT2.UFnITS is “0”, successive transmission can be performed by writing the next transmit data before transmission is completed, after a transmission interrupt request signal (INTLTn) has been generated.

(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 **13.5.11 Transmission start wait function** for details.

The UFnWTX register value will be read when the UFnWTX register has been read.

Reset input sets this register to 0000H.

Figure 13-11. 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	UFnWTX.8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnWTX.7	UFnWTX.6	UFnWTX.5	UFnWTX.4	UFnWTX.3	UFnWTX.2	UFnWTX.1	UFnWTX.0

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 UFnWTX.8 bit.

3. Writing to the UFnWTX register is prohibited when using the UFnBUF0 to UFnBUF8 registers.

Remark The UFnWTX.8 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 **13.5.11 Transmission start wait function** for details.

The UFnTXB register value will be read when the UFnWTXB register has been read.

Reset input sets this register to 00H.

Figure 13-12. 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	UFnWTX.7	UFnWTX.6	UFnWTX.5	UFnWTX.4	UFnWTX.3	UFnWTX.2	UFnWTX.1	UFnWTX.0

(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 UFnWTX.8 bit of UFnWTX register.

3. Writing to the UFnWTXB 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).

(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.

Figure 13-13. Format of LIN-UARTn Receive Data Register (UFnRX)

Address: FFF4AH, FFF4BH (UF0RX), FFF4EH, FFF4FH (UF1RX) After reset: 0000H R

	15	14	13	12	11	10	9	8
UFnRX	0	0	0	0	0	0	0	UFnRX.8
(n = 0, 1)	7	6	5	4	3	2	1	0
	UFnRX.7	UFnRX.6	UFnRX.5	UFnRX.4	UFnRX.3	UFnRX.2	UFnRX.1	UFnRX.0

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.

Caution “0” is written to the UFnRX8 bit of UFnRX register when writing data to the UFnRXB register.

Remark The UFnRX.8 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 13-14. 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	UFnRX.7	UFnRX.6	UFnRX.5	UFnRX.4	UFnRX.3	UFnRX.2	UFnRX.1	UFnRX.0

(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 **13.7 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 **13.8.3 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 13-15. 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	UFnID.7	UFnID.6	UFnID.5	UFnID.4	UFnID.3	UFnID.2	UFnID.1	UFnID.0

(n = 0, 1)

Caution Set to 00H before starting communication when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B). Writing is prohibited during communication operation 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.

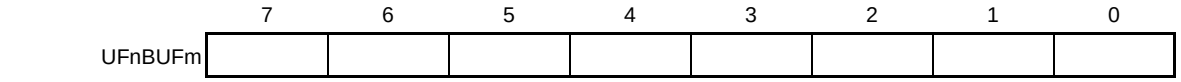
See **13.6.1 UART buffer mode transmission** and **13.7 LIN Communication Automatic Baud Rate Mode** for details.

These registers can be read or written in 8-bit units.

Reset input sets these registers to 00H.

Figure 13-16. 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)



(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 **13.6.1 UART buffer mode transmission** and **13.7 LIN Communication Automatic Baud Rate Mode** for details.

Reset input sets this register to 0000H.

Figure 13-17. 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

UFnTW	Transmission start wait bit
0	Starts transmission immediately when buffer data transmission is requested.
1	Delays starting of transmission until completion of stop bit of reception when buffer data transmission is requested.

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 **13.5.11 Transmission start wait function** and **13.7 LIN Communication Automatic Baud Rate Mode** for details.

Cautions 1. Setting this bit is prohibited except when switching to response transmission after header reception.

2. The UFnTW bit becomes valid at the same time as the UFnTRQ bit is set (1).

UFnCON	Successive selection bit
0	The data group to be transmitted or received next is the last data group.
1	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.)

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 **13.7.5 Multi-byte response transmission/reception function** for details.

Cautions 1. Setting this bit is prohibited except when the multi-byte transmission/reception function is used.

2. Set the UFnCON bit at the same time as setting UFnNO, UFnRRQ, and UFnTRQ for 16-bit access.

UFnECS	Enhanced checksum selection bit
0	Classic checksum (used only for data byte calculation)
1	Enhanced checksum (used for calculating data byte + PID byte)

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 **13.7.4 Automatic checksum function** for details.

Figure 13-17. 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.

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).

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 **13.7 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 **13.6.1 UART buffer mode transmission** and **13.7 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). When automatic checksum function is enabled, the checksum bits (one byte) need not be included in buffer length.
See **13.6.1 UART buffer mode transmission** and **13.7 LIN Communication Automatic Baud Rate Mode** for details.

(18) Serial communication pin select registers 0, 1 (STSEL0, STSEL1)

The STSEL0, 1 register are used to switch the input source to the serial array unit and the LIN-UARTn communication pins.

This register can be read or written in 1-bit units or 8-bit units.

Figure 13-18. Format of STSEL0 Register

Address: FFF3C After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL0	0	SCSI100	0	SCSI010	SCSI001	SCSI000	SUARTF1	SUARTF0

SUARTF0	Communication pin selection of UARTF0	
	LTxD0	LRxD0
0	P71	P70
1	P15	P14

SUARTF1	Communication pin selection of UARTF1	
	LTxD1	LRxD1
0	P10	P11
1	P131	P132

SCSI001	SCSI000	CSI00 communication pin selection		
		SCK00	SI00	SO00
0	0	P10	P11	P12
0	1	P04	P03	P02
1	0	P34	P33	P32
Other than the above		Setting prohibited (same as “00” setting)		

SCSI010	CSI01 communication pin selection		
	SCK01	SI01	SO01
0	P74	P75	P13
1	P56	P55	P54

SCSI100	CSI10 communication pin selection		
	SCK10	SI10	SO10
0	P133	P132	P131
1	P51	P52	P53

Figure 13-19. Format of STSEL1 Register

Address: FFF3D After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL1	SIIC1	SIIC0	0	0	SCAN1	SCAN0	TMCAN1	TMCAN0

TMCAN0	Input source switch of TAU unit1 CH4
0	Input from TI14 (after selected by TIS141~0 bits)
1	TSOUT of aFCAN0 (CAN0 time stamp function)

TMCAN1	Input source switch of TAU unit1 CH5
0	Input from TI15 (after selected by TIS151~0 bits)
1	TSOUT of aFCAN1 (CAN1 time stamp function)

SCAN0	Communication pin selection of aFCAN0	
	CTxD0	CRxD0
0	P71	P70
1	P00	P01

SCAN1	Communication pin selection of aFCAN1	
	CTxD1	CRxD1
0	P62	P63
1	P134	P135

SIIC1	SIIC0	Communication pin selection of IIC11	
		SCL11	SDA11
0	0	P60	P61
0	1	P30	P31
1	0	P136	P50
Other than the above		Setting prohibited	

(19) Port mode registers 1, 7, 13 (PM1, PM7, PM13)

The PM1, PM7 and PM13 registers are used to set ports 1, 7, and 13 to input or output in 1-bit units.

The PM1, PM7 and PM13 registers can be set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH (PM13 is set to FEH).

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 **1.3 Ordering Information** and **2.1 Pin Function List**.

2. See **CHAPTER 4 PORT FUNCTIONS** for port settings.

Figure 13-20. Format of Port Mode Registers 1, 7, 13 (PM1, PM7, PM13)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF2DH After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	PM136	PM135	PM134	PM133	PM132	PM131	0

PMmn	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)

13.4 Interrupt Request Signals

The following three interrupt request signals are generated from LIN-UARTn.

- LIN-UARTn reception status interrupt (INTLSn)
- LIN-UARTn reception interrupt (INTLRn)
- LIN-UARTn transmission interrupt (INTLTn)

Table 13-2 shows the default priority order of these three interrupt request signals.

Table 13-2. Interrupts and Their Default Priorities

Interrupt	Default Priority
Status	Low
Reception complete	|
Transmission start/complete	High

(1) LIN-UARTn reception status interrupt (INTLSn)

LIN-UARTn reception status interrupt 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 **13.5.10 LIN-UART reception status interrupt generation sources** for details.

(2) LIN-UARTn reception interrupt (INTLRn)

LIN-UARTn reception interrupt 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, LIN-UARTn reception interrupt is not generated, but LIN-UARTn reception status interrupt is generated.

LIN-UARTn reception interrupt is not generated in the reception disabled status.

- If expansion bit operation is enabled (UFnCL = UFnEBE = 1) and expansion bit data comparison is disabled (UFnEBC = 0), LIN-UART reception interrupt 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), LIN-UART reception itnerrupt is generated.
- 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.

(3) LIN-UARTn transmission interrupt (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.

- When in automatic baud rate mode (UFnMD1, UFnMD0 = 11B), a transmission complete interrupt request signal is generated at the start of transmission of the last byte of a response.

Remark n = 0, 1

13.5 Operation

13.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 13-21, 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.

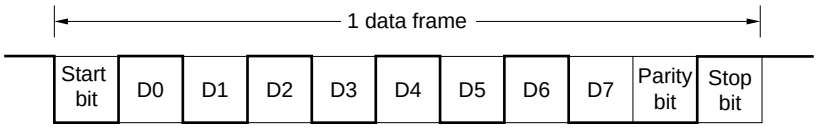
Moreover, the UFnTDL bit and UFnRDL bit of the UFnOPT0 register are used to control UART output/inverted output for the LTxDn pin and UART input/inverted input for the LRxDn pin, respectively.

Remark n = 0, 1

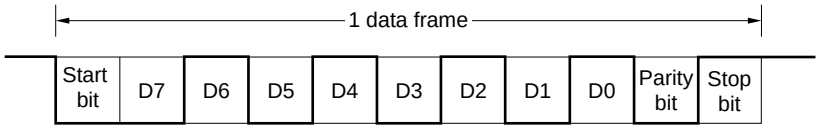
- Start bit.....1 bit
- Character bits.....7 bits/8 bits
- Expansion bit .....1 bit
- Parity bit.....Even parity/odd parity/0 parity/no parity
- Stop bit.....1 bit/2 bits
- Transmission/reception level setting .....Forward/inversion
- Transmission/reception direction setting .....Forward/inversion

Figure 13-21. Format of LIN-UART Transmit/Receive Data (1/2)

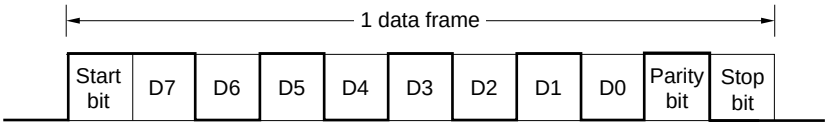
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



(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

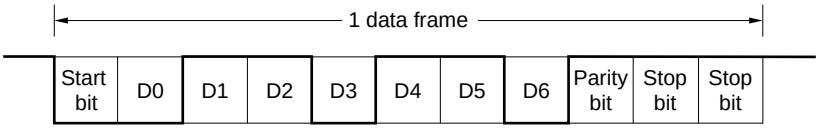
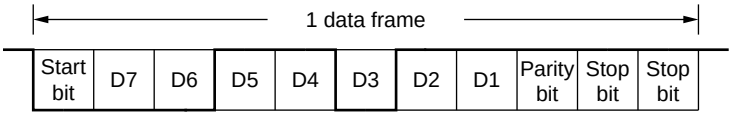
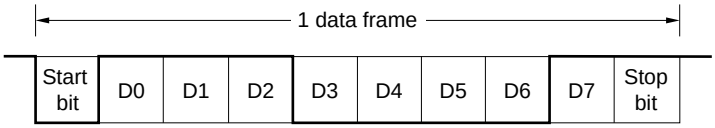


Figure 13-21. Format of LIN-UART Transmit/Receive Data (2/2)

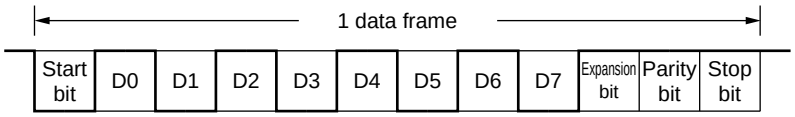
(e) 7-bit data length, MSB first, odd parity, 2 stop bits, transfer data: 36H



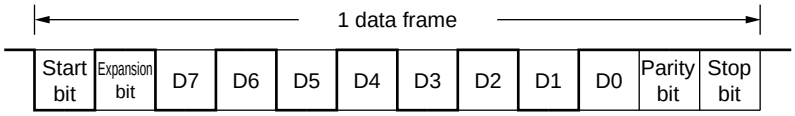
(f) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



(g) 8-bit data length, LSB first, even parity, expansion bit: enabled, 1 stop bit, transfer data: 155H



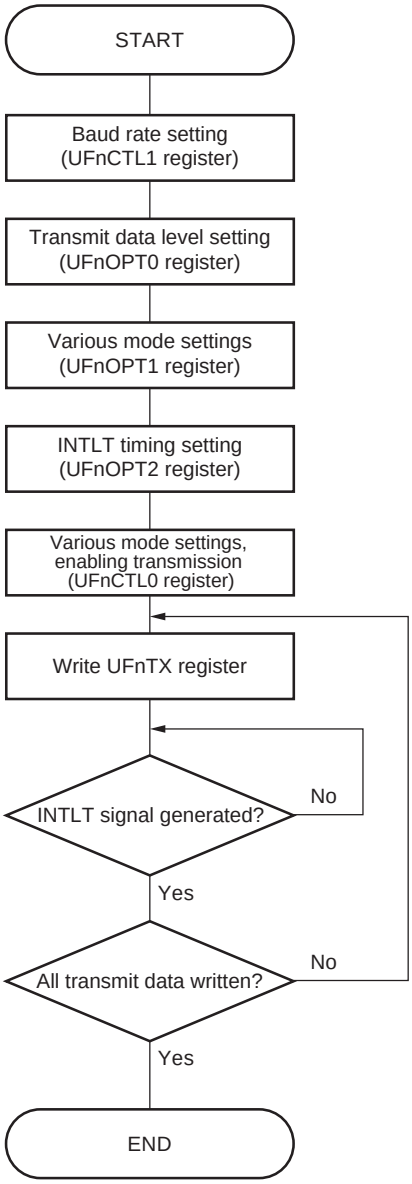
(h) 8-bit data length, MSB first, even parity, expansion bit: enabled, 1 stop bit, transfer data: 155H



13.5.2 Data transmission

Figure 13-22 shows the procedure for transmitting data.

Figure 13-22. 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).

Remarks1. See **(2)** of **13.11 Cautions on Use for details of starting LIN-UART**.

2. n = 0,1

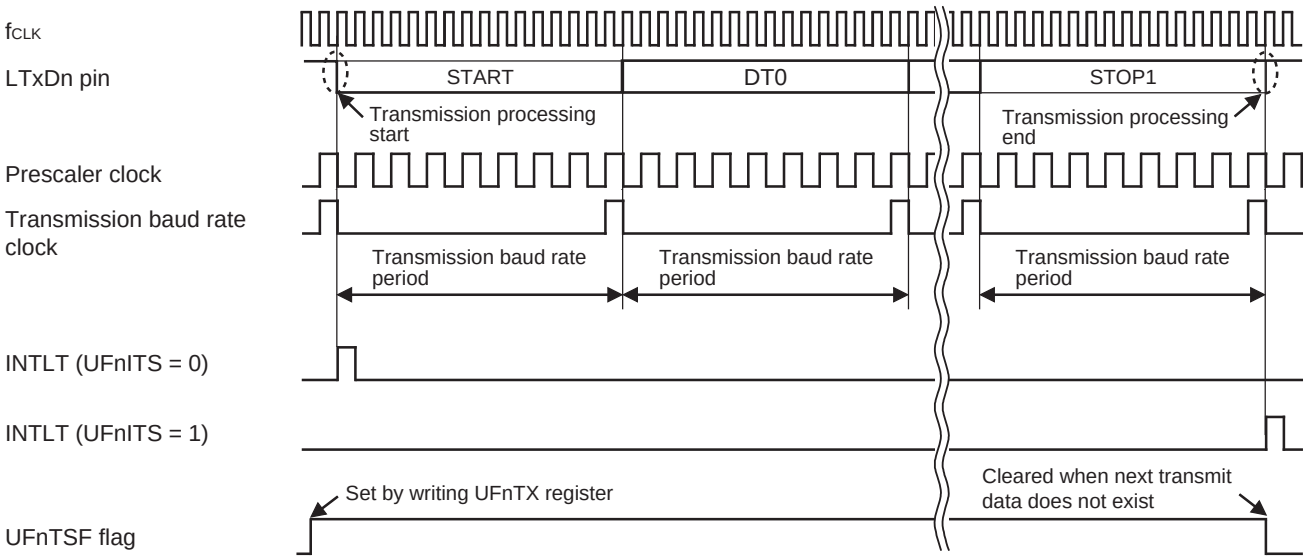
A transmission operation is started by writing transmit data to the transmit data register (UFnTX).

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 13-23. Data Transmission Timing Chart



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 13-24. Diagram of Timing When Starting Successive Transmission (UFnITS = 0)

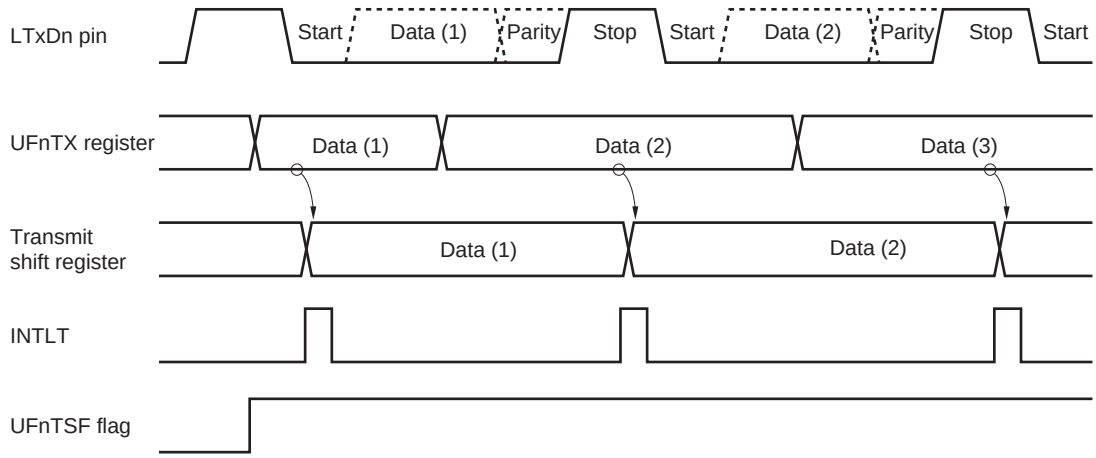
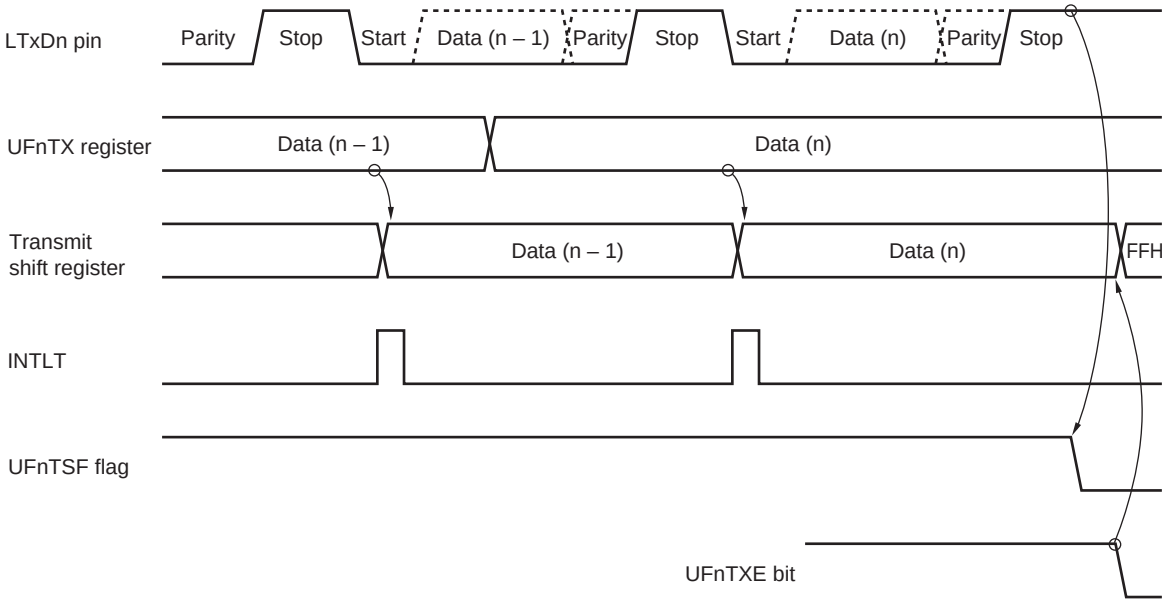


Figure 13-25. Diagram of Timing When Ending Successive Transmission (UFnITS = 0)

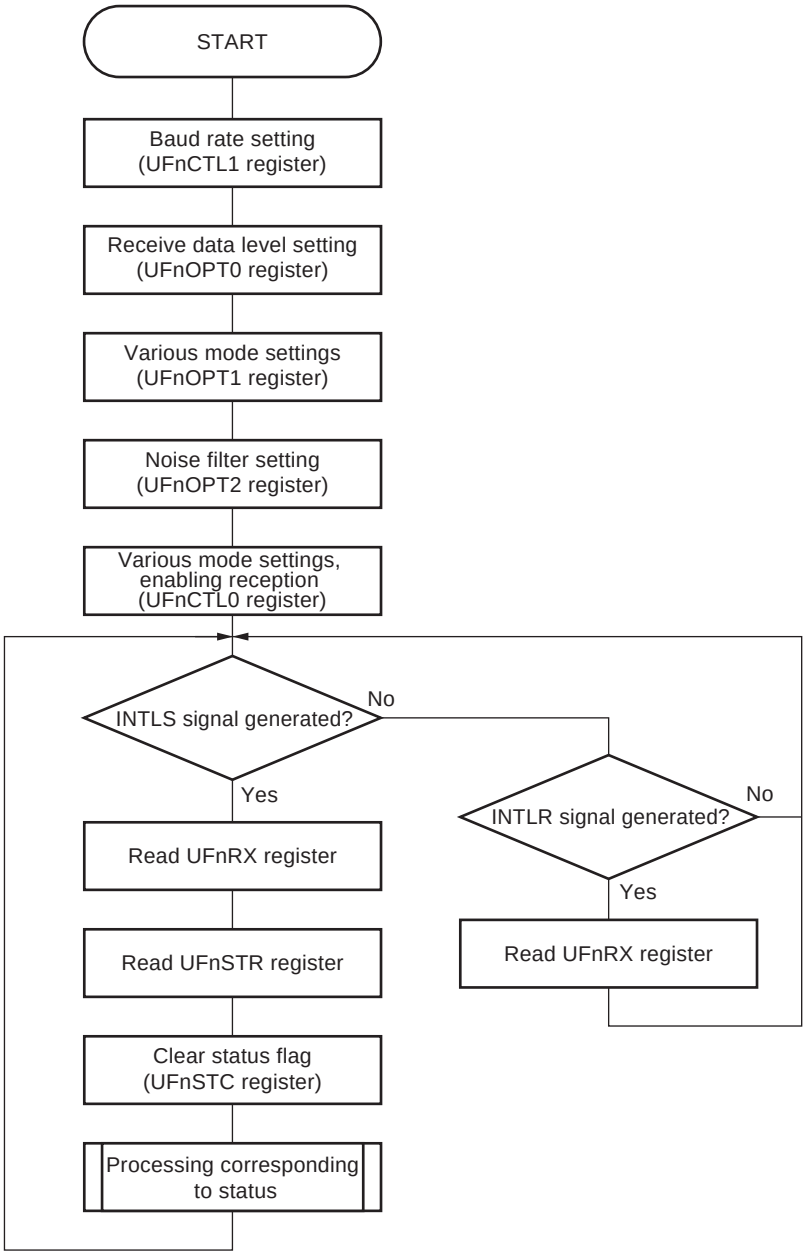


Remark n = 0, 1

13.5.3 Data reception

Figure 13-26 shows the procedure for receiving data.

Figure 13-26. 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.

Remarks1. See **(2)** of **13.11 Cautions on Use for details of starting LIN-UART**.

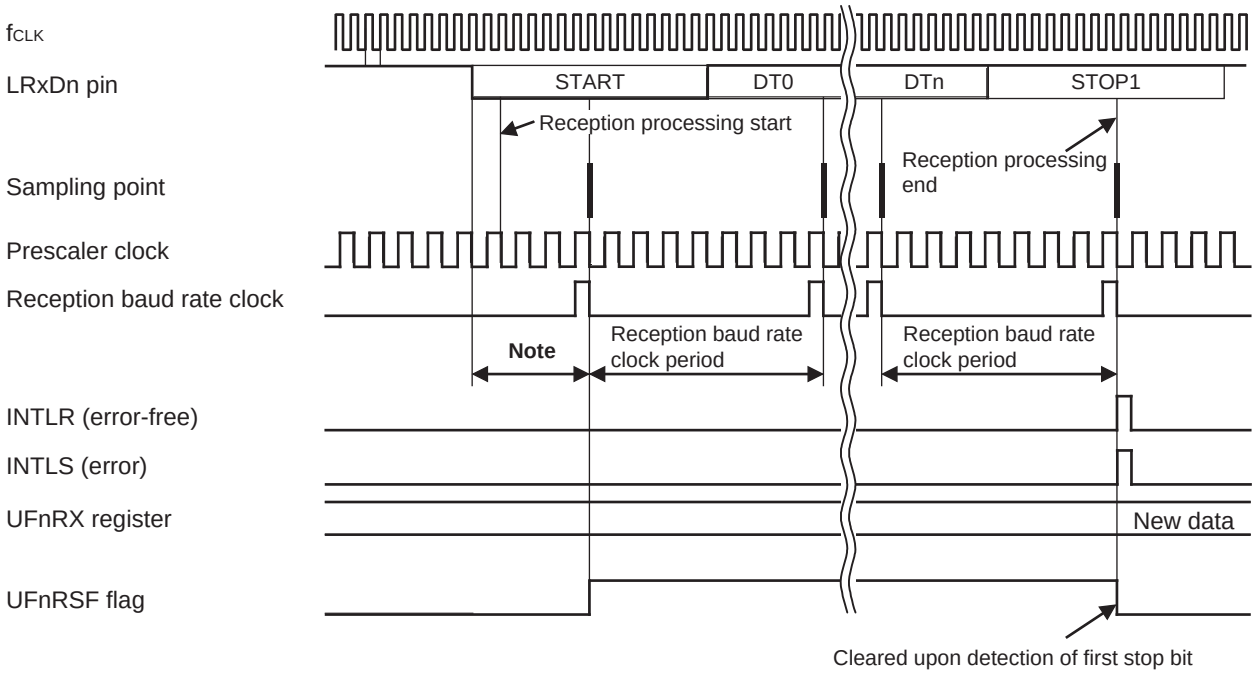
2. n = 0, 1

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 13-27. Data Reception Timing Chart



Note One-half the reception baud rate clock period

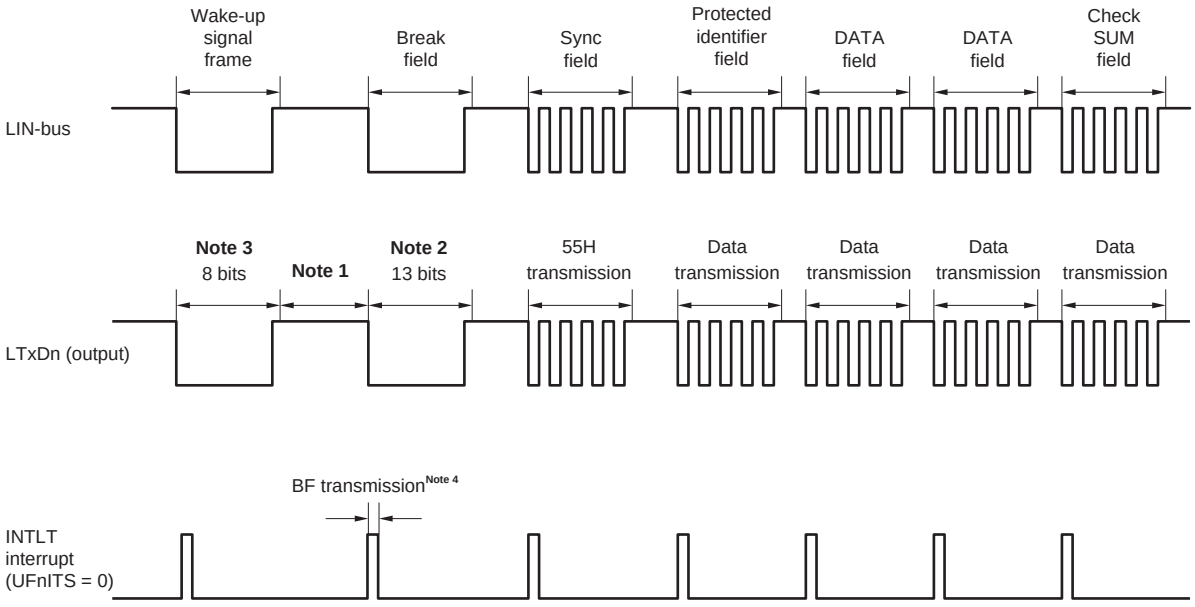
Cautions
1. **The start bit is not recognized when a high level is detected at a timing of half the reception baud rate clock period after the falling edge of the LRxDn pin was detected.**
2. **A reception always operates with the number of stop bits as 1. At that time, the second stop bit is ignored.**
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

13.5.4 BF transmission/reception format

The RL78/D1A has a BF (Break Field) transmission/reception control function to enable use of the LIN (Local Interconnect Network) function.

Figure 13-28. 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.

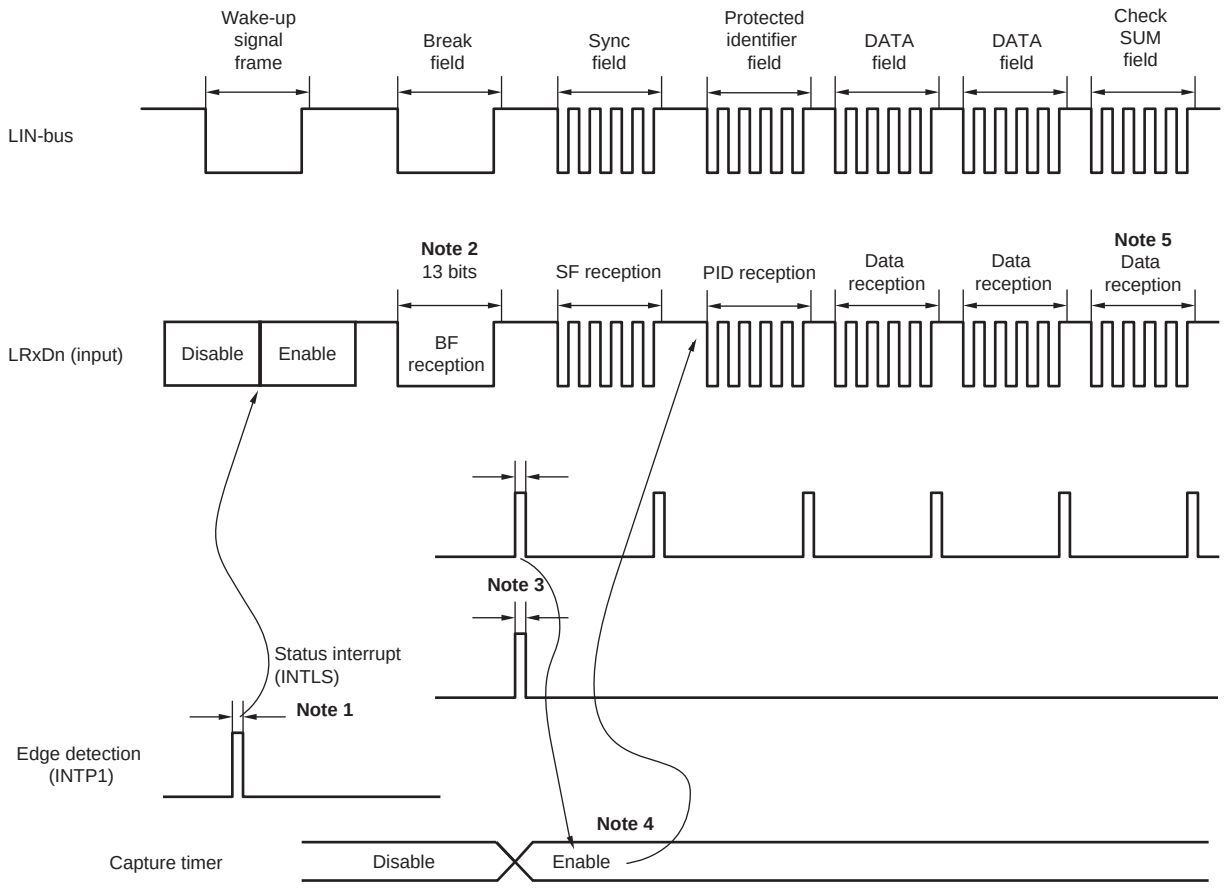
3. 80H transfer in the 8-bit mode or BF transmission is substituted for the wakeup signal.

4. The LIN-UART transmission interrupt (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 the LIN-UART transmission interrupt is always generated.

Remarks1. Figure 13-28 shows the LIN transmission manipulation outline when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B). See **13.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

2. n =0, 1

Figure 13-29. LIN Reception Manipulation Outline



Notes

1. A wakeup signal is detected by detecting the interrupt edge of a pin (INTP1). After having received the wakeup signal, enable LIN-UARTn, enable reception operation, and set the BF reception trigger bit if needed.
2. If a BF reception of at least 11 bits is detected, the BF reception is judged to be ended normally.
3. When BF reception has ended normally in normal UART mode (UFnMD1, UFnMD0 = 00B), a reception complete interrupt request signal (INTLR) is generated. When BF reception has ended normally in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), a status interrupt request signal (INTLSn) is generated, 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. Moreover, data transfer from the receive shift register to the receive data register (UFnRX) is also not performed. At this time, UFnRX retains the previous value.
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.

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

Caution With the sync field, the transfer baud rate is calculated using the capture function of the TAU. At this point, stop reception operation to stop generation of a reception interrupt in the LIN-UARTn.

Remarks1. See **13.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).

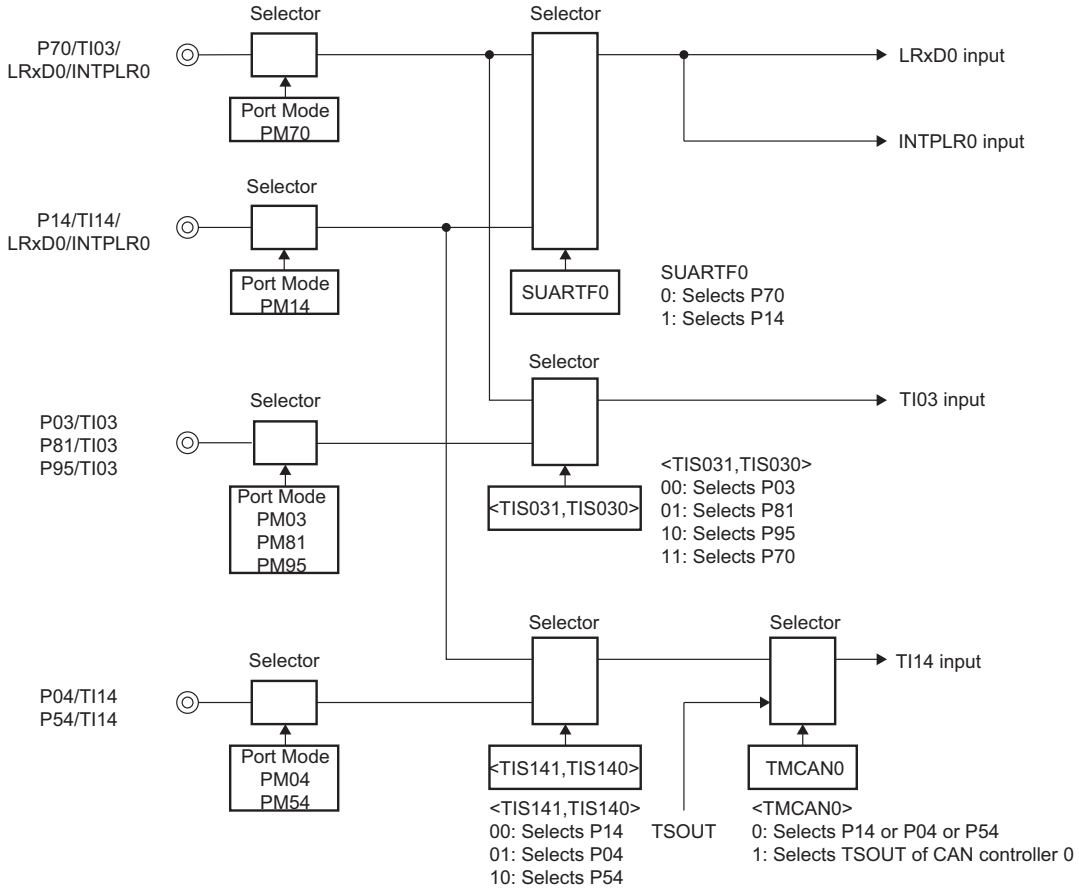
2. n = 0, 1

Figure 13-30 shows the port configurations for LIN reception manipulation.

Wakeup signals transmitted from the LIN master are received via INTP1 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).

(1) LIN-UART0

Figure 13-30. Port Configuration of LIN Reception Manipulation



Remarks 1. Figure 13-30 shows the port configuration when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B).

2. TMCAN0: Bit 0 of the serial communication pin select register 1 (STSEL1) (see **Figure 13-19**)

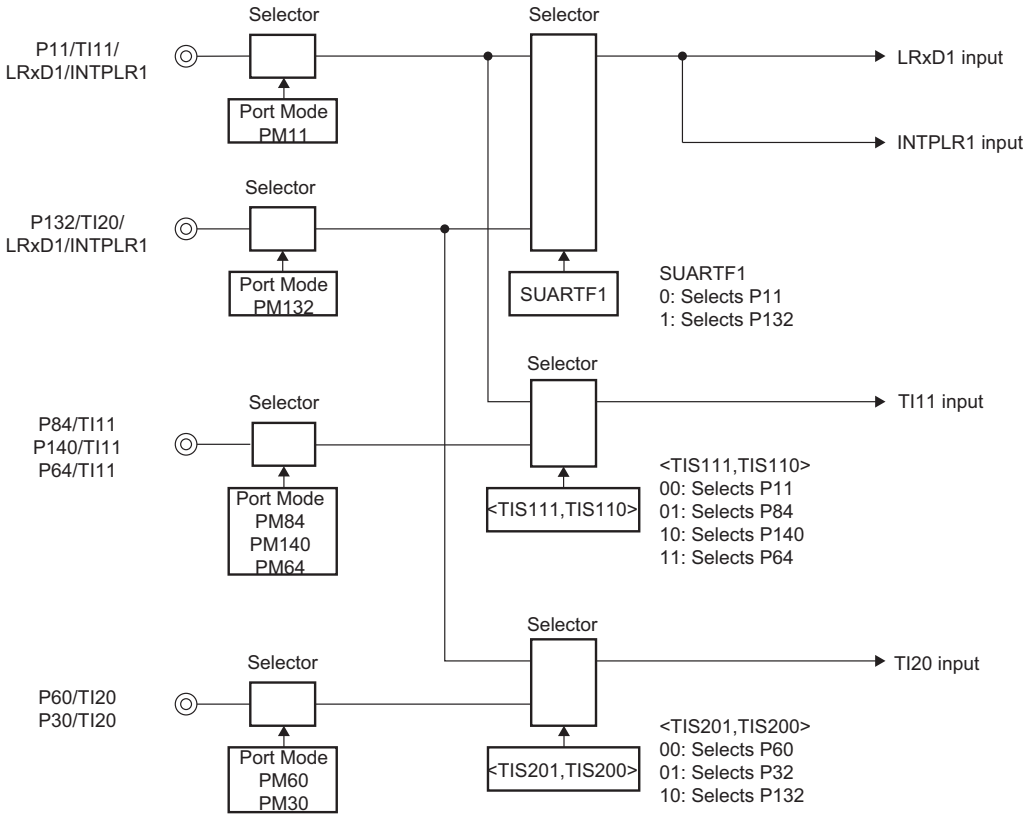
A summary of the peripheral functions to be used in LIN communication operation is given below.

<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 4 (TI14) of the timer array unit 1 (TAU1); 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 TI14 input edges in capture mode)
- Asynchronous serial interface LIN-UART0

(2) LIN-UART1

Figure 13-31. Port Configuration of LIN Reception Manipulation



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
- Asynchronous serial interface LIN-UART1

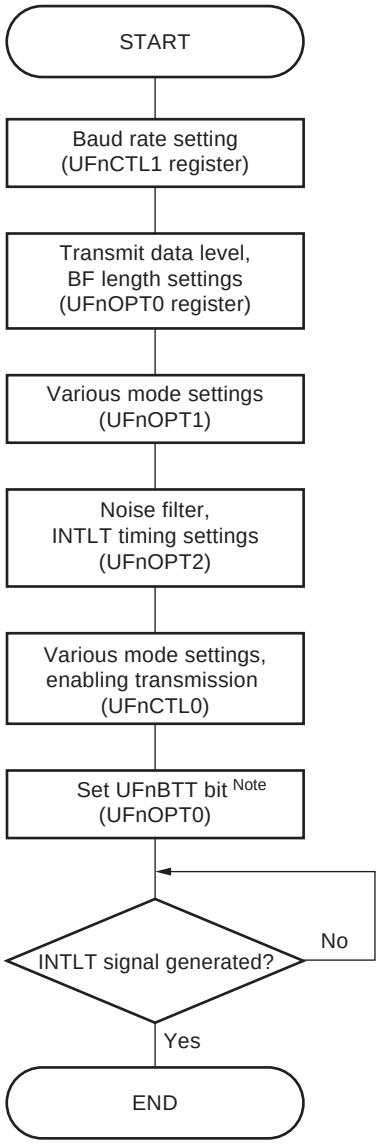
Remarks 1. Figure 13-31 shows the port configuration when in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B).

2. The above is the port configuration of LIN reception manipulation for R5F10DPJXFB.
The port configuration of LIN reception manipulation differs in depending on the product.

13.5.5 BF transmission

Figure 13-32 describes the processing of BF transmission in LIN communication.

Figure 13-32. BF Transmission Processing Flow



Note **In normal UART mode (UFnMD1, UFnMD0 = 00B), set the UFnBRT bit at the same time as setting the UFnBTT bit.**

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).
The LIN-UART transmission interrupt is generated when starting transmission (UFnITS = 0).

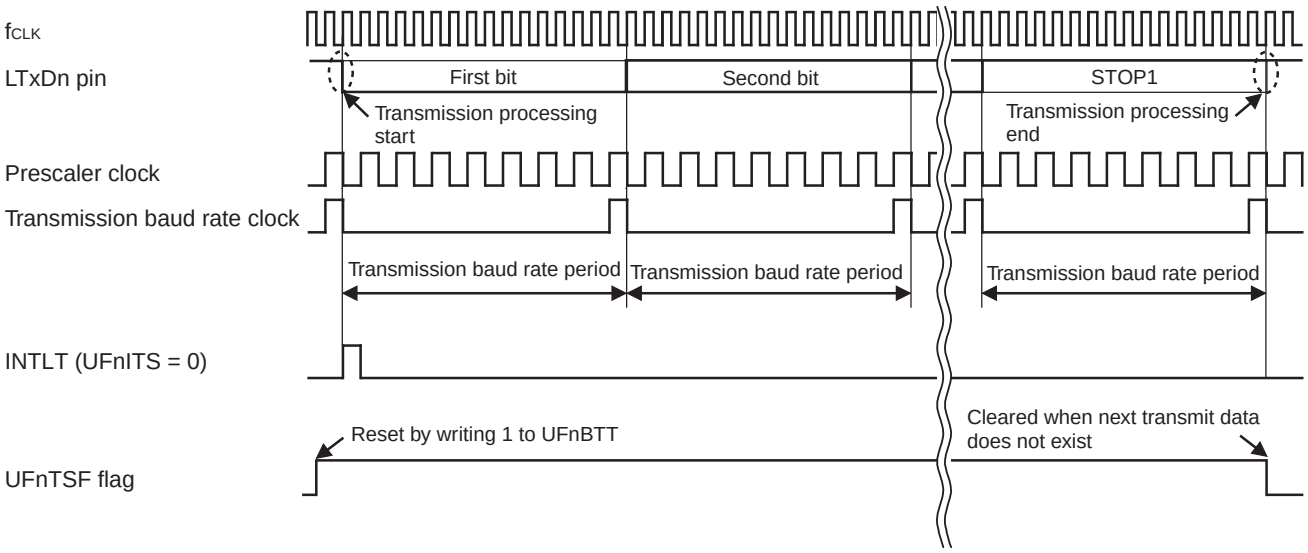
Remarks 1. See **(2)** of **13.11 Cautions on Use for details of starting LIN-UART**.

2. n = 0, 1

A BF transmission operation is started when a BF transmission trigger (UFnBTT) is set. 13 to 20 bits of low level (the length specified by the BF length selection bits (UFnBLS2 to UFnBLS0)) is output to the LTxDn pin. LIN-UARTn transmission interrupt (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 LIN-UARTn reception interrupt (INTLRn) during the BF transmission or the LIN-UARTn reception status interrupt (INTLSn).

Figure 13-33. BF Transmission Timing Example



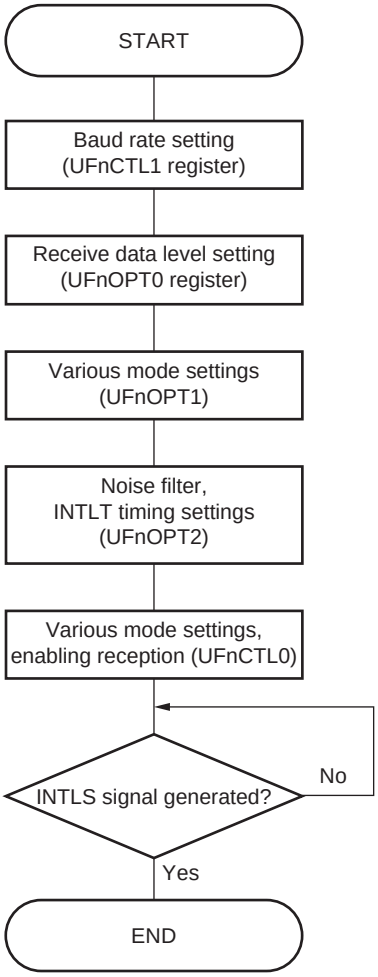
Caution When the stop bit length is set to 2 bits (UFnSL = 1), the transmission status flag (UFnTSF) is cleared when transmission of the second stop bit has been completed.

Remark n = 0, 1

13.5.6 BF reception

Figure 13-34 describes the processing of BF reception in LIN communication.

Figure 13-34. 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 generated when starting transmission (UFnITS = 0).
BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B) as the mode.

Remarks 1. Figure 13-34 shows the reception processing flow of LIN communication in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B).
See **13.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UFnMD1, UFnMD0 = 11B).
2. See **(2)** of **13.11 Cautions on Use for details of starting LIN-UART**.
3. n = 0, 1

When the BF reception trigger bit (UFnBRT) is set, a successful BF reception wait state (UFnBRF = 1) is entered, the LRxDn input level is monitored, and start bit detection is performed.

When the falling edge of the LRxDn input is detected, the BF length is measured by counting up the internal counter until a rising edge is detected. If the BF length is 11 bits or more when a rising edge is detected, BF reception is judged as being normal, and BF reception ends. When ending BF reception, a successful BF reception flag (UFnBSF) is set at the same time as generation of the LIN-UARTn reception status interrupt (INTLSn).

In automatic baud rate mode, detection of overrun, parity, and framing errors (UFnOVE, UFnPE, UFnFE) is limited. 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, the error status flag (UFnSTR) is set at the same time as generation of the status interrupt request signal (INTLSn).

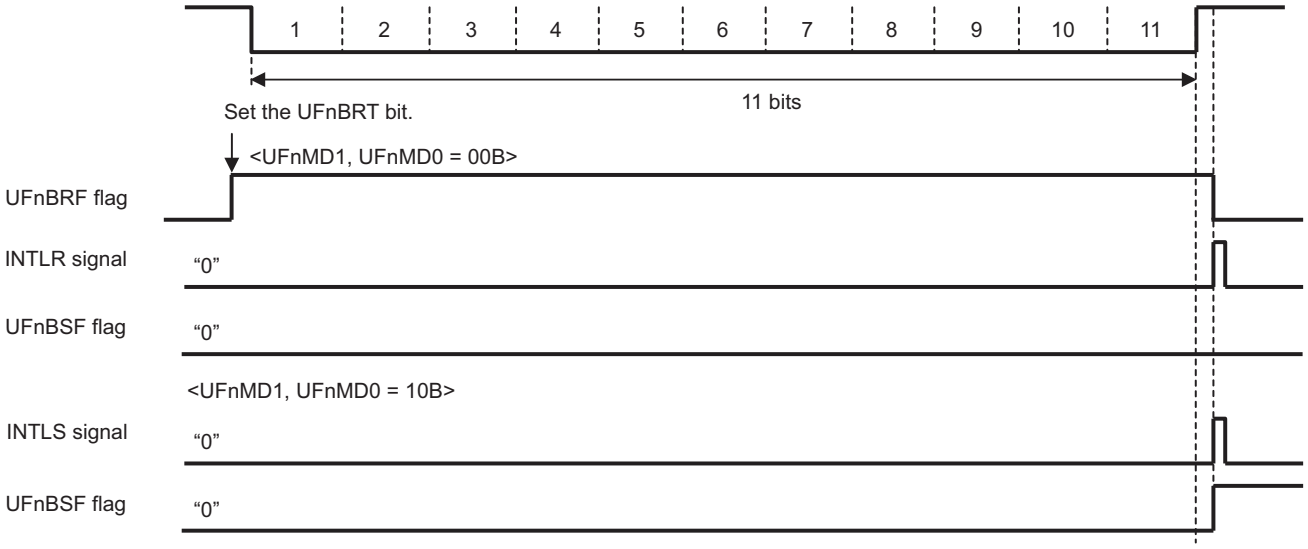
When performing a transmission for which a data consistency check is enabled (UFnDCS = 1), a data consistency error flag (UFnDCE) is set and LIN-UARTn reception status interrupt (INTLSn) is output when a mismatch between the transmit data and receive data is detected, regardless of whether BF reception is performed successfully or fails. At that time, INTLRn is not output.

When in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), LIN-UART can detect a new BF reception even during data communication or in automatic baud rate mode. See **13.5.9 (2) BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B)** for details.

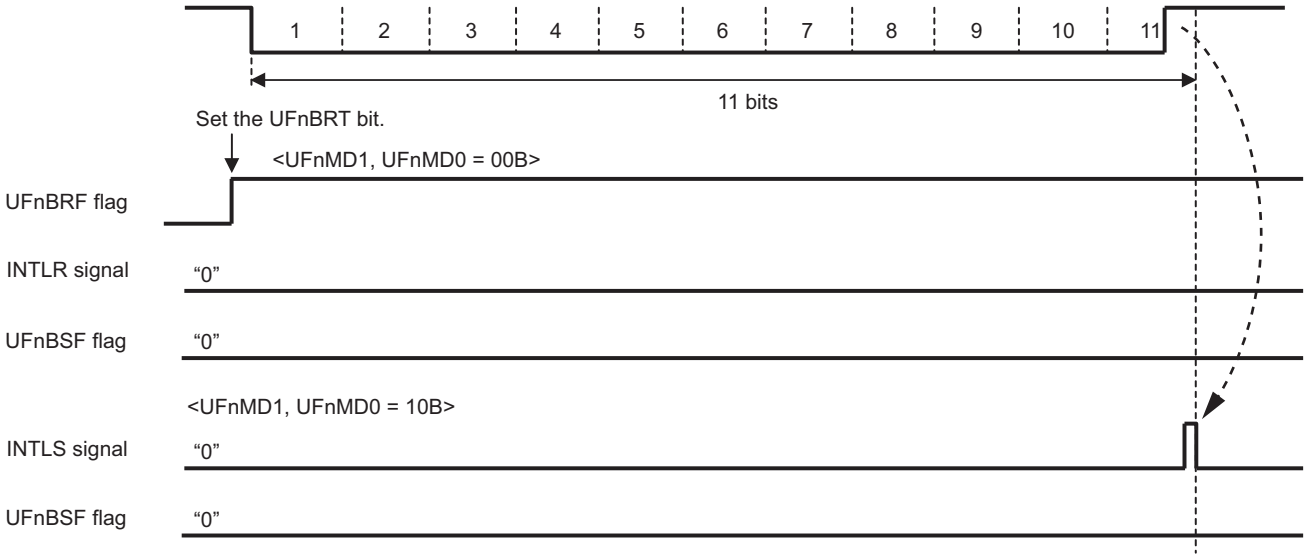
Remark n = 0, 1

Figure 13-35. 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 reset by setting the UFnBRT bit to "1" and cleared upon normal BF reception. In BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), the bit is reset or cleared in the same way as described above.

Remark n = 0, 1

13.5.7 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.

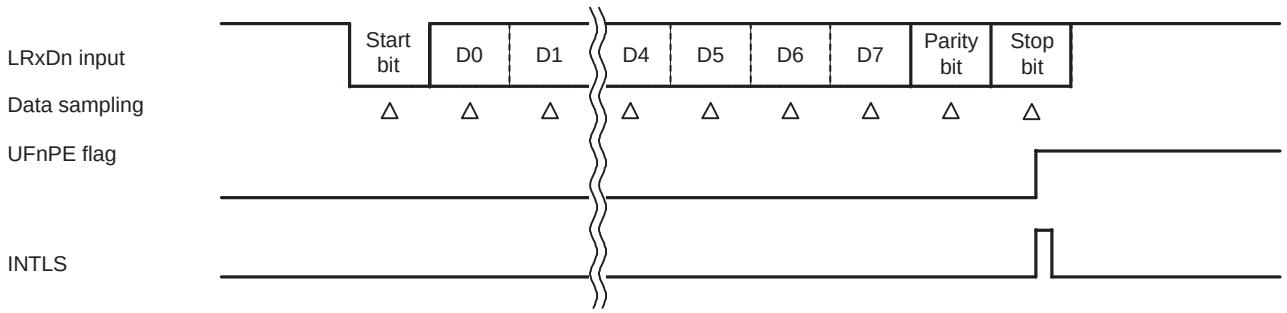
During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(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 13-36. Parity Error Occurrence Timing



13.5.8 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) must be set, after the end of transmission has been confirmed (UFnTSF = 0) and the data consistency error flag (UFnDCE) has been cleared or the UFnEN bit of the PERX register has been cleared and then set. When a buffer is used, communication is stopped even if data not transferred remains in the buffer.

When reception is disabled (UFnRXE = 0), storing receive data and thereby generating LIN-UARTn reception interrupt (INTLRn) as well as setting UFnBSF, UFnFE, and UFnOVE and thereby generating LIN-UARTn reception status interrupt (INTLSn) are not performed since the reception operation itself is not performed. Consequently, receive data is not required to be read.

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.

Remark n = 0, 1

(1) Mismatch between transmission and reception signals

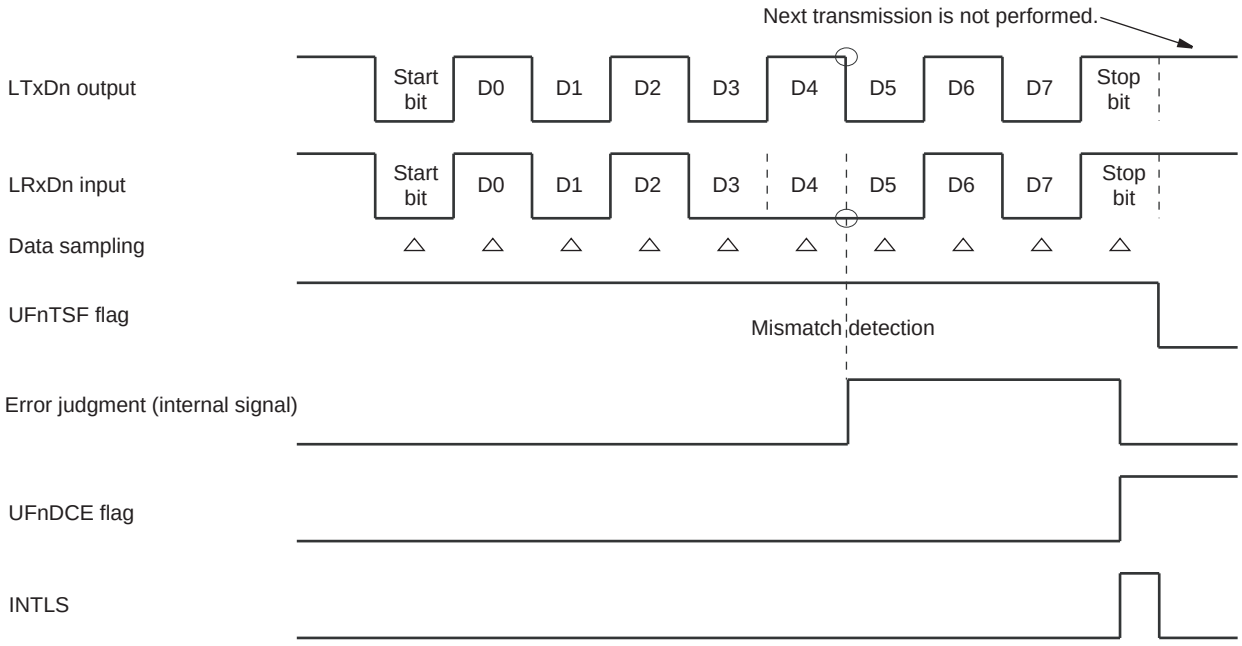
Serial transmission and reception signals are compared during data (or BF) transmission, a detected mismatch is judged as being abnormal, and the UFnDCE bit is set (1) at the same time a status interrupt (INTLSn) is generated.

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.

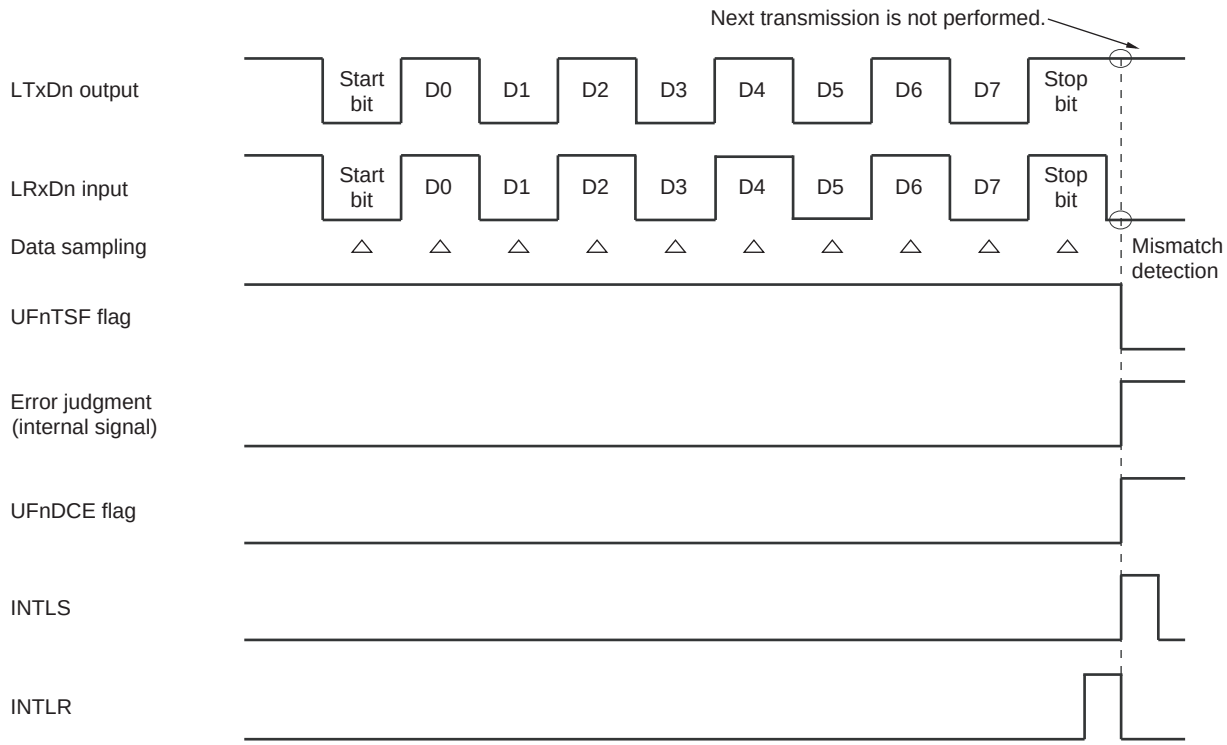
A consistency check is not performed for the second stop bit, even if the stop bit length is specified as two bits by using the stop bit length select bit (UFnSL).

Figure 13-37. Data Consistency Error Occurrence Timing Example 1 (UFnBRF = 0)



Remark n = 0, 1

Figure 13-38. Data Consistency Error Occurrence Timing Example 2 (UFnBRF = 0)

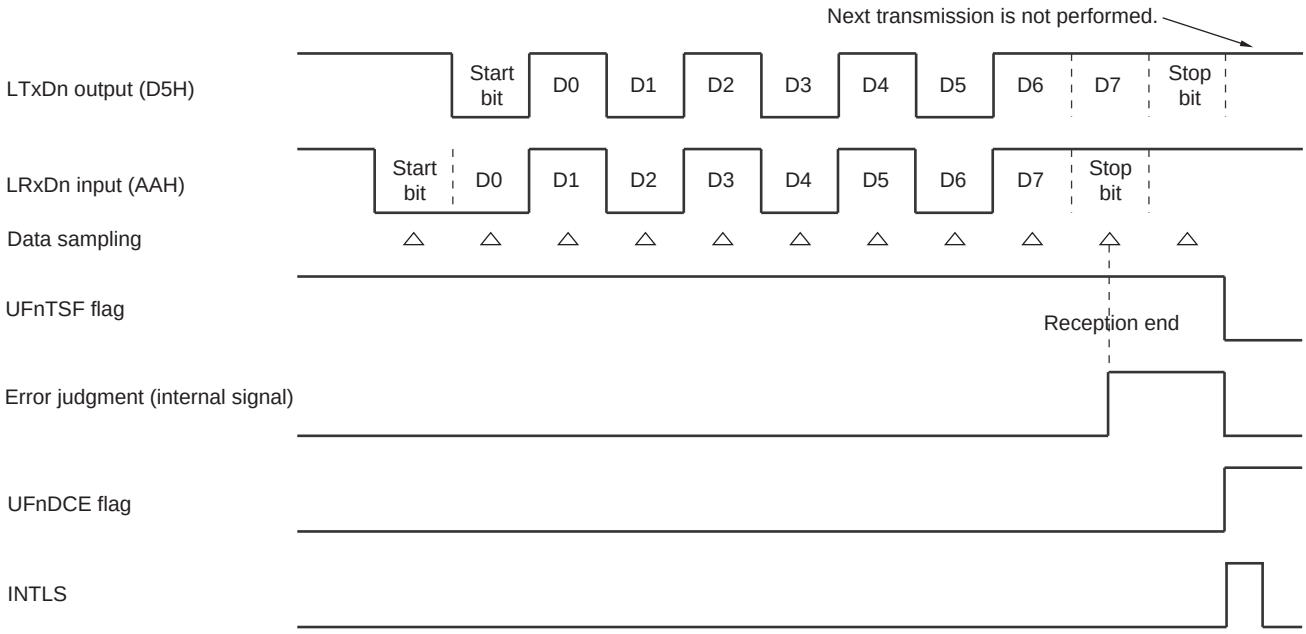


Remark n = 0, 1

(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 13-39. Timing Example of Consistency Error Occurrence due to Early End of Reception Processing



Remark n = 0, 1

13.5.9 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). When BF reception has been successfully completed, a reception complete interrupt (INTLRn) is generated.

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 13-40**). 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 **13.5.8 (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 **13.5.8 (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 13-41**) 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 13-42**).

Caution The successful BF reception flag (UFnBSF) is not set in normal UART mode.

Figure 13-40. Timing of Judging Framing or Overrun Error in Normal UART Mode

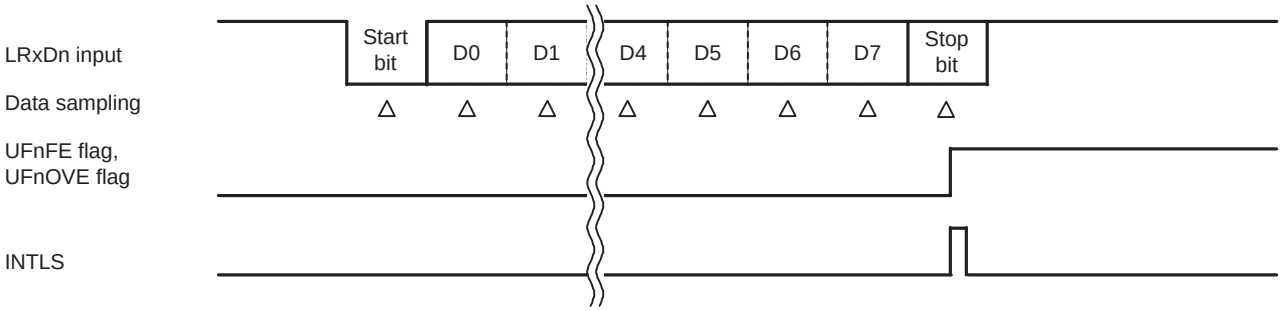


Figure 13-41. Timing of Occurrence of Data Consistency Error When BF Is Transmitted When UFnBRF = 1 (When Reception Is in Progress After Transmission of Stop Bit Has Stopped (Previous Input Data = 1))

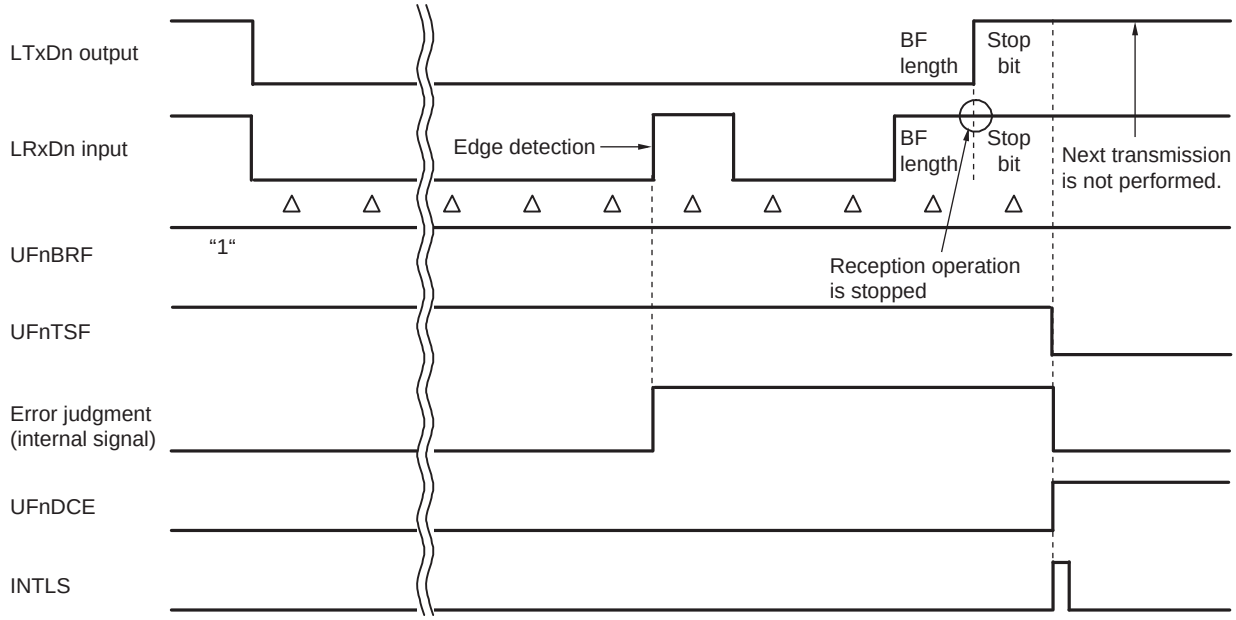
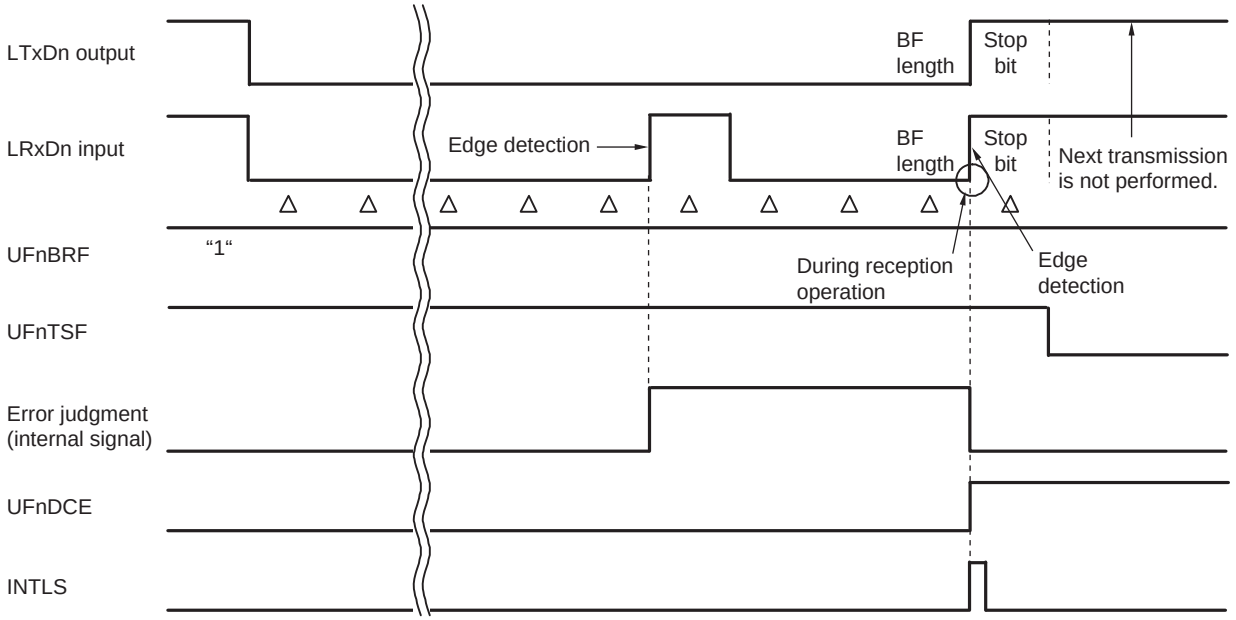


Figure 13-42. Timing of Occurrence of Data Consistency Error When BF Is Transmitted When UFnBRF = 1 (When Reception Is in Progress After Transmission of Stop Bit Has Started (Previous Input Data = 0))



(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 13-43**). 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 13-44**). Detection of framing or overrun errors is not performed. At this time, receive data is not stored into the UFnRX register.

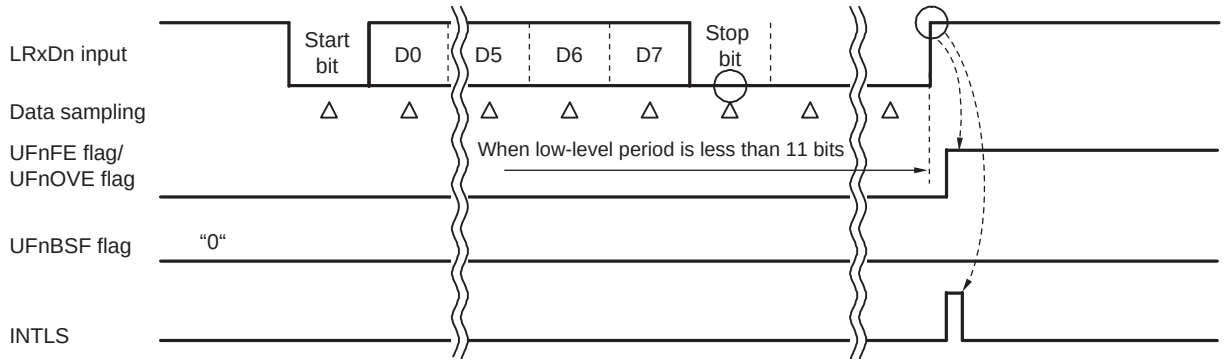
If a reception operation is stopped when starting to transmit the stop bit of data or a BF while UFnBRF is “0”, the data consistency error interrupt and flag are changed when the bit following the stop bit is started (see **13.5.8 (2)**).

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 **13.5.8 (1)** and **Figure 13-45**).

On the other hand, if input data “1” is detected during BF transmission with UFnBRF set to “1”, it is performed after transmission of the first stop bit has been completed (see **Figure 13-46**). After BF transmission has been completed, it is performed at a bit for which “1” is detected (see **Figure 13-47**).

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 13-43. Framing Error/Overrun Error Judgment Timing upon BF Reception Failure (When UFnBRF = 0)



Remark n = 0, 1

Figure 13-44. Status Interrupt Occurrence Timing upon Successful BF Reception (When UFnBRF = 0)

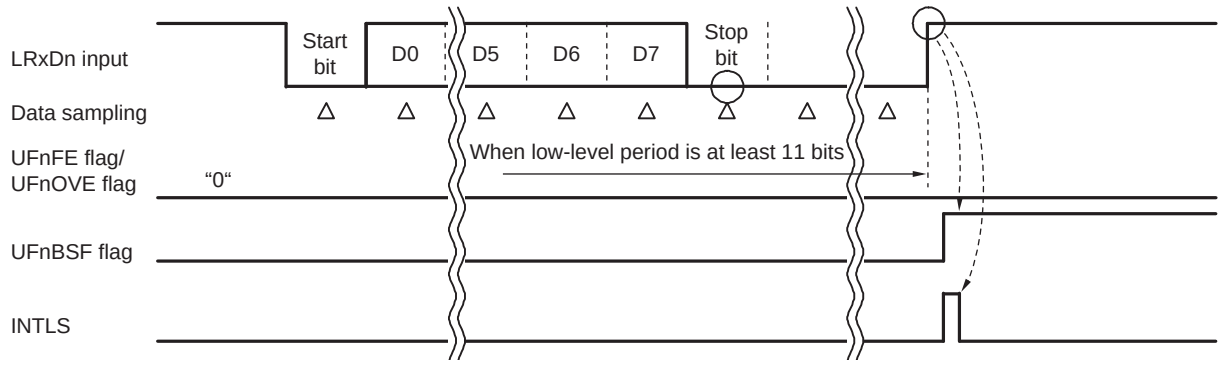
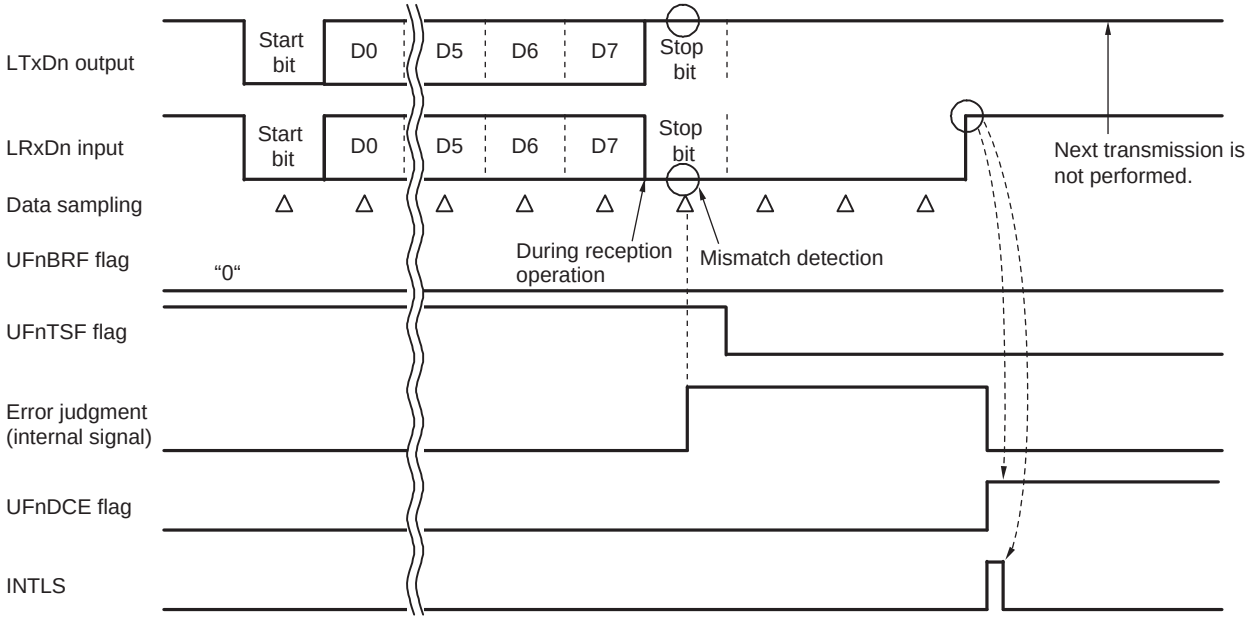


Figure 13-45. Example of Data Consistency Error Occurrence Timing When UFnBRF = 0



Remark n = 0, 1

Figure 13-46. Example of Consistency Error Occurrence Timing During BF Transmission When UFnBRF = 1 (If Reception Operation Is Stopped When Input Data "1" Is Detected After Stop Bit (Previous Bit Is "1"))

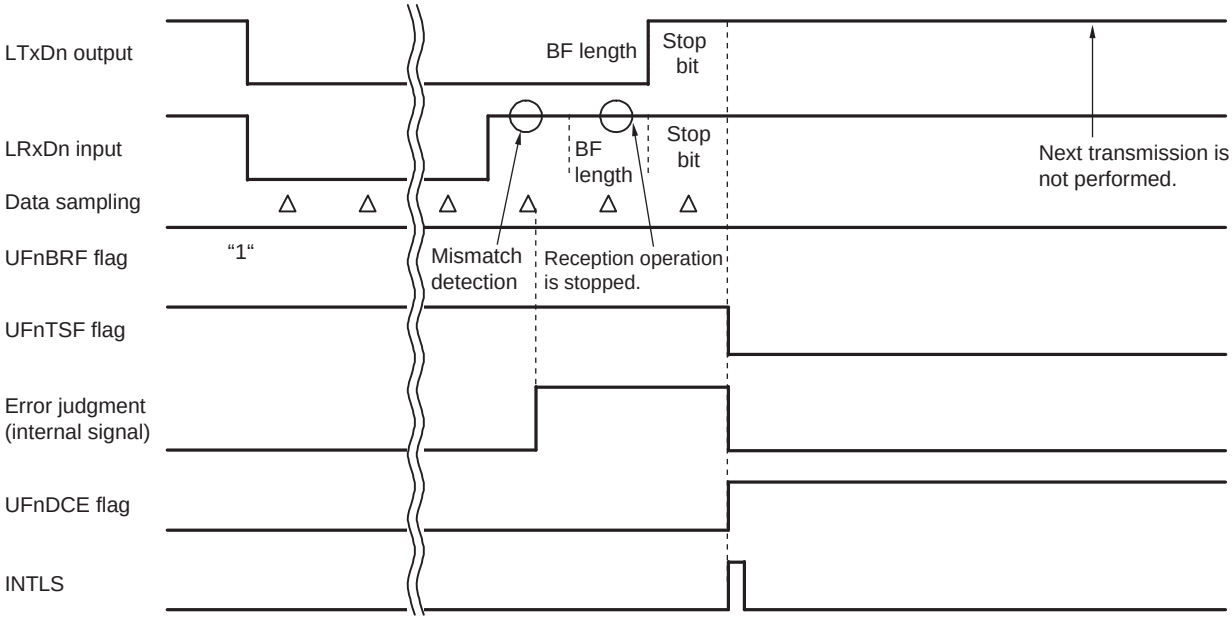
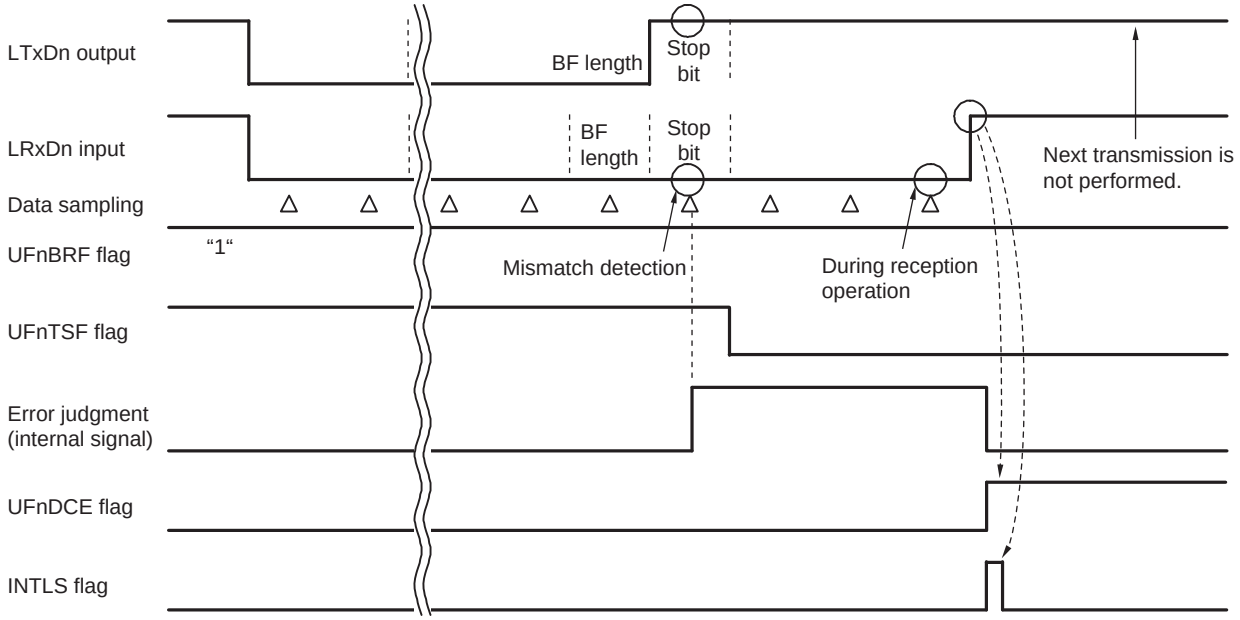


Figure 13-47. Example of Consistency Error Occurrence Timing During BF Transmission When UFnBRF = 1 (If During Reception Operation When Input Data "1" Is Detected After Stop Bit (Previous Bit Is "0"))



Remark n = 0, 1

13.5.10 LIN-UART reception status interrupt generation sources

LIN-UART reception 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, LIN-UARTn reception status interrupt (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 LIN-UARTn reception 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 LIN-UART reception status interrupt generation timing and status flag change timing differ, depending on the mode setting and generation source.

Table 13-3. LIN-UART Reception 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.

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 13-4. 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.

Remark ×: don't care

13.5.11 Transmission start wait function

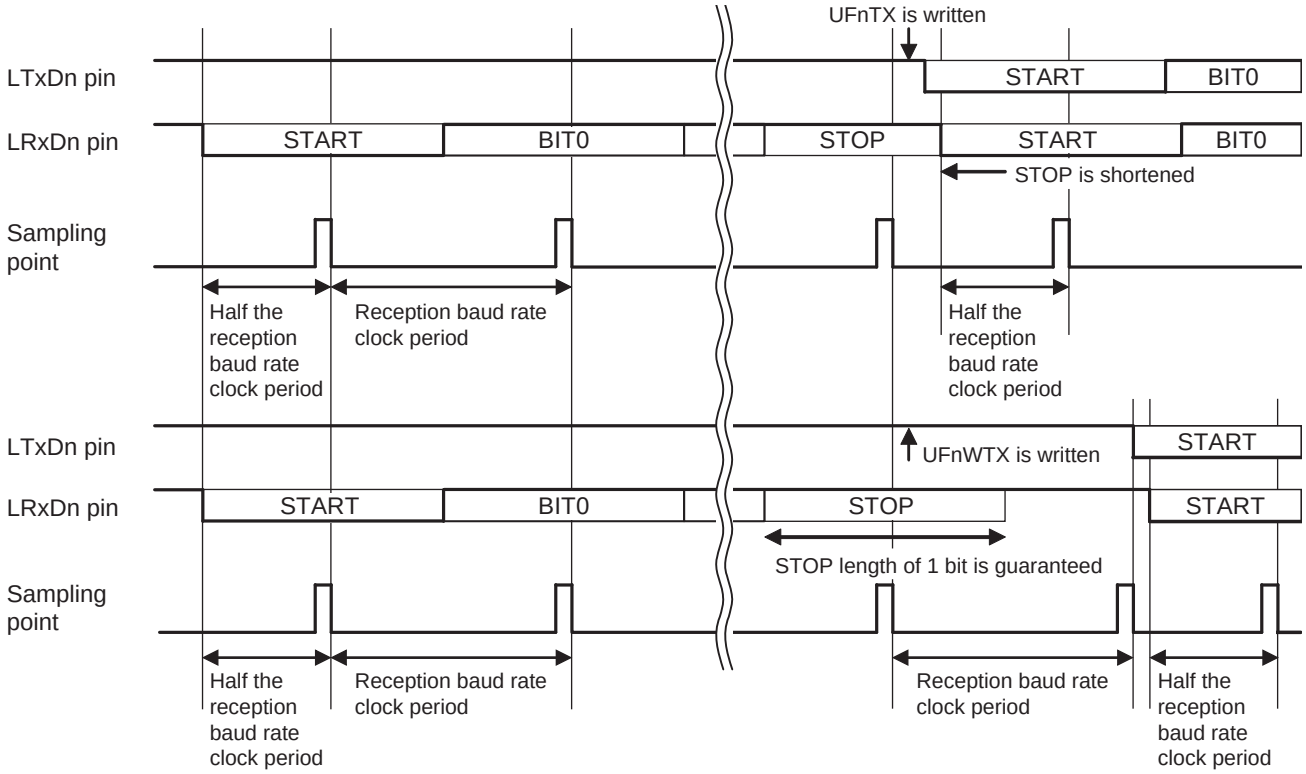
The RL78/D1A is provided with a function to guarantee the stop bit length of reception when reception is switched to transmission to perform LIN communication.

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 13-48. When Transmit Data Has Been Written During Stop Bit of Receive Data



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

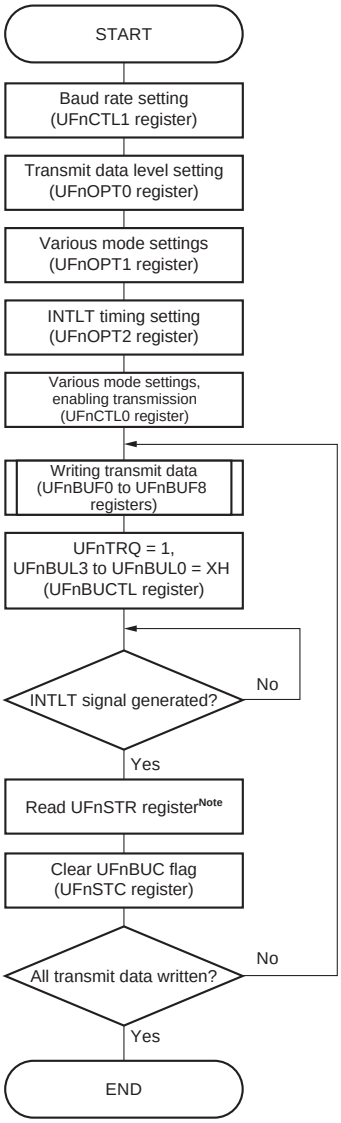
13.6 UART Buffer Mode

The RL78/D1A is provided with a 9-byte transmission buffer that can be used for normal UART communication (UFnMD1, UFnMD0 = 00B).

13.6.1 UART buffer mode transmission

The following figure shows the procedure for transmitting data in UART buffer mode.

Figure 13-49. UART Buffer Mode Transmission Processing Flow



Note This can be omitted.

Cautions 1. Set the following values when performing data transmission in UART buffer transmission mode.

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. UFnPRQ must not be set to 1 before completion of receive data reading.

Remarks 1. See **(2)** of **13.11 Cautions on Use for details of starting LIN-UART**.

2. X: don't care

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 13-50. UART Buffer Mode Transmission Example (UFnITS = 0)

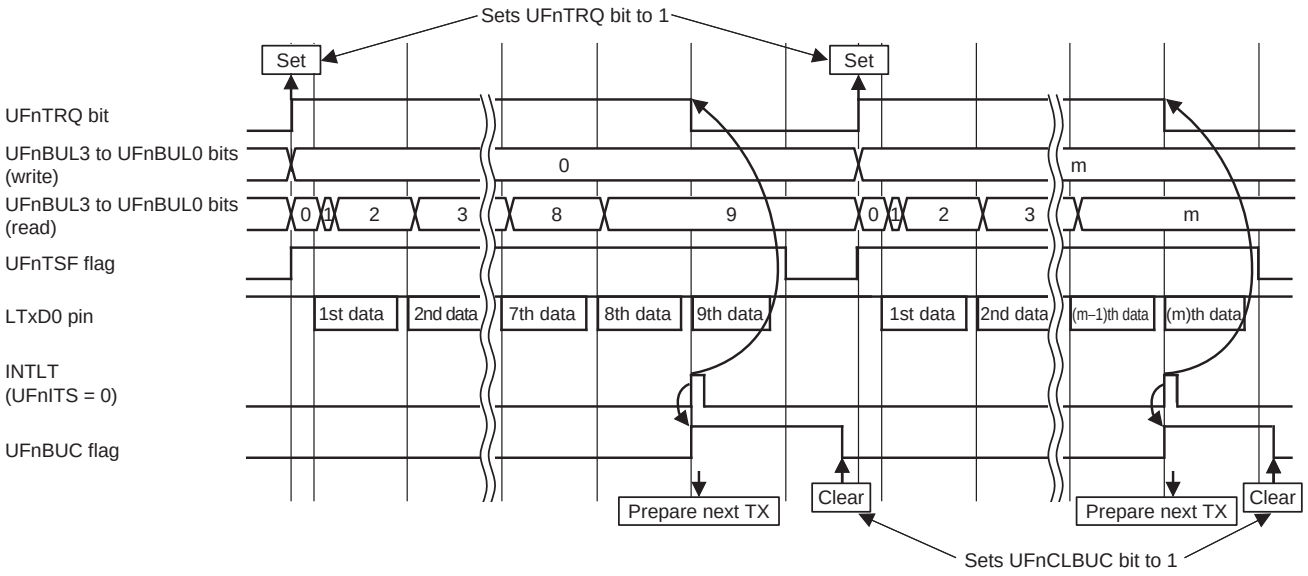
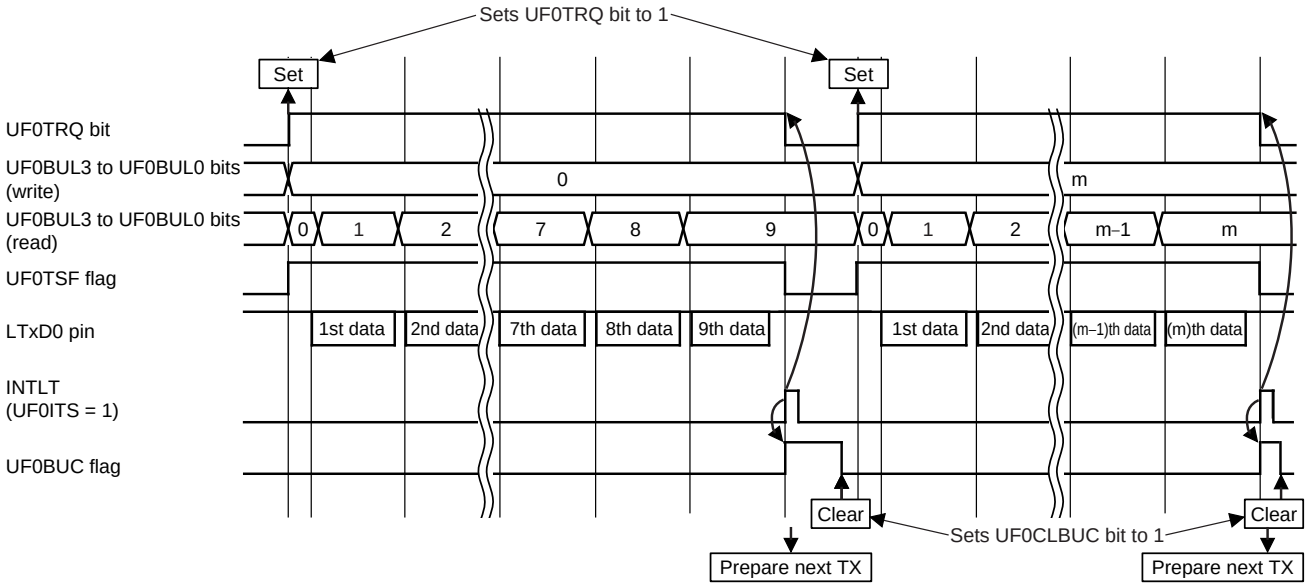


Figure 13-51. UART Buffer Mode Transmission Example (UFnITS = 1)



Remark m = 1 to 9

13.7 LIN Communication Automatic Baud Rate Mode

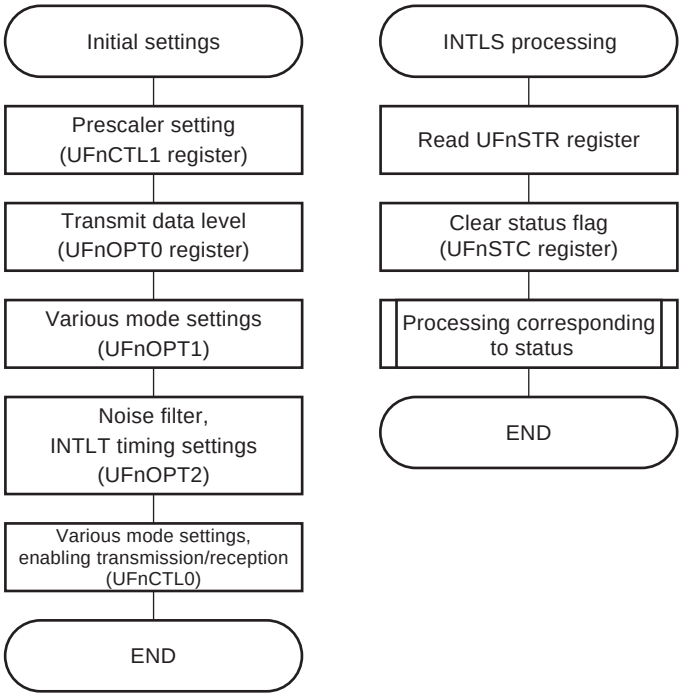
In LIN communication automatic baud rate mode, a BF and an SF are automatically detected and the baud rate is set according to the measurement result of the SF.

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 13-52. Basic Processing Flow Example of LIN Communication Automatic Baud Rate Mode (1/2)



Cautions 1. Set the following values when performing LIN communication automatic baud rate mode.

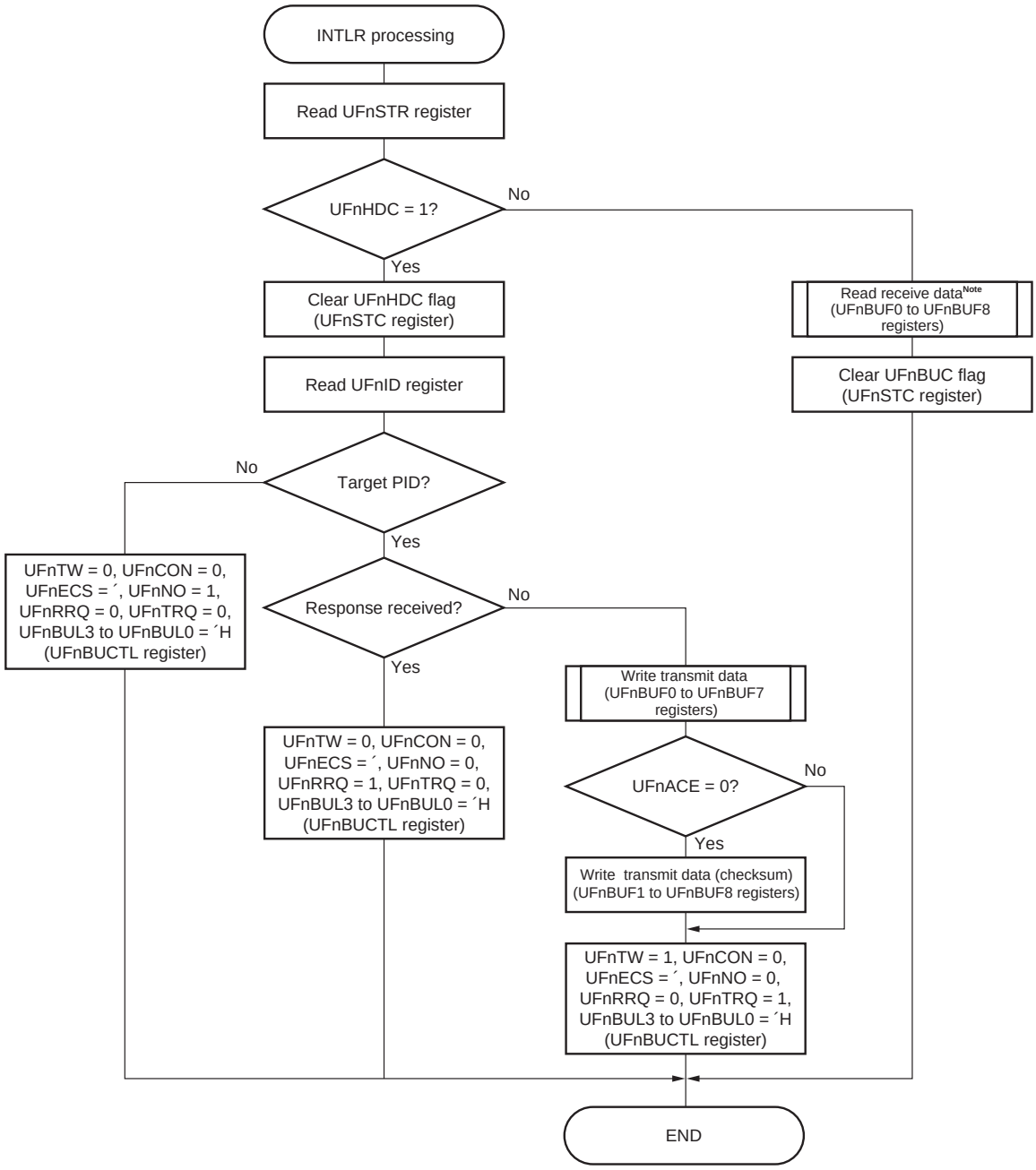
The transmit and receive data levels are normal input (UFnTDL = UFnRDL = 0).
Expansion bits are disabled (UFnEBE = 0).
Automatic baud rate mode (UFnMD1, UFnMD0 = 11B) as the mode.
Consistency check selection (UFnDCS = 1).
Transmission interrupt is transmission start (UFnITS = 0).
Communication direction 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.

3. The checksum field should be included when UFnACE = 0.

Remark See **(2)** of **13.11 Cautions on Use for details of starting LIN-UART**.

Figure 13-52. Basic Processing Flow Example of LIN Communication Automatic Baud Rate Mode (2/2)



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 before completion of receive data reading, because, when the UFnRRQ bit is set, storing (overwriting) into a buffer is performed even if reading receive data has not ended.

3. Setting (1) the UFnTW bit is prohibited, except when operation is switched to response transmission after header reception.

Remark ×: don't care

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).

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.

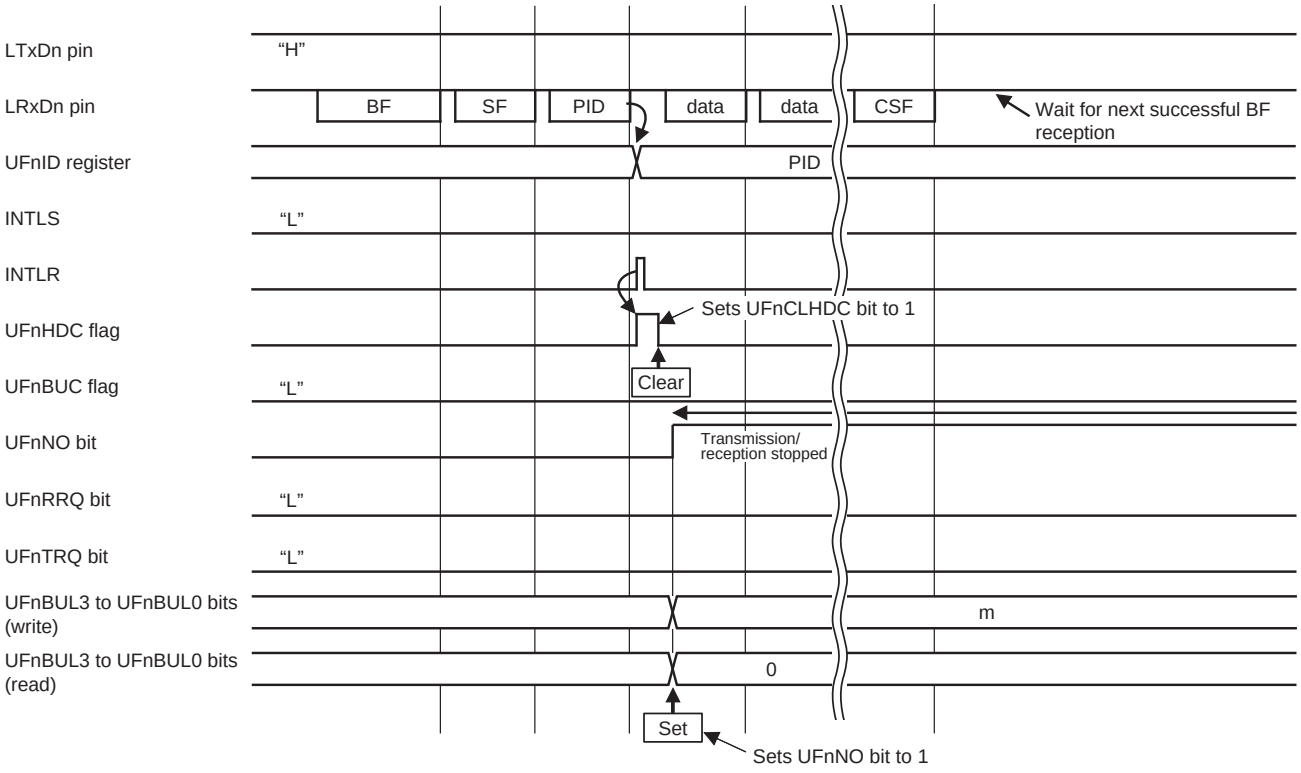
For a response transmission PID, the UFnTRQ bit is set at the same time as the response data length (UFnBUL3 to UFnBUL0), and response transmission processing is performed, after transmit data has been set to a buffer. At that time, the receive data will be stored in the UFnRX register. However, no overrun 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 **13.7.2 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.

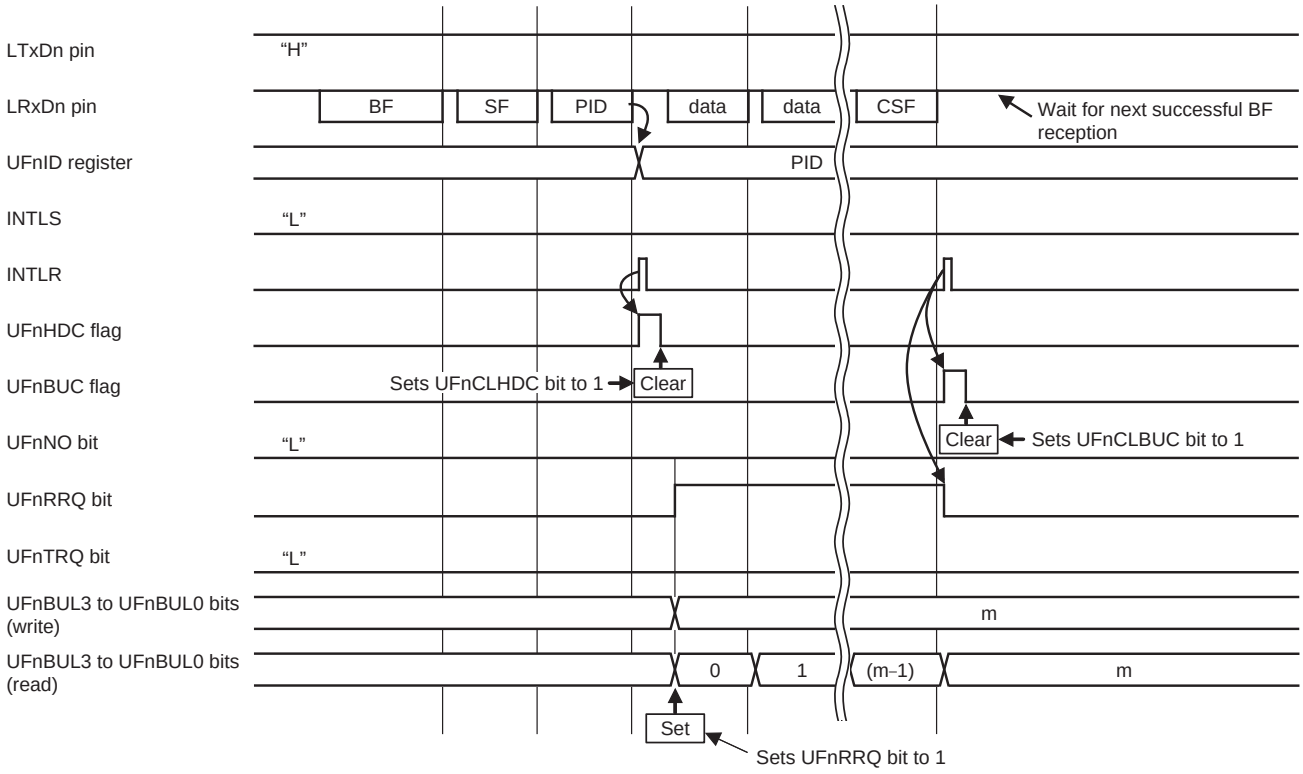
In automatic baud rate mode, no overrun error occurs, because a buffer is used (the UFnRX register is not used).

Figure 13-53. LIN Communication Automatic Baud Rate Mode (Non-Target PID)



Remark n = 0, 1, m = 1 to 9

Figure 13-54. 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 = 3) 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

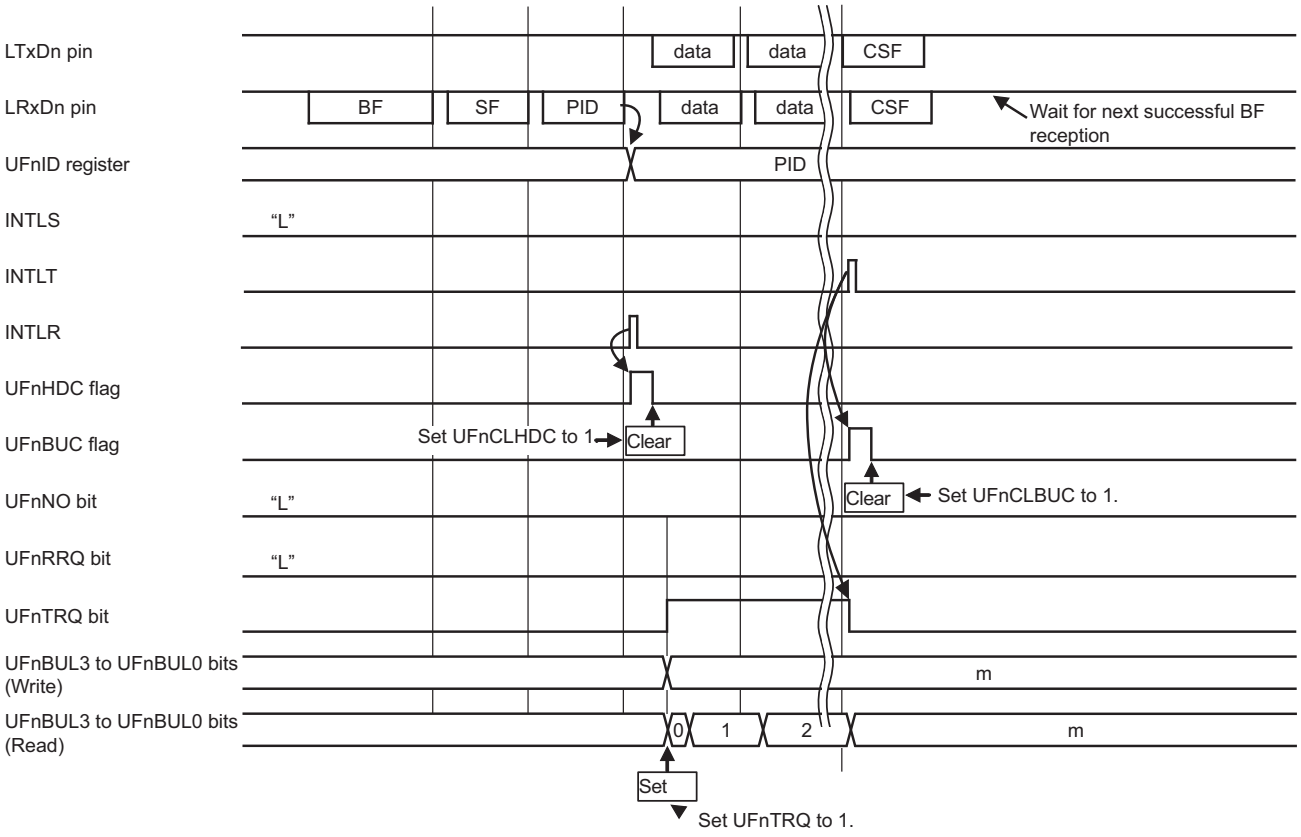
(2) When 3-byte data is received (UFnBUL3 to UFnBUL0 = 4)

	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

Figure 13-55. 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 = 3) 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

13.7.1 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. In both cases, a header reception completion flag (UFnHDC) is set. 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 13-56. Example of BF/SF Reception Failure

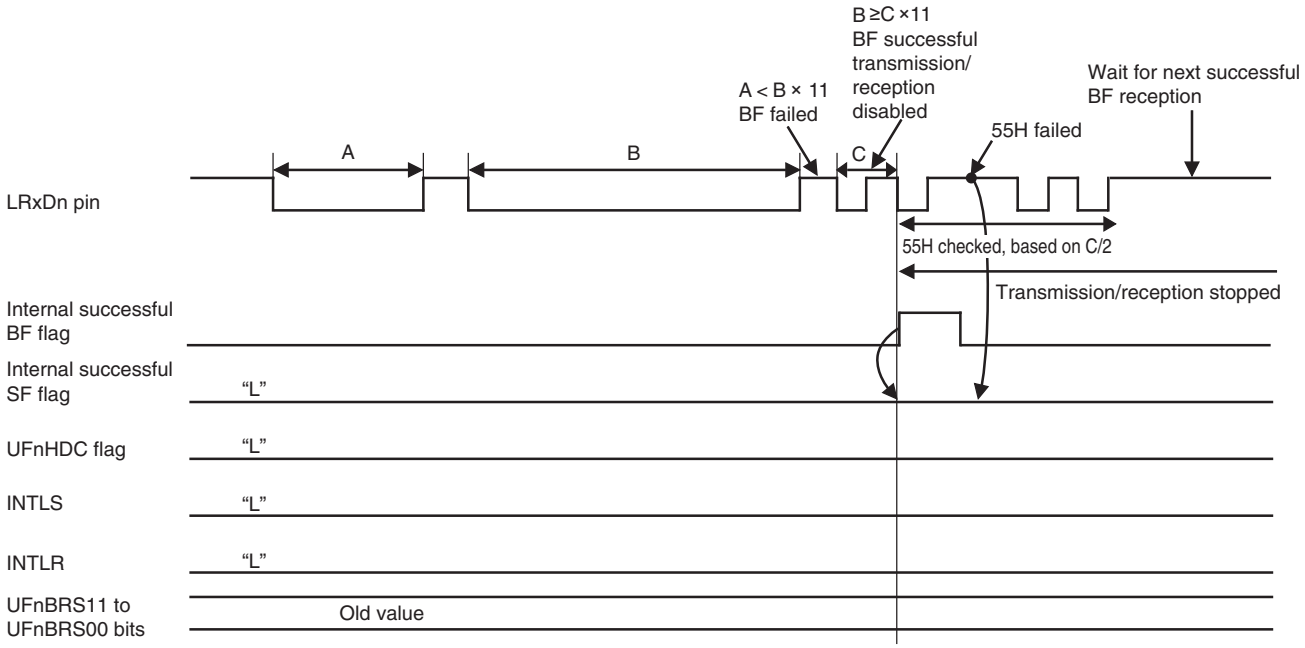
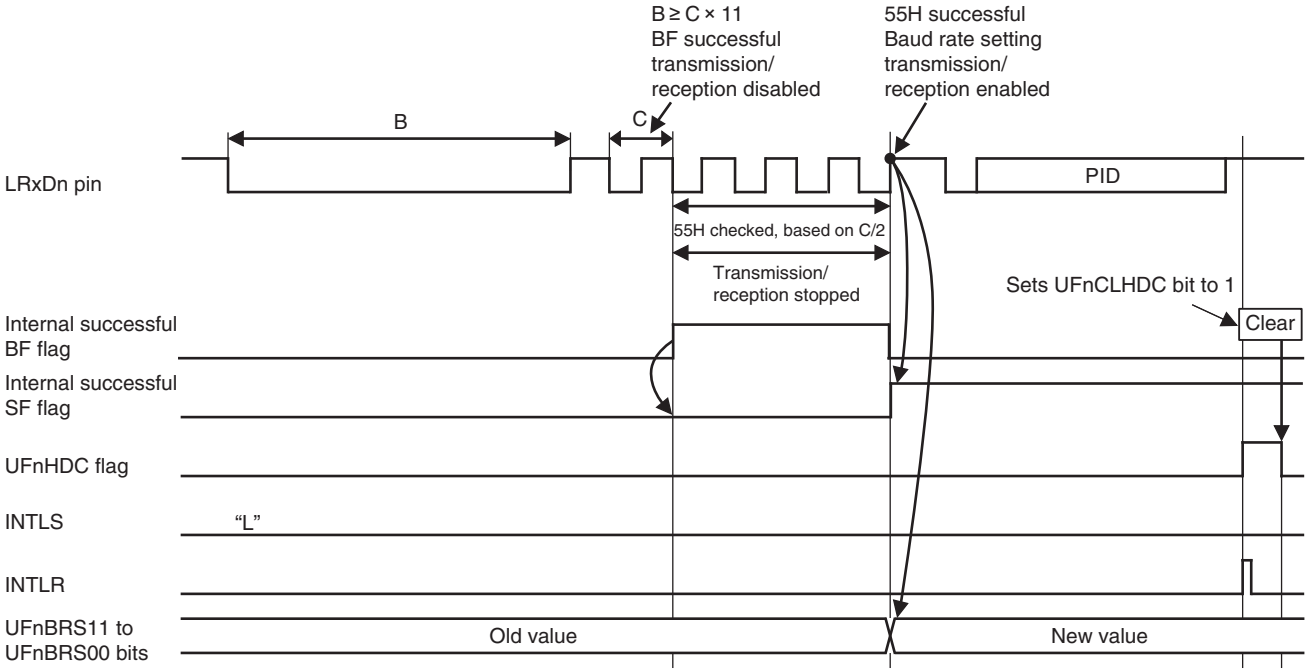
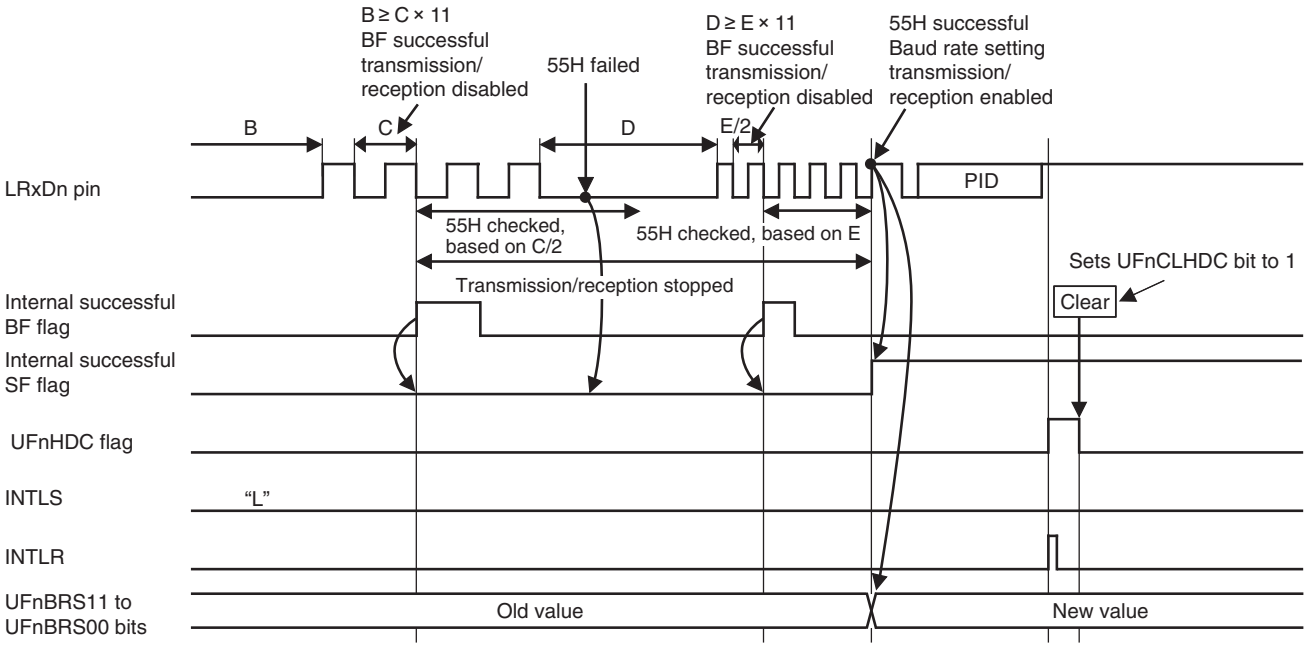


Figure 13-57. Example of Successful BF, SF, and PID Reception



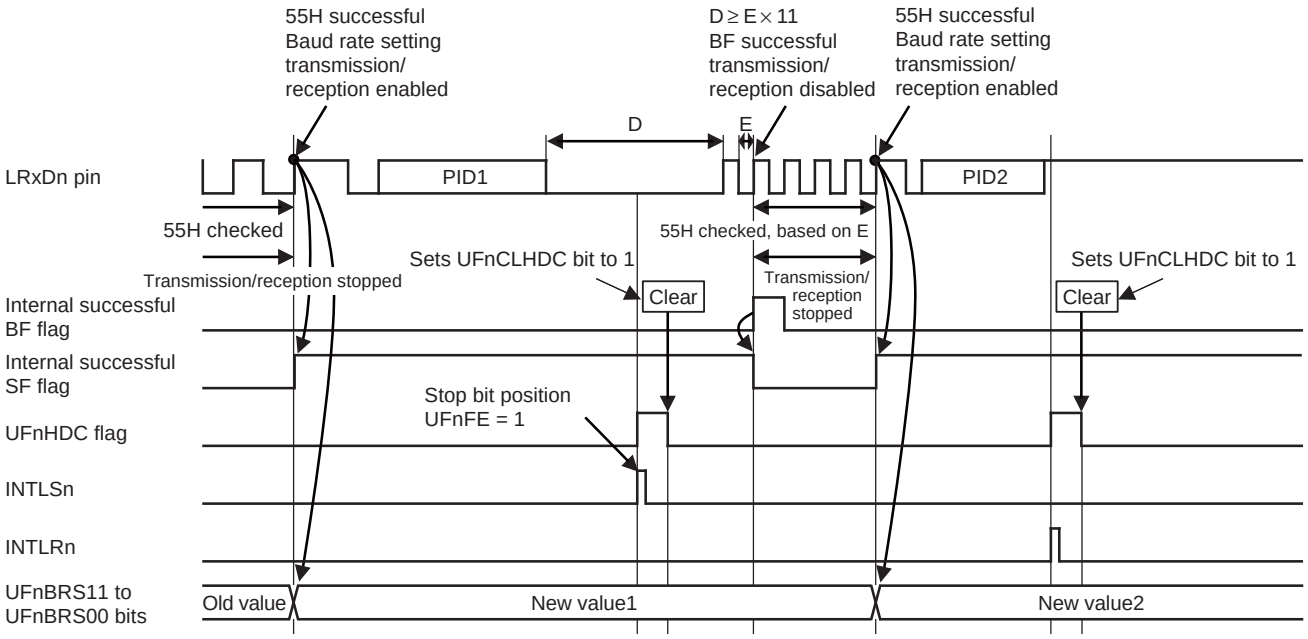
Caution When a PID reception error has occurred, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn) and other error flags (such as UFnFE and UFnIPE) change.

Figure 13-58. Example of Successful BF Reception During SF Reception (No PID Reception Error)



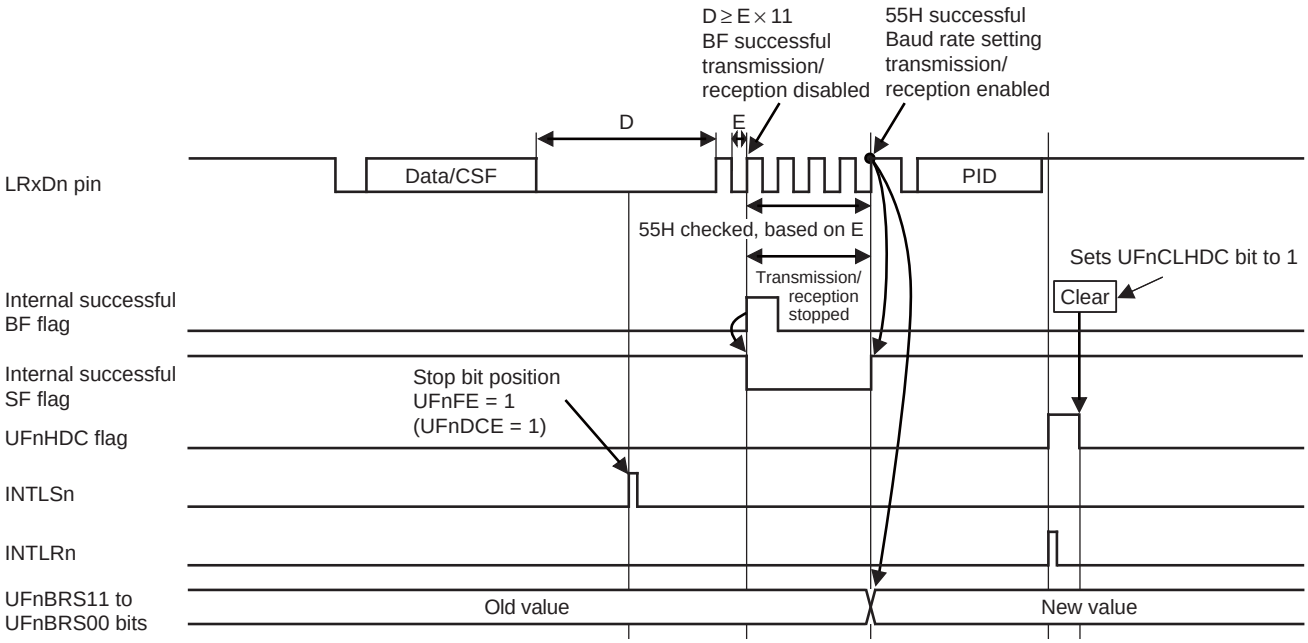
Caution When a PID reception error has occurred, a status interrupt request signal (INTLSn) is generated instead of a reception complete interrupt request signal (INTLRn) and other error flags (such as UFnFE and UFnIPE) change.

Figure 13-59. 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 13-60. Example of Successful BF Reception During Data/CSF Reception (No PID Reception Error)



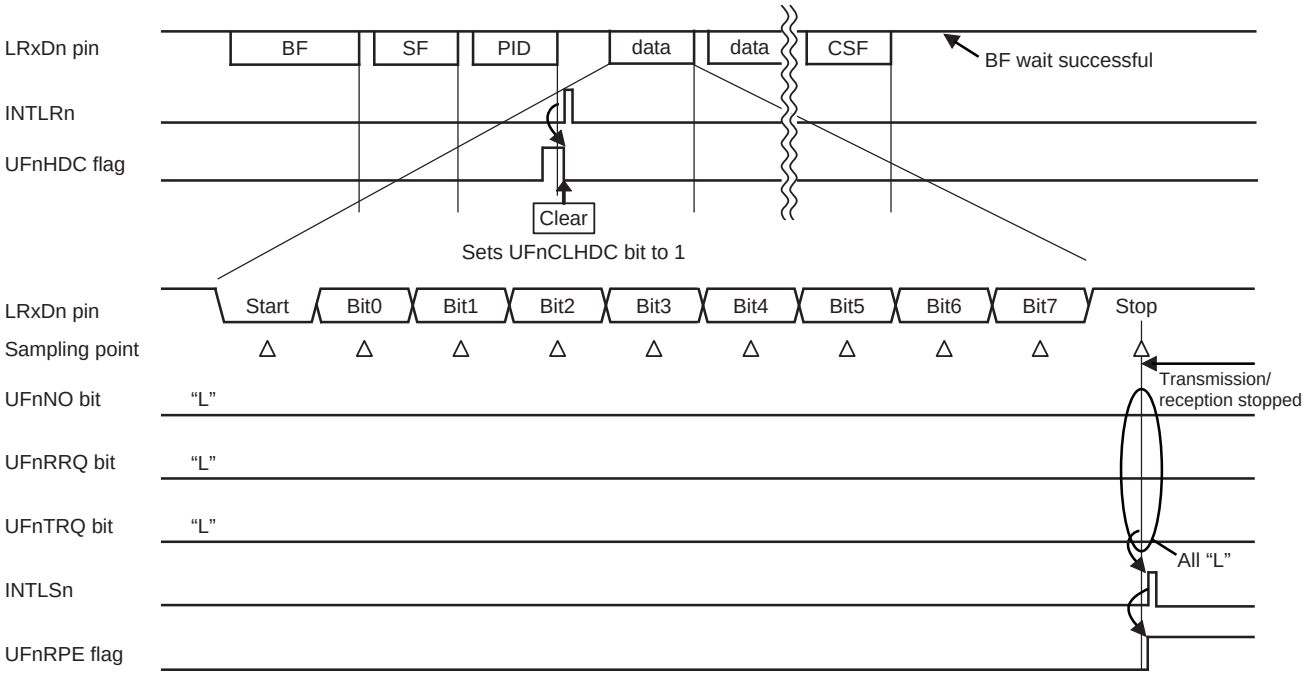
Caution If the Data/CSF stop bit position comes after the point where the internal successful BF flag has been set, the UFnBUC flag and error flags (such as UFnFE, UFnDCE, UFnCSE, and UFnRPE) are not set, and INTLSn is also not generated.

13.7.2 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.

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 13-61. Response Preparation Error Occurrence Example

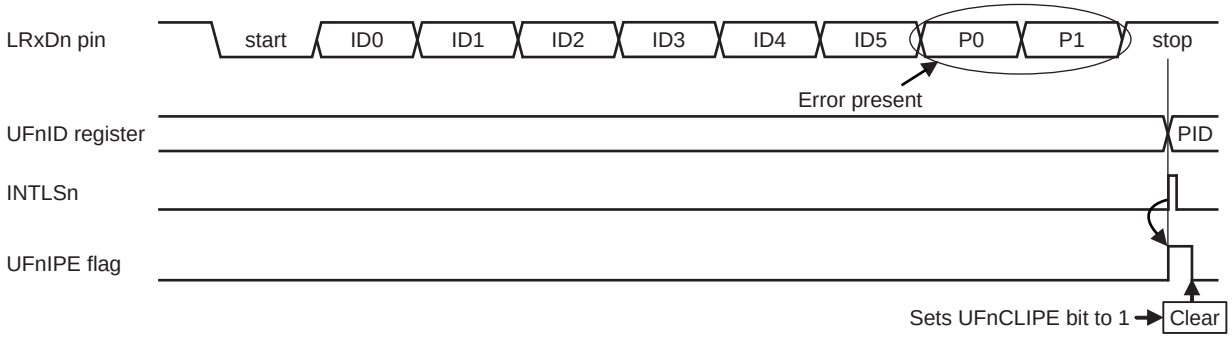


Caution If UFnCON = 0, no response preparation error will occur, because a BF reception wait state is entered after communication of the number of bytes set using the UFnBUL3 to UFnBUL0 bits is completed.
If UFnCON = 1, a response preparation error check state is entered again after communication of 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.

13.7.3 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 13-62. PID Parity Error Occurrence Example



13.7.4 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).

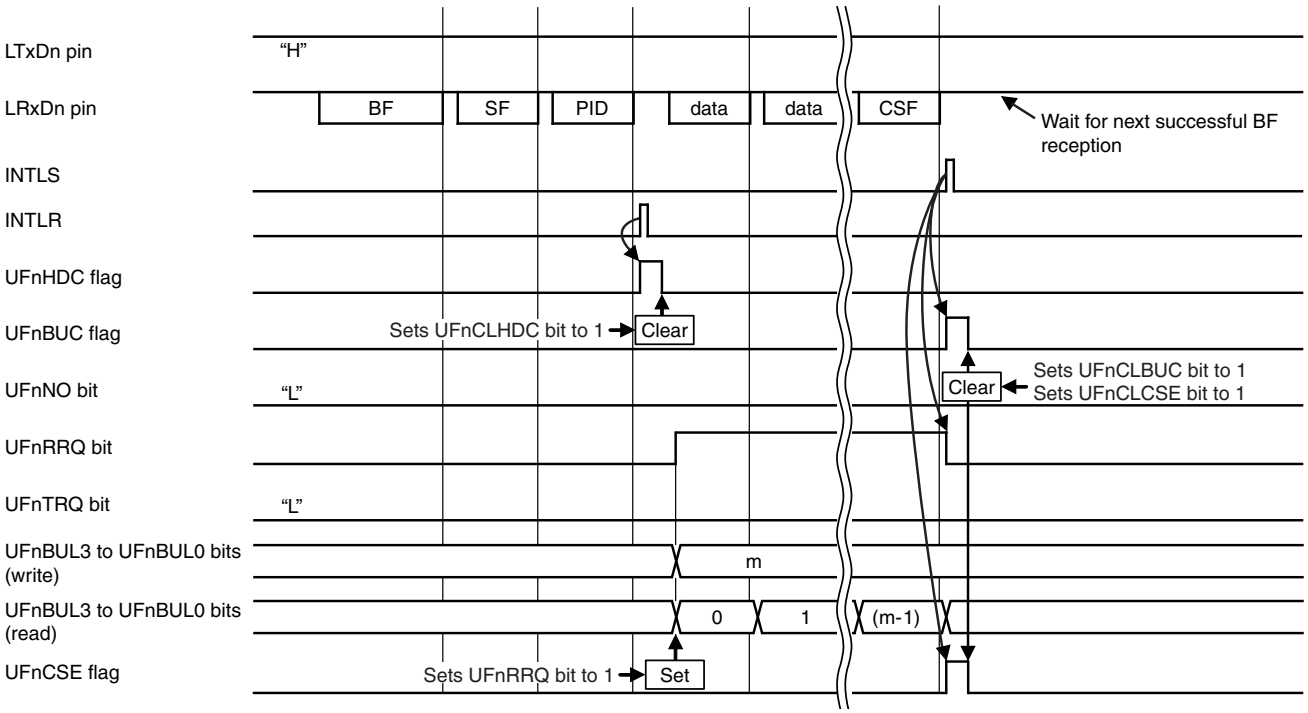
During response transmission, calculation is performed when data is transferred in 1-byte units from a buffer register to a transmit shift register[Note], and the calculation result is automatically added to the end of response transmission and transmitted. A checksum is not required to be set to a buffer 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 When the enhanced checksum is selected, the value of the UFnID register is set to its initial value for calculation at the time transfer starts.

Remark n = 0, 1

Figure 13-63. Automatic Checksum Error Occurrence Example (Response Reception)



Remark n = 0, 1, m = 1 to 9

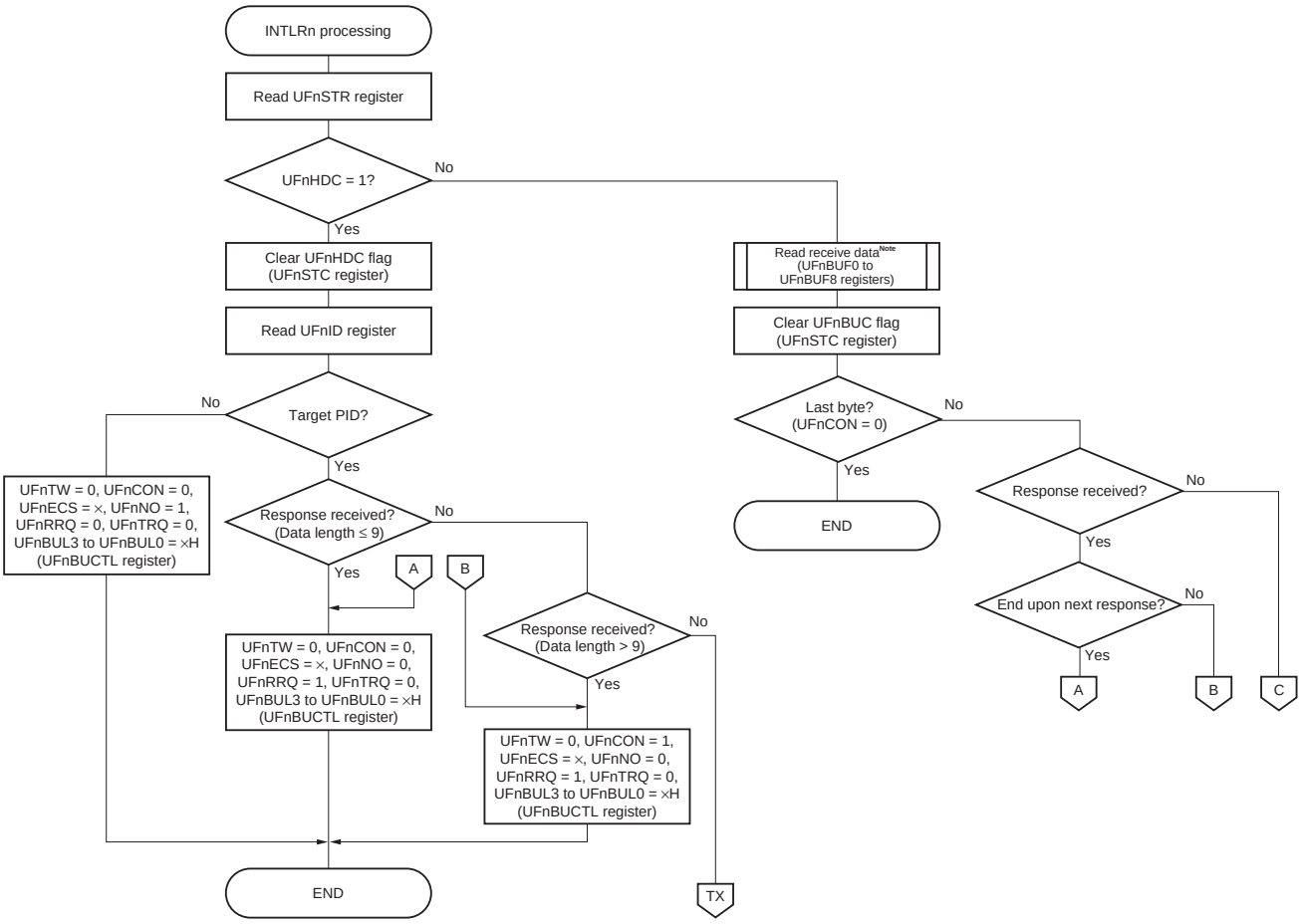
13.7.5 Multi-byte response transmission/reception function

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 **13.7 LIN Communication Automatic Baud Rate Mode**.

The response preparation error detection function, ID parity check function, and automatic checksum function are valid.

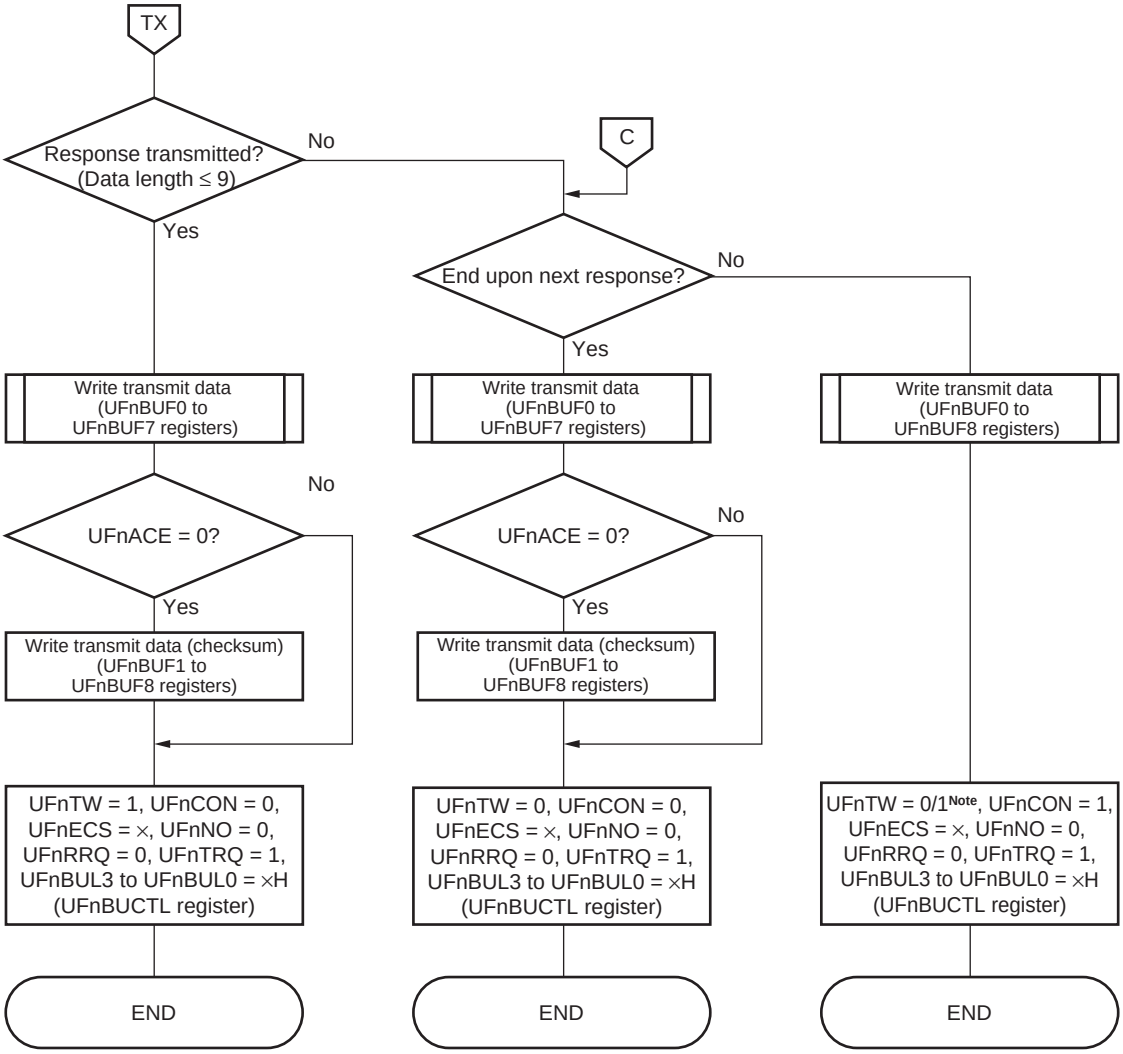
Figure 13-64. Multi-Byte Transmission/Reception Processing Flow Example (1/2)



Note This can be omitted.

Remark ×: don't care, n = 0, 1

Figure 13-64. Multi-Byte Transmission/Reception Processing Flow Example (2/2)



Note Set UFnTW to 1 during only the first data transmission after PID reception.

Cautions 1. When the buffer length bits (UFnBUL3 to UFnBULn) 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.

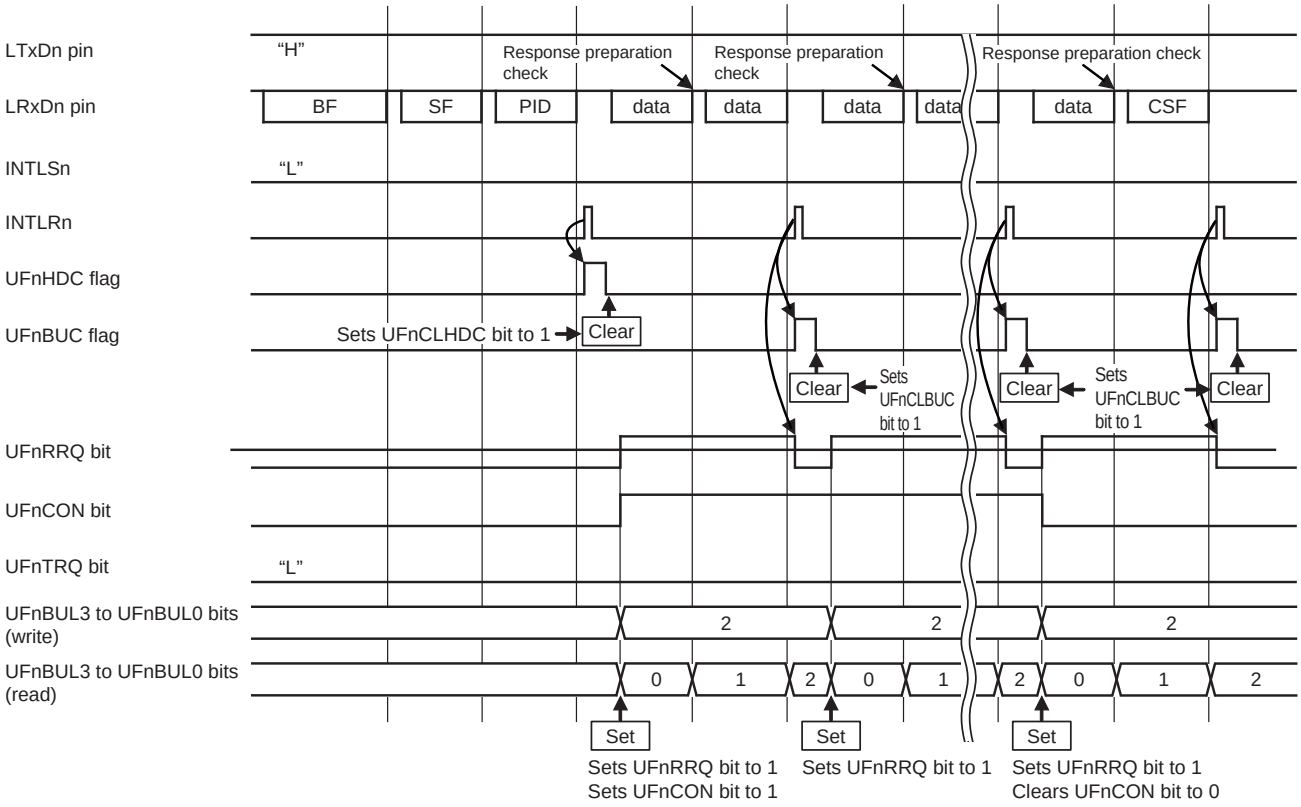
Example: When UFnBUL3 to UFnBUL0 are set to "1", data is always stored only into the UFnBUF0 register.

2. Do not set the UFnRRQ bit before completion of receive data acquisition.

3. Setting the UFnTW bit is prohibited, except when operation is switched to response transmission after header reception.

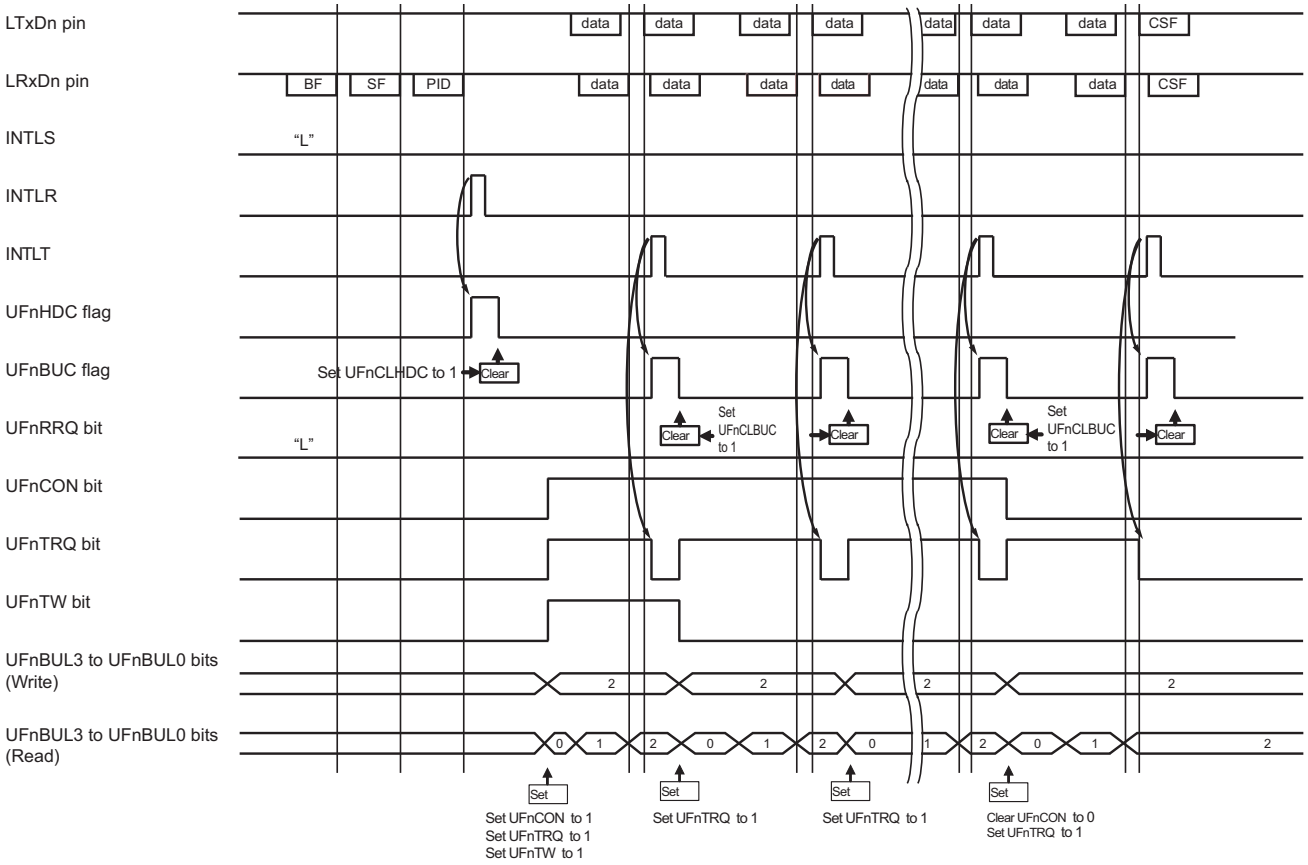
Remark ×: don't care, n = 0, 1

Figure 13-65. 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".

Figure 13-66. Multi-Byte Transmission Implementation Example



Caution When UFnBUL3 to UFnBUL0 are "2", data of the UFnBUF0 and UFnBUF1 bits are always transmitted and stored.

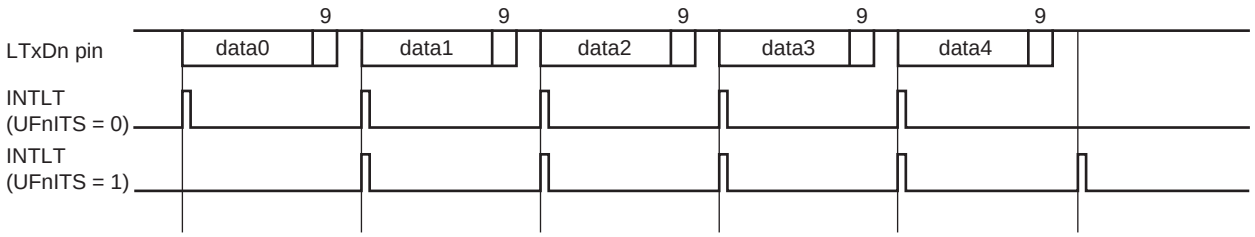
13.8 Expansion Bit Mode

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 **13.5.1 Data format** for the communication data format.

13.8.1 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 13-67. Expansion Bit Mode Transmission Example (LSB First)

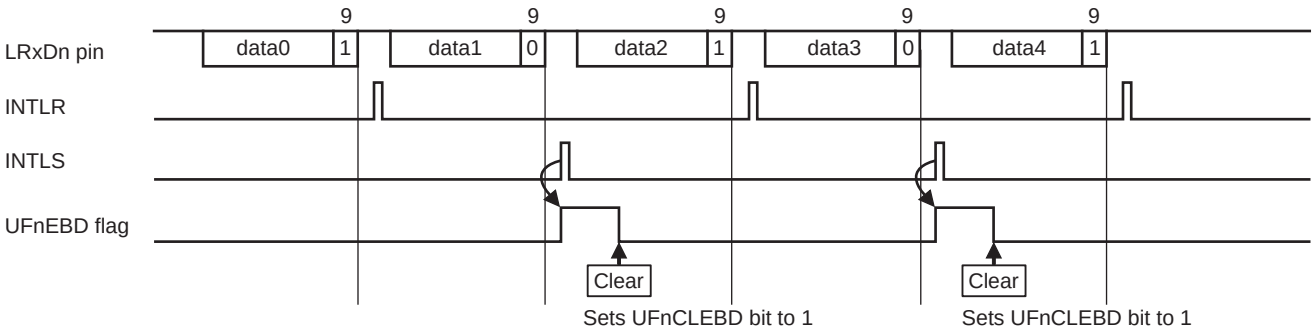


Remark n = 0, 1

13.8.2 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 13-68. 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

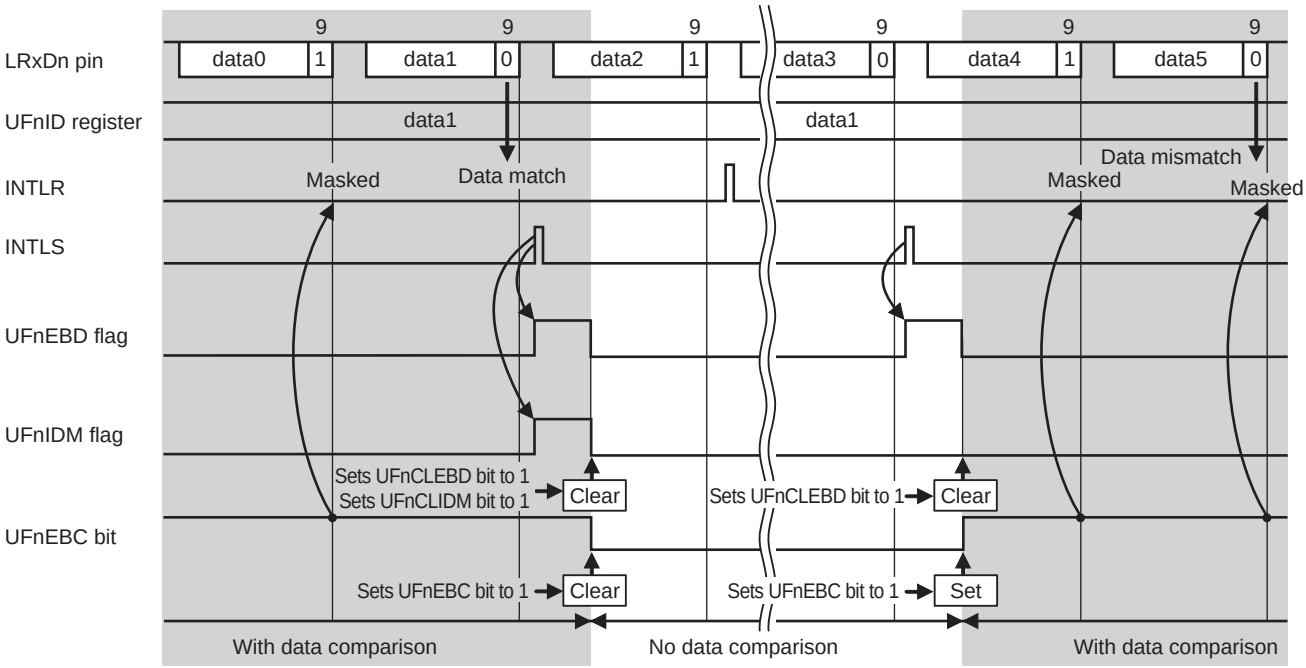
13.8.3 Expansion bit mode reception (with data comparison)

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 13-69. 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

13.9 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 13-70**). See **13.10 (1) (a) Prescaler clock (f_{UCLK})** regarding the base clock.

Figure 13-70. Noise Filter Circuit Example

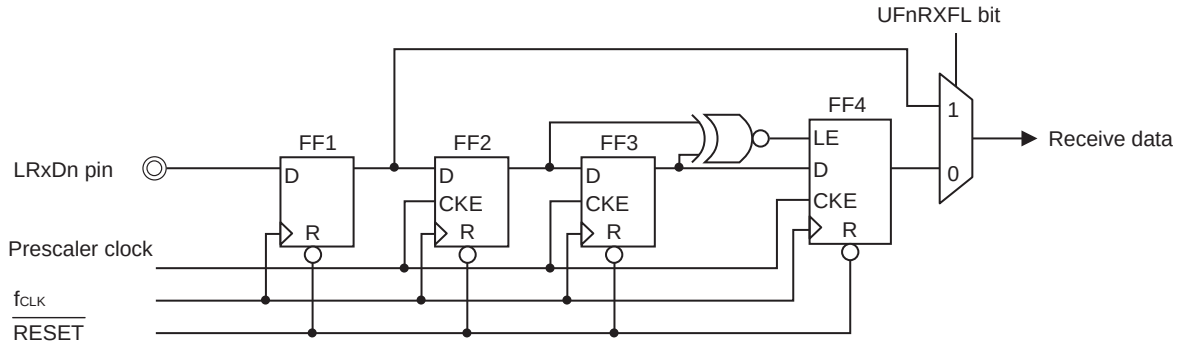
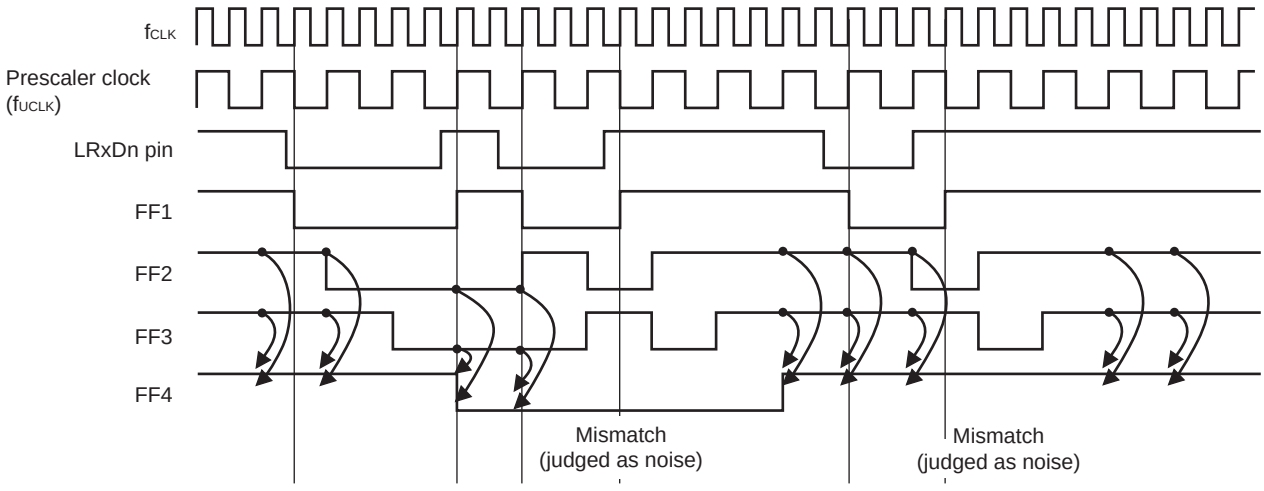


Figure 13-71. Noise Filter Timing Chart Example (UFnPRS = 1)



Remark n = 0, 1

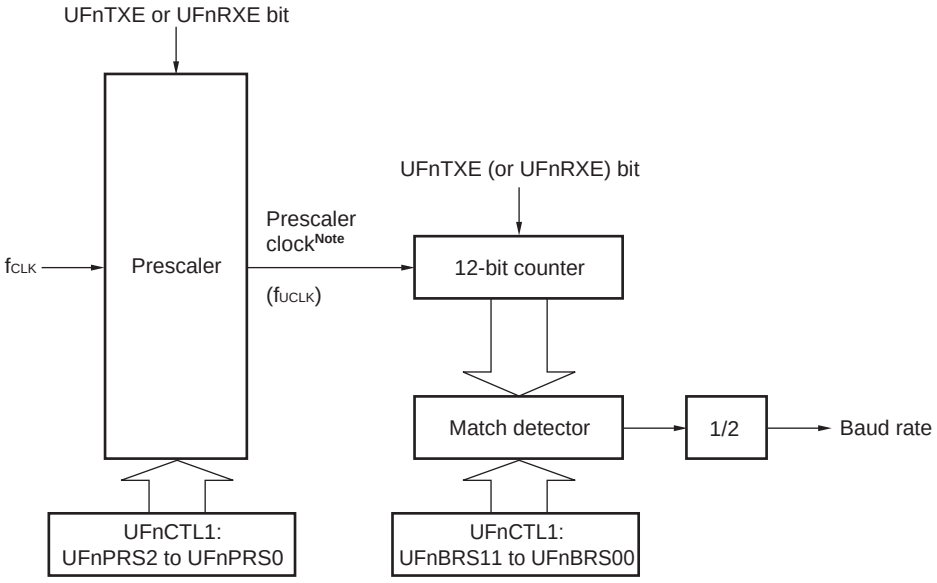
13.10 Dedicated Baud Rate Generator

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 13-72. Configuration of Baud Rate Generator



Note Clock that divides f_{CLK} by 1, 2, 4, 8, 16, 32, 64, or 128

In automatic baud rate mode, confirm that the receive pin is high before setting the UFnRXE bit to 1.

(a) Prescaler clock (f_{UCLK})

When the UFnEN 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 13-73. Format of LIN-UARTn Control Register 1 (UFnCTL1)

Address: F0242H, F0243H (UF0CTL1), F0262H, F0263H (UF1CTL1) After reset: 0FFFH R/W

	15	14	13	12	11	10	9	8
UFnCTL1	UFnPRS2	UFnPRS1	UFnPRS0	0	UFnBRS11	UFnBRS10	UFnBRS9	UFnBRS8
	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)

UFnBRS11	UFnBRS10	UFnBRS09	UFnBRS08	UFnBRS07	UFnBRS06	UFnBRS05	UFnBRS04	UFnBRS03	UFnBRS02	UFnBRS01	UFnBRS00	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 [%]

(5) Baud rate setting example

Table 13-5. 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	–	–	–	–

Table 13-6. 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	384	–0.16	156	0.16
1200	625	0.00	384	–0.16	313	0.16	78	0.16
2400	313	–0.16	313	0.16	156	0.16	625	0.16
4800	156	0.16	156	0.16	94	0.16	313	–2.34
9600	78	0.16	94	0.16	78	–2.34	156	–2.34
19200	39	0.16	78	–2.34	156	–2.34	78	–2.34
31250	24	0.00	192	0.00	96	0.00	–	–
38400	20	–2.34	156	–2.34	78	–2.34	–	–
76800	10	–2.34	78	–2.34	–	–	–	–
128000	6	–2.34	–	–	–	–	–	–
153600	5	–2.34	–	–	–	–	–	–
312500	–	–	–	–	–	–	–	–
1000000	–	–	–	–	–	–	–	–

Table 13-7. 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 13-8. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 12 MHz, UFnPRS2 to UFnPRS0 = 4 to 7)

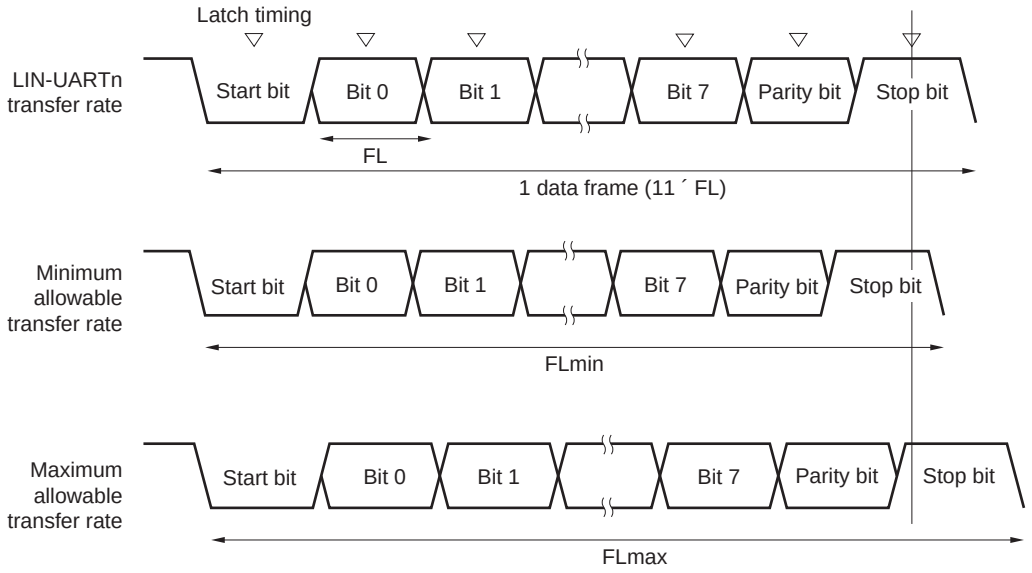
Target Baud Rate (bps)	UFnPRS2 to UFnPRS0							
	4		5		6		7	
	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)	UFnBRS11 to UFnBRS00	ERR (%)
300	1250	0.00	625	0.00	313	–0.16	156	0.16
600	625	0.00	313	–0.16	156	0.16	78	0.16
1200	313	–0.16	156	0.16	78	0.16	39	0.16
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.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 13-74. Allowable Baud Rate Range During Reception



In the figure above, the bits from the start bit to the stop bit is 11 bits long.

As shown in Figure 13-74, 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.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20\ k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

Table 13-9 shows the allowable baud rate error between LIN-UARTn and the transmission source calculated from the above-described equations for obtaining the minimum and maximum baud rate values.

Table 13-9. 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]

13.11 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 UF0TX 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 14 CAN CONTROLLER

<R>

	R5F10CGx	R5F10DGx	R5F10CLx	R5F10DLx	R5F10CMx	R5F10DMx	R5F10TPJ/ R5F10DPE/F/G	R5F10DPJ/K/L E5F10DSx
aFCAN	0 channel	1 channel	0 channel	1 channel	0 channel	1 channel	1 channel	2 channels

14.1 Outline Description

This product features an on-chip CAN (Controller Area Network) controller that complies with CAN protocol as standardized in ISO 11898.

14.1.1 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

14.1.2 Overview of functions

Table 14-1 presents an overview of the CAN controller functions.

Table 14-1. 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)

14.1.3 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.

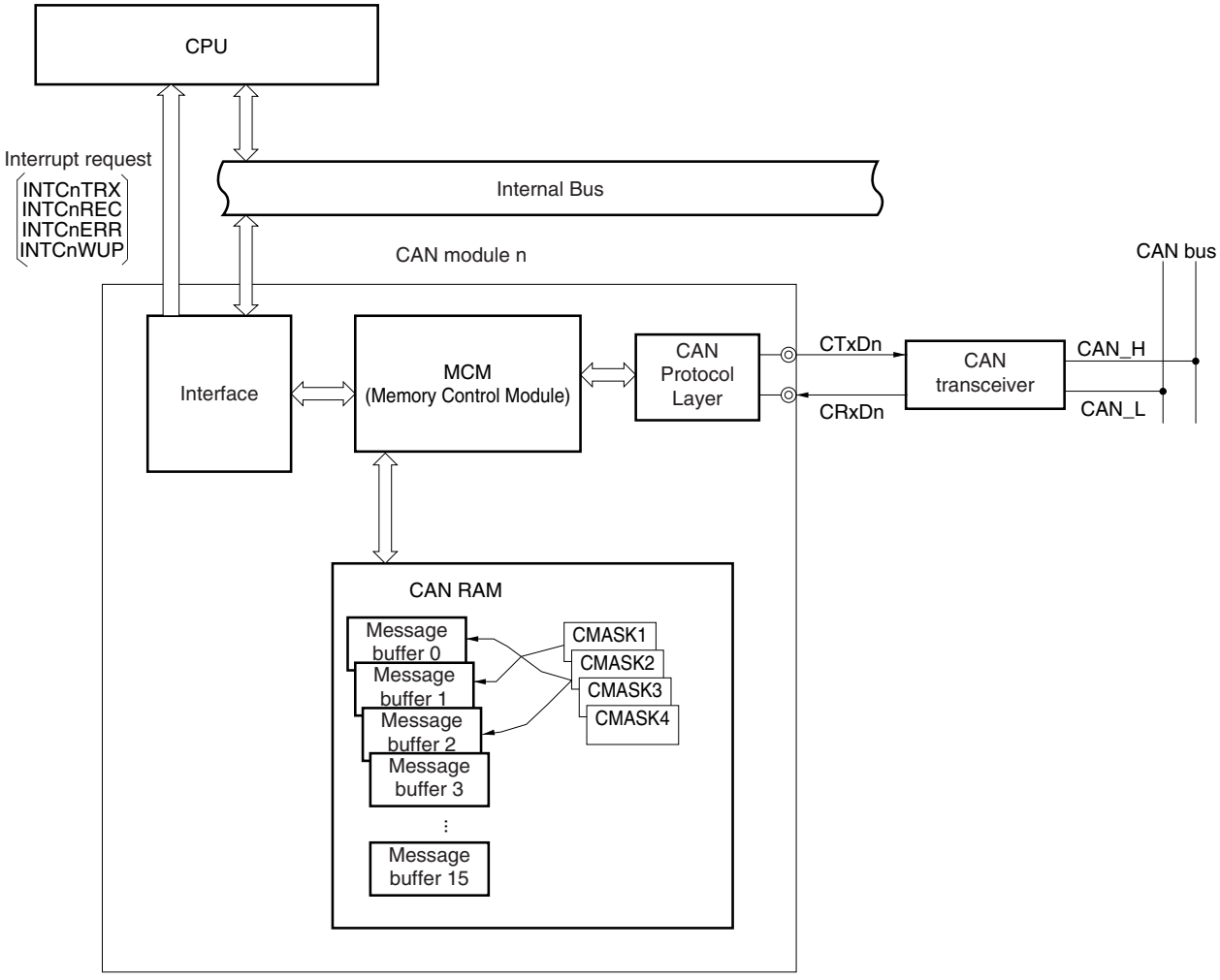
(3) CAN protocol layer

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 14-1. Block Diagram of CAN Module n



(n = 0, 1)

14.2 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 14-2. 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

14.2.1 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 x 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.

14.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 14-2. 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.

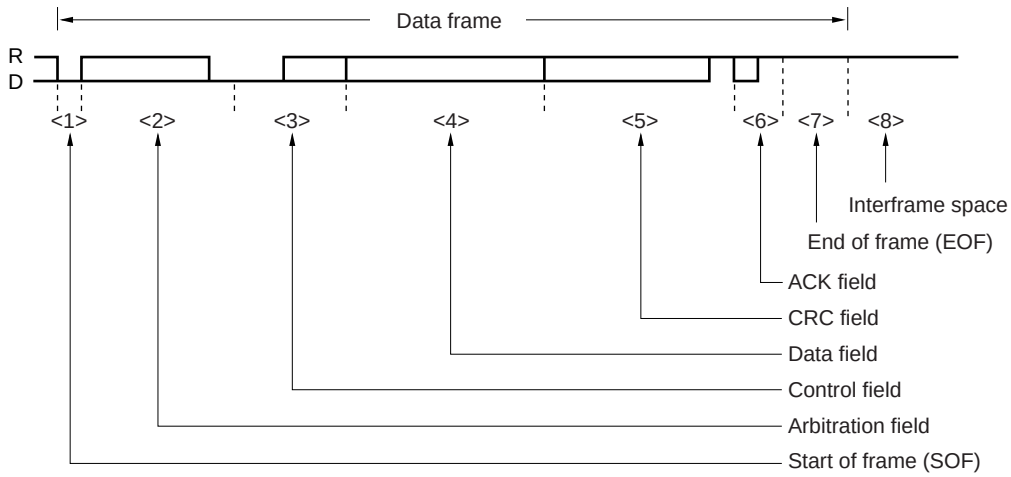
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

14.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 14-3. Data Frame

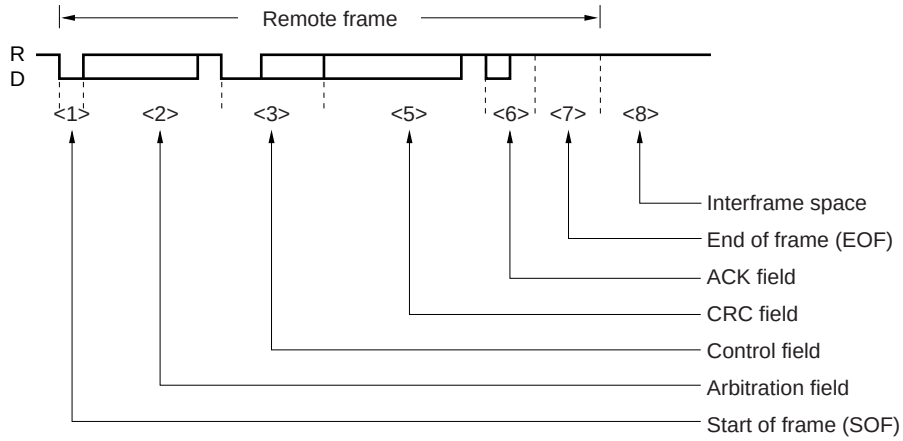


Remark D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 14-4. 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

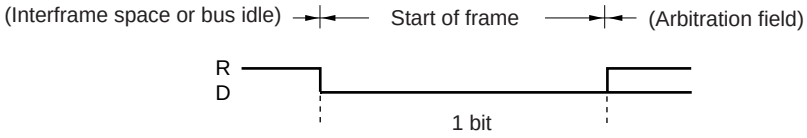
R: Recessive = 1

(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 14-5. Start of Frame (SOF)



Remark D: Dominant = 0

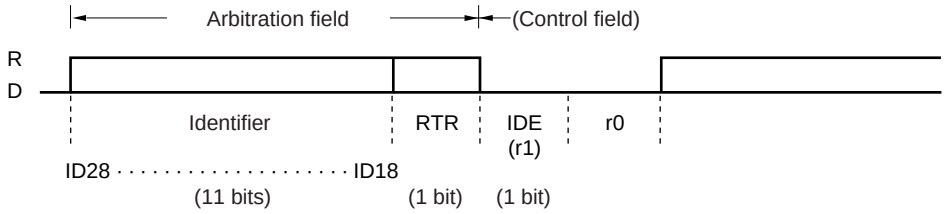
R: Recessive = 1

- 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 14-6. Arbitration Field (in Standard Format Mode)



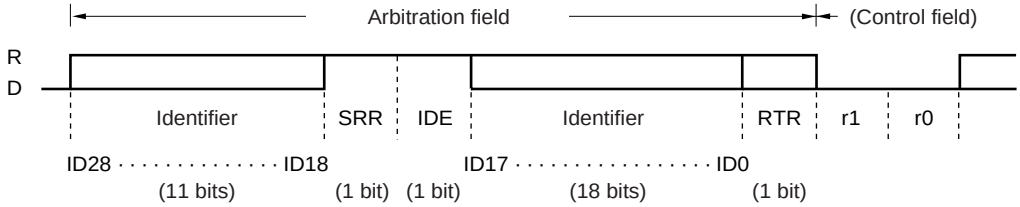
Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Figure 14-7. 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 14-3. RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

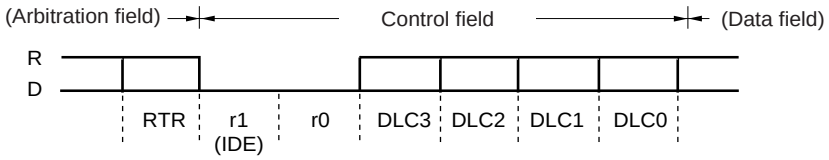
Table 14-4. Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

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 14-8. 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 14-5. Data Length Setting

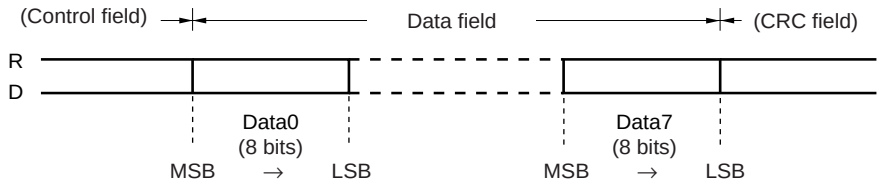
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 byte
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than the above				8 bytes regardless of the value of DLC3 to DLC0

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 14-9. Data Field

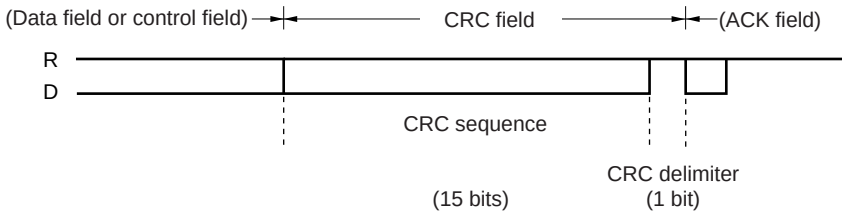


Remark D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 14-10. CRC Field



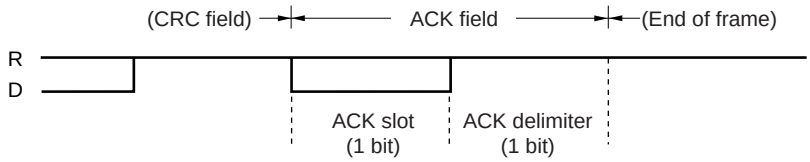
Remark D: Dominant = 0
R: Recessive = 1

- 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 14-11. ACK Field



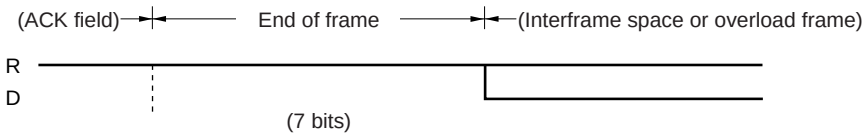
Remark D: Dominant = 0
R: Recessive = 1

- 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 14-12. End of Frame (EOF)



Remark D: Dominant = 0
R: Recessive = 1

<8> Interframe space

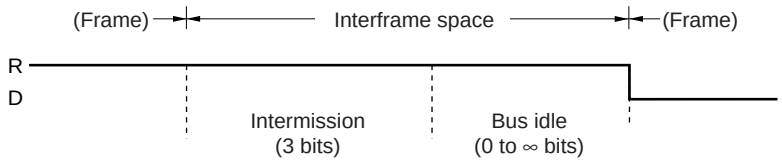
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- 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 14-13. Interframe Space (Error Active Node)



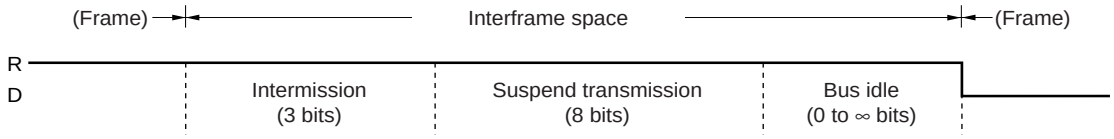
Remarks 1. Bus idle: State in which the bus is not used by any node.

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 14-14. 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

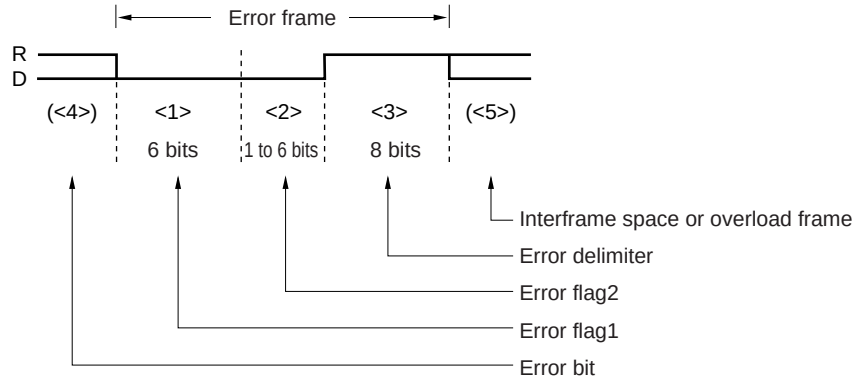
Table 14-6. Operation in Error Status

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.

14.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 14-15. Error Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 14-7. 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

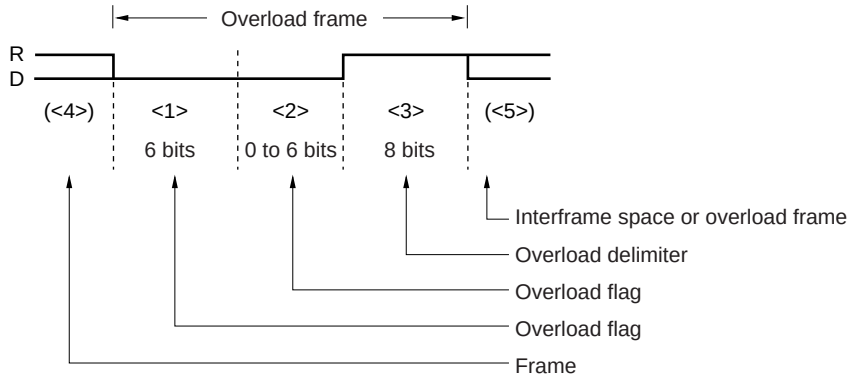
14.2.5 Overload frame

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 14-16. Overload Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 14-8. 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.

14.3 Functions

14.3.1 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 14-9. 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.

14.3.2 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 14-10. 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.

14.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

14.3.4 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.

14.3.5 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).

14.3.6 Error control function

(1) Error types

Table 14-11. 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 14-12. 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 (C0ERC, C1ERC) as shown in Table 14-13.
The present error state is indicated by the CAN module information register (C0INFO, C1INFO).
When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the C0INFO, C1INFO 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 C0INFO, C1INFO 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 C0INFO, C1INFO 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 14-13. Types of Error States

Type	Operation	Value of Error Counter	Indication of C0INFO, C1INFO Register	Operation specific to Given Error State
Error active	Transmission	0-95	TECS1, TECS0 = 00	- Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0-95	RECS1, RECS0 = 00	
	Transmission	96-127	TECS1, TECS0 = 01	
	Reception	96-127	RECS1, RECS0 = 01	
Error passive	Transmission	128-255	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 14-14. 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
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) - Value of 119 to 127 is set (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 (refer to timing <1> in **Figure 14-17**). This request will be immediately acknowledged, and the OPMODE bits of the C0CTRL, C1CTRL 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.

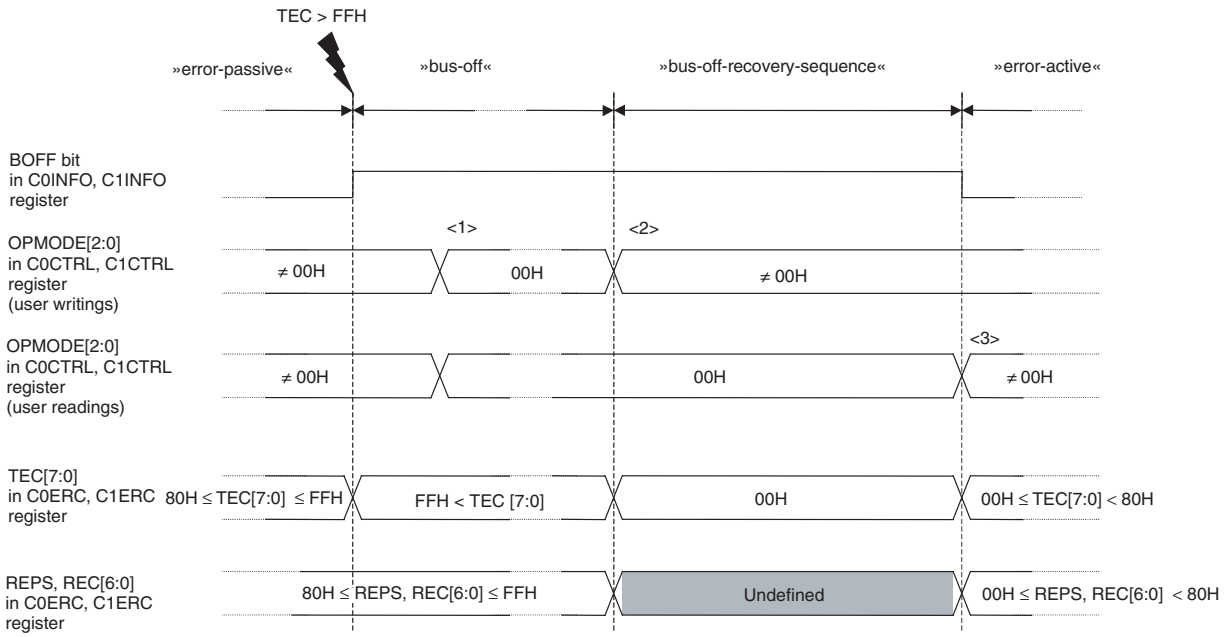
Next, the user requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in **Figure 14-17**). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the 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 (refer to timing <3> in **Figure 14-17**), 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 C0CTRL, C1CTRL register.

During the bus-off period and bus-off recovery sequence, the BOFF bit of the C0INFO, C1INFO 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 14-17. Recovery Operation from Bus-off State through Normal Recovery Sequence



(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, refer to **(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 C0CTRL, C1CTRL 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, refer to the processing in **Figure 14-82**.

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 (C0ERC, C1ERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (C0ERC, C1ERC) and CAN module information register (C0INFO, C1INFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the C0CTRL, C1CTRL 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 C0ERC, C1ERC and C0INFO, C1INFO registers are not initialized.

2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

14.3.7 Baud rate control function

(1) Prescaler

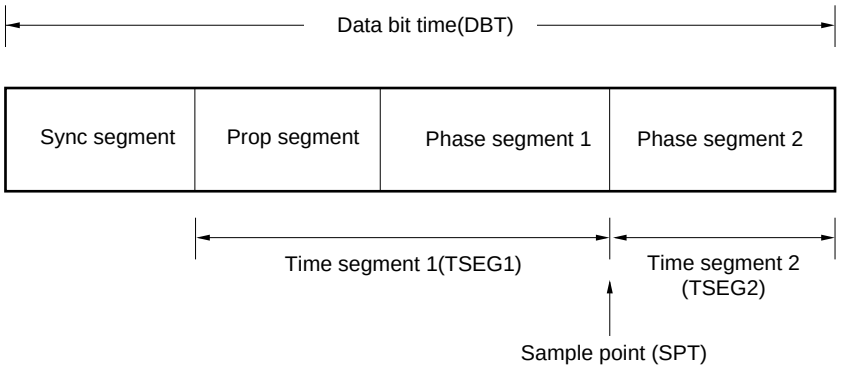
The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer basic clock (f_{TQ}) derived from the CAN module system clock (f_{CANMOD}), and divided by 1 to 256 (refer to **14.6 (12) CAN Bit Rate Prescaler Register (C0BRP, C1BRP)**).

(2) Data bit time (8-25 time quanta)

One data bit time is defined as shown in **Figure 14-18**.

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 14-18**. 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 14-18. Segment Setting

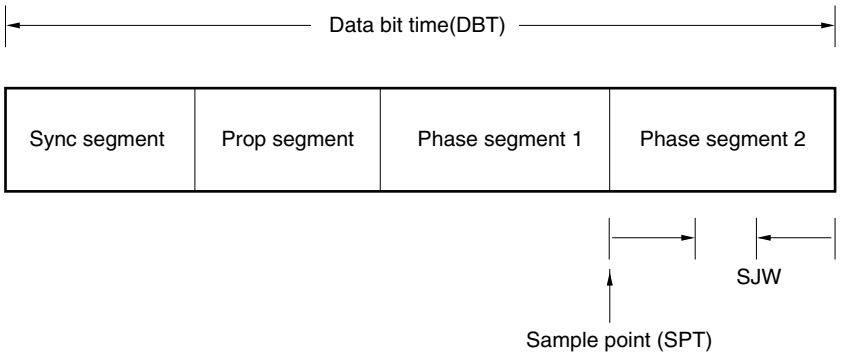


Segment Name	Settable Range	Notes on Setting to Confirm to CAN Specification
Time Segment 1 (TSEG1)	2TQ-16TQ	—
Time Segment 2 (TSEG2)	1TQ-8TQ	IPT of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length less or equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
Resynchronization jump width(SJW)	1TQ-4TQ	The length of time segment 1 minus 1TQ or 4 TQ, whichever is smaller.

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 14-19.

Figure 14-19. 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 x (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

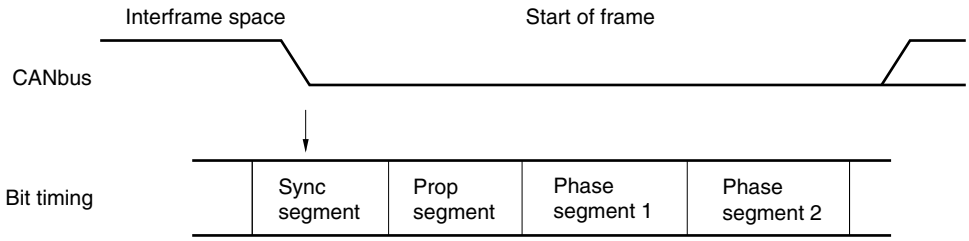
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(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 14-20. Hard-synchronization at Recognition of Dominant Level during Bus Idle

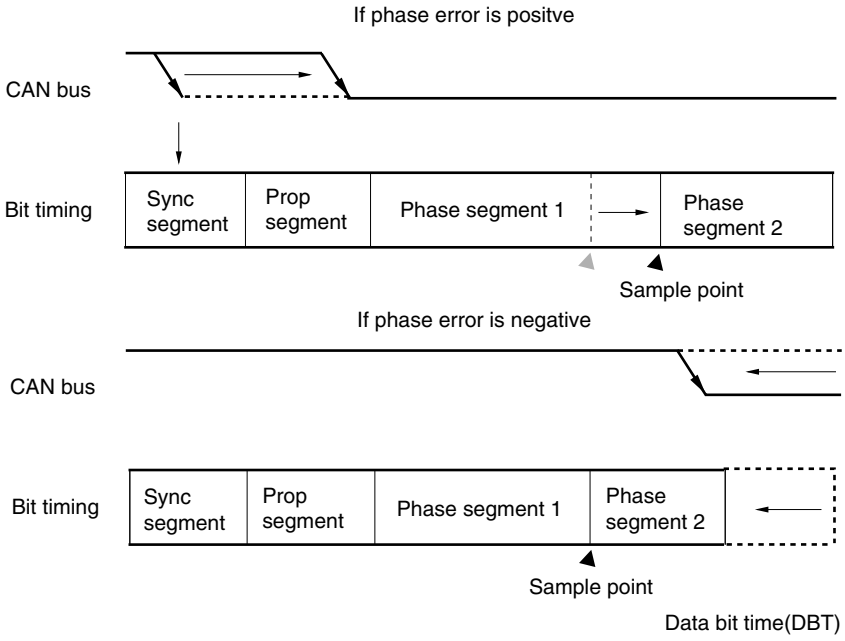


(b) Resynchronization

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.

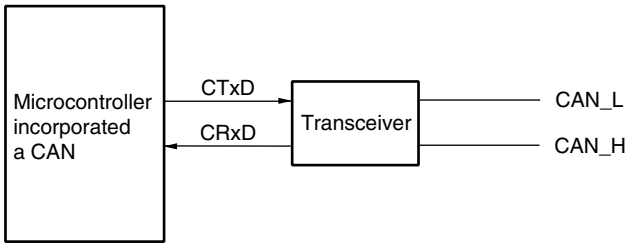
Figure 14-21. Resynchronization



14.4 Connection with Target System

The microcontroller incorporated a CAN has to be connected to the CAN bus using an external transceiver.

Figure 14-22. Connection to CAN Bus



14.5 Internal Registers of CAN Controller

14.5.1 CAN controller configuration

Table 14-15. List of CAN Controller Registers (1/2)

Item	Register Name
Control registers	Peripheral clock select register (PCKSEL)
	Serial communication pin select register 1 (STSEL1)
	Port register 0, 6, 7, 13 (P0, P6, P7, P13)
	Port mode register 0, 6, 7, 13 (PM0, PM6, PM7, PM13)
CAN global registers	CAN global module control register (C0GMCTRL, C1GMCTRL)
	CAN global module clock select register (C0GMCS, C1GMCS)
	CAN global automatic block transmission control register (C0GMABT, C1GMABT)
	CAN global automatic block transmission delay setting register (C0GMABTD, C1GMABTD)
CAN module registers	CAN module mask 1 register L (C0MASK1L, C1MASK1L)
	CAN module mask 1 register H (C0MASK1H, C1MASK1H)
	CAN module mask 2 register L (C0MASK2L, C1MASK2L)
	CAN module mask 2 register H (C0MASK2H, C1MASK2H)
	CAN module mask 3 register L (C0MASK3L, C1MASK3L)
	CAN module mask 3 register H (C0MASK3H, C1MASK3H)
	CAN module mask 4 register L (C0MASK4L, C1MASK4L)
	CAN module mask 4 register H (C0MASK4H, C1MASK4H)
	CAN module control register (C0CTRL, C1CTRL)
	CAN module last error code register (C0LEC, C1LEC)
	CAN module information register (C0INFO, C1INFO)
	CAN module error counter register (C0ERC, C1ERC)
	CAN module interrupt enable register (C0IE, C1IE)
	CAN module interrupt status register (C0INTS, C1INTS)
	CAN module bit rate prescaler register (C0BRP, C1BRP)
	CAN module bit rate register (C0BTR, C1BTR)
	CAN module last in-pointer register (C0LIPT, C1LIPT)
	CAN module receive history list register (C0RGPT, C1RGPT)
	CAN module last out-pointer register (C0LOPT, C1LOPT)
	CAN module transmit history list register (C0TGPT, C1TGPT)
	CAN module time stamp register (C0TS, C1TS)

Remark 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>.

Table 14-15. List of CAN Controller Registers (2/2)

Item	Register Name
Message buffer registers	CAN message data byte 01 register m (C0MDB01m, C1MDB01m)
	CAN message data byte 0 register m (C0MDB0m, C1MDB0m)
	CAN message data byte 1 register m (C0MDB1m, C1MDB1m)
	CAN message data byte 23 register m (C0MDB23m, C1MDB23m)
	CAN message data byte 2 register m (C0MDB2m, C1MDB2m)
	CAN message data byte 3 Register m (C0MDB3m, C1MDB3m)
	CAN message data byte 45 Register m (C0MDB45m, C1MDB45m)
	CAN message data byte 4 Register m (C0MDB4m, C1MDB4m)
	CAN message data byte 5 Register m (C0MDB5m, C1MDB5m)
	CAN message data byte 67 Register m (C0MDB67m, C1MDB67m)
	CAN message data byte 6 register m (C0MDB6m, C1MDB6m)
	CAN message data byte 7 register m (C0MDB7m, C1MDB7m)
	CAN message data length register m (C0MDLCm, C1MDLCm)
	CAN message configuration register m (C0MCONFm, C1MCONFm)
	CAN message ID register L m (C0MIDLm, C1MIDLm)
	CAN message ID register H m (C0MIDHm, C1MIDHm)
	CAN message control register m (C0MCTRLm, C1MCTRLm)

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

14.5.2 Register access type

The peripheral I/O register for the CAN controller is assigned to 000F05C0H to 000F06FFH. For details, refer to **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 14-16. Register Access Types (1/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F05C0H	CAN0 global module control register	C0GMCTRL	R/W	–	–	√	0000H
000F05C6H	CAN0 global automatic block transmission control register	C0GMABT		–	–	√	0000H
000F05C8H	CAN0 global automatic block transmission delay setting register	C0GMABTD		–	√	–	00H
000F05CEH	CAN0 global module clock select register	C0GMCS		–	√	–	0FH
000F05D0H	CAN0 module mask 1 register	C0MASK1L		–	–	√	Undefined
000F05D2H		C0MASK1H					Undefined
000F05D4H	CAN0 module mask 2 register	C0MASK2L		–	–	√	Undefined
000F05D6H		C0MASK2H					Undefined
000F05D8H	CAN0 module mask 3 register	C0MASK3L		–	–	√	Undefined
000F05DAH		C0MASK3H					Undefined
000F05DCH	CAN0 module mask 4 register	C0MASK4L		–	–	√	Undefined
000F05DEH		C0MASK4H					Undefined
000F05E0H	CAN0 module control register	C0CTRL		–	–	√	0000H
000F05E2H	CAN0 module last error code register	C0LEC		–	√	–	00H
000F05E3H	CAN0 module information register	C0INFO	R	–	√	–	00H
000F05E4H	CAN0 module error counter register	C0ERC		–	–	√	0000H
000F05E6H	CAN0 module interrupt enable register	C0IE	R/W	–	–	√	0000H
000F05E8H	CAN0 module interrupt status register	C0INTS		–	–	√	0000H
000F05EAH	CAN0 module bit rate prescaler register	C0BRP		–	√	–	FFH
000F05ECH	CAN0 module bit rate register	C0BTR		–	–	√	370FH
000F05EEH	CAN0 module last in-pointer register	C0LIPT	R	–	√	–	Undefined
000F05F0H	CAN0 module receive history list register	C0RGPT	R/W	–	–	√	xx02H
000F05F2H	CAN0 module last out-pointer register	C0LOPT	R	–	√	–	Undefined
000F05F4H	CAN0 module transmit history list register	C0TGPT	R/W	–	–	√	xx02H
000F05F6H	CAN0 module time stamp register	C0TS		–	–	√	0000H

Table 14-16. Register Access Types (2/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0600H	CAN0 message data byte 01 register 00	C0MDB0100	R/W	–	–	√	Undefined
000F0600H	CAN0 message data byte 0 register 00	C0MDB000		–	√	–	Undefined
000F0601H	CAN0 message data byte 1 register 00	C0MDB100		–	√	–	Undefined
000F0602H	CAN0 message data byte 23 register 00	C0MDB2300		–	–	√	Undefined
000F0602H	CAN0 message data byte 2 register 00	C0MDB200		–	√	–	Undefined
000F0603H	CAN0 message data byte 3 register 00	C0MDB300		–	√	–	Undefined
000F0604H	CAN0 message data byte 45 register 00	C0MDB4500		–	–	√	Undefined
000F0604H	CAN0 message data byte 4 register 00	C0MDB400		–	√	–	Undefined
000F0605H	CAN0 message data byte 5 register 00	C0MDB500		–	√	–	Undefined
000F0606H	CAN0 message data byte 67 register 00	C0MDB6700		–	–	√	Undefined
000F0606H	CAN0 message data byte 6 register 00	C0MDB600		–	√	–	Undefined
000F0607H	CAN0 message data byte 7 register 00	C0MDB700		–	√	–	Undefined
000F0608H	CAN0 message data length register 00	C0MDLC00		–	√	–	0000xxxxB
000F0609H	CAN0 message configuration register 00	C0MCONF00		–	√	–	Undefined
000F060AH	CAN0 message ID register 00	C0MIDL00		–	–	√	Undefined
000F060CH		C0MIDH00		–	–	√	Undefined
000F060EH	CAN0 message control register 00	C0MCTRL00		–	–	√	00x00000 000xx000B
000F0610H	CAN0 message data byte 01 register 01	C0MDB0101		–	–	√	Undefined
000F0610H	CAN0 message data byte 0 register 01	C0MDB001		–	√	–	Undefined
000F0611H	CAN0 message data byte 1 register 01	C0MDB101		–	√	–	Undefined
000F0612H	CAN0 message data byte 23 register 01	C0MDB2301		–	–	√	Undefined
000F0612H	CAN0 message data byte 2 register 01	C0MDB201		–	√	–	Undefined
000F0613H	CAN0 message data byte 3 register 01	C0MDB301		–	√	–	Undefined
000F0614H	CAN0 message data byte 45 register 01	C0MDB4501		–	–	√	Undefined
000F0614H	CAN0 message data byte 4 register 01	C0MDB401		–	√	–	Undefined
000F0615H	CAN0 message data byte 5 register 01	C0MDB501		–	√	–	Undefined
000F0616H	CAN0 message data byte 67 register 01	C0MDB6701		–	–	√	Undefined
000F0616H	CAN0 message data byte 6 register 01	C0MDB601		–	√	–	Undefined
000F0617H	CAN0 message data byte 7 register 01	C0MDB701		–	√	–	Undefined
000F0618H	CAN0 message data length register 01	C0MDLC01		–	√	–	0000xxxxB
000F0619H	CAN0 message configuration register 01	C0MCONF01		–	√	–	Undefined
000F061AH	CAN0 message ID register 01	C0MIDL01		–	–	√	Undefined
000F061CH		C0MIDH01		–	–	√	Undefined
000F061EH	CAN0 message control register 01	C0MCTRL01		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (3/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0620H	CAN0 message data byte 01 register 02	C0MDB0102	R/W	–	–	√	Undefined
000F0620H	CAN0 message data byte 0 register 02	C0MDB002		–	√	–	Undefined
000F0621H	CAN0 message data byte 1 register 02	C0MDB102		–	√	–	Undefined
000F0622H	CAN0 message data byte 23 register 02	C0MDB2302		–	–	√	Undefined
000F0622H	CAN0 message data byte 2 register 02	C0MDB202		–	√	–	Undefined
000F0623H	CAN0 message data byte 3 register 02	C0MDB302		–	√	–	Undefined
000F0624H	CAN0 message data byte 45 register 02	C0MDB4502		–	–	√	Undefined
000F0624H	CAN0 message data byte 4 register 02	C0MDB402		–	√	–	Undefined
000F625H	CAN0 message data byte 5 register 02	C0MDB502		–	√	–	Undefined
000F0626H	CAN0 message data byte 67 register 02	C0MDB6702		–	–	√	Undefined
000F0626H	CAN0 message data byte 6 register 02	C0MDB602		–	√	–	Undefined
000F0627H	CAN0 message data byte 7 register 02	C0MDB702		–	√	–	Undefined
000F0628H	CAN0 message data length register 02	C0MDLC02		–	√	–	0000xxxxB
000F0629H	CAN0 message configuration register 02	C0MCONF02		–	√	–	Undefined
000F062AH	CAN0 message ID register 02	C0MIDL02		–	–	√	Undefined
000F062CH		C0MIDH02		–	–	√	Undefined
000F062EH	CAN0 message control register 02	C0MCTRL02		–	–	√	00x00000 000xx000B
000F0630H	CAN0 message data byte 01 register 03	C0MDB0103		–	–	√	Undefined
000F0630H	CAN0 message data byte 0 register 03	C0MDB003		–	√	–	Undefined
000F0631H	CAN0 message data byte 1 register 03	C0MDB103		–	√	–	Undefined
000F0632H	CAN0 message data byte 23 register 03	C0MDB2303		–	–	√	Undefined
000F0632H	CAN0 message data byte 2 register 03	C0MDB203		–	√	–	Undefined
000F0633H	CAN0 message data byte 3 register 03	C0MDB303		–	√	–	Undefined
000F0634H	CAN0 message data byte 45 register 03	C0MDB4503		–	–	√	Undefined
000F0634H	CAN0 message data byte 4 register 03	C0MDB403		–	√	–	Undefined
000F0635H	CAN0 message data byte 5 register 03	C0MDB503		–	√	–	Undefined
000F0636H	CAN0 message data byte 67 register 03	C0MDB6703		–	–	√	Undefined
000F0636H	CAN0 message data byte 6 register 03	C0MDB603		–	√	–	Undefined
000F0637H	CAN0 message data byte 7 register 03	C0MDB703		–	√	–	Undefined
000F0638H	CAN0 message data length register 03	C0MDLC03		–	√	–	0000xxxxB
000F0639H	CAN0 message configuration register 03	C0MCONF03		–	√	–	Undefined
000F063AH	CAN0 message ID register 03	C0MIDL03		–	–	√	Undefined
000F063CH		C0MIDH03		–	–	√	Undefined
000F063EH	CAN0 message control register 03	C0MCTRL03		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (4/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0640H	CAN0 message data byte 01 register 04	C0MDB0104	R/W	–	–	√	Undefined
000F0640H	CAN0 message data byte 0 register 04	C0MDB004		–	√	–	Undefined
000F0641H	CAN0 message data byte 1 register 04	C0MDB104		–	√	–	Undefined
000F0642H	CAN0 message data byte 23 register 04	C0MDB2304		–	–	√	Undefined
000F0642H	CAN0 message data byte 2 register 04	C0MDB204		–	√	–	Undefined
000F0643H	CAN0 message data byte 3 register 04	C0MDB304		–	√	–	Undefined
000F0644H	CAN0 message data byte 45 register 04	C0MDB4504		–	–	√	Undefined
000F0644H	CAN0 message data byte 4 register 04	C0MDB404		–	√	–	Undefined
000F0645H	CAN0 message data byte 5 register 04	C0MDB504		–	√	–	Undefined
000F0646H	CAN0 message data byte 67 register 04	C0MDB6704		–	–	√	Undefined
000F0646H	CAN0 message data byte 6 register 04	C0MDB604		–	√	–	Undefined
000F0647H	CAN0 message data byte 7 register 04	C0MDB704		–	√	–	Undefined
000F0648H	CAN0 message data length register 04	C0MDLC04		–	√	–	0000xxxxB
000F0649H	CAN0 message configuration register 04	C0MCONF04		–	√	–	Undefined
000F064AH	CAN0 message ID register 04	C0MIDL04		–	–	√	Undefined
000F064CH		C0MIDH04		–	–	√	Undefined
000F064EH	CAN0 message control register 04	C0MCTRL04		–	–	√	00x00000 000xx000B
000F0650H	CAN0 message data byte 01 register 05	C0MDB0105		–	–	√	Undefined
000F0650H	CAN0 message data byte 0 register 05	C0MDB005		–	√	–	Undefined
000F0651H	CAN0 message data byte 1 register 05	C0MDB105		–	√	–	Undefined
000F0652H	CAN0 message data byte 23 register 05	C0MDB2305		–	–	√	Undefined
000F0652H	CAN0 message data byte 2 register 05	C0MDB205		–	√	–	Undefined
000F0653H	CAN0 message data byte 3 register 05	C0MDB305		–	√	–	Undefined
000F0654H	CAN0 message data byte 45 register 05	C0MDB4505		–	–	√	Undefined
000F0654H	CAN0 message data byte 4 register 05	C0MDB405		–	√	–	Undefined
000F0655H	CAN0 message data byte 5 register 05	C0MDB505		–	√	–	Undefined
000F0656H	CAN0 message data byte 67 register 05	C0MDB6705		–	–	√	Undefined
000F0656H	CAN0 message data byte 6 register 05	C0MDB605		–	√	–	Undefined
000F0657H	CAN0 message data byte 7 register 05	C0MDB705		–	√	–	Undefined
000F0658H	CAN0 message data length register 05	C0MDLC05		–	√	–	0000xxxxB
000F0659H	CAN0 message configuration register 05	C0MCONF05		–	√	–	Undefined
000F065AH	CAN0 message ID register 05	C0MIDL05		–	–	√	Undefined
000F065CH		C0MIDH05		–	–	√	Undefined
000F065EH	CAN0 message configuration register 05	C0MCTRL05		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (5/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0660H	CAN0 message data byte 01 register 06	C0MDB0106	R/W	–	–	√	Undefined
000F0660H	CAN0 message data byte 0 register 06	C0MDB006		–	√	–	Undefined
000F0661H	CAN0 message data byte 1 register 06	C0MDB106		–	√	–	Undefined
000F0662H	CAN0 message data byte 23 register 06	C0MDB2306		–	–	√	Undefined
000F0662H	CAN0 message data byte 2 register 06	C0MDB206		–	√	–	Undefined
000F0663H	CAN0 message data byte 3 register 06	C0MDB306		–	√	–	Undefined
000F0664H	CAN0 message data byte 45 register 06	C0MDB4506		–	–	√	Undefined
000F0664H	CAN0 message data byte 4 register 06	C0MDB406		–	√	–	Undefined
000F0665H	CAN0 message data byte 5 register 06	C0MDB506		–	√	–	Undefined
000F0666H	CAN0 message data byte 67 register 06	C0MDB6706		–	–	√	Undefined
000F0666H	CAN0 message data byte 6 register 06	C0MDB606		–	√	–	Undefined
000F0667H	CAN0 message data byte 7 register 06	C0MDB706		–	√	–	Undefined
000F0668H	CAN0 message data length register 06	C0MDLC06		–	√	–	0000xxxxB
000F0669H	CAN0 message configuration register 06	C0MCONF06		–	√	–	Undefined
000F066AH	CAN0 message ID register 06	C0MIDL06		–	–	√	Undefined
000F066CH		C0MIDH06		–	–	√	Undefined
000F066EH	CAN0 message control register 06	C0MCTRL06		–	–	√	00x00000 000xx000B
000F0670H	CAN0 message data byte 01 register 07	C0MDB0107		–	–	√	Undefined
000F0670H	CAN0 message data byte 0 register 07	C0MDB007		–	√	–	Undefined
000F0671H	CAN0 message data byte 1 register 07	C0MDB107		–	√	–	Undefined
000F0672H	CAN0 message data byte 23 register 07	C0MDB2307		–	–	√	Undefined
000F0672H	CAN0 message data byte 2 register 07	C0MDB207		–	√	–	Undefined
000F0673H	CAN0 message data byte 3 register 07	C0MDB307		–	√	–	Undefined
000F0674H	CAN0 message data byte 45 register 07	C0MDB4507		–	–	√	Undefined
000F0674H	CAN0 message data byte 4 register 07	C0MDB407		–	√	–	Undefined
000F0675H	CAN0 message data byte 5 register 07	C0MDB507		–	√	–	Undefined
000F0676H	CAN0 message data byte 67 register 07	C0MDB6707		–	–	√	Undefined
000F0676H	CAN0 message data byte 6 register 07	C0MDB607		–	√	–	Undefined
000F0677H	CAN0 message data byte 7 register 07	C0MDB707		–	√	–	Undefined
000F0678H	CAN0 message data length register 07	C0MDLC07		–	√	–	0000xxxxB
000F0679H	CAN0 message configuration register 07	C0MCONF07		–	√	–	Undefined
000F067AH	CAN0 message ID register 07	C0MIDL07		–	–	√	Undefined
000F067CH		C0MIDH07		–	–	√	Undefined
000F067EH	CAN0 message control register 07	C0MCTRL07		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (6/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0680H	CAN0 message data byte 01 register 08	C0MDB0108	R/W	–	–	√	Undefined
000F0680H	CAN0 message data byte 0 register 08	C0MDB008		–	√	–	Undefined
000F0681H	CAN0 message data byte 1 register 08	C0MDB108		–	√	–	Undefined
000F0682H	CAN0 message data byte 23 register 08	C0MDB2308		–	–	√	Undefined
000F0682H	CAN0 message data byte 2 register 08	C0MDB208		–	√	–	Undefined
000F0683H	CAN0 message data byte 3 register 08	C0MDB308		–	√	–	Undefined
000F0684H	CAN0 message data byte 45 register 08	C0MDB4508		–	–	√	Undefined
000F0684H	CAN0 message data byte 4 register 08	C0MDB408		–	√	–	Undefined
000F0685H	CAN0 message data byte 5 register 08	C0MDB508		–	√	–	Undefined
000F0686H	CAN0 message data byte 67 register 08	C0MDB6708		–	–	√	Undefined
000F0686H	CAN0 message data byte 6 register 08	C0MDB608		–	√	–	Undefined
000F0687H	CAN0 message data byte 7 register 08	C0MDB708		–	√	–	Undefined
000F0688H	CAN0 message data length register 08	C0MDLC08		–	√	–	0000xxxxB
000F0689H	CAN0 message configuration register 08	C0MCONF08		–	√	–	Undefined
000F068AH	CAN0 message ID register 08	C0MIDL08		–	–	√	Undefined
000F068CH		C0MIDH08		–	–	√	Undefined
000F068EH	CAN0 message control register 08	C0MCTRL08		–	–	√	00x00000 000xx000B
000F0690H	CAN0 message data byte 01 register 09	C0MDB0109		–	–	√	Undefined
000F0690H	CAN0 message data byte 0 register 09	C0MDB009		–	√	–	Undefined
000F0691H	CAN0 message data byte 1 register 09	C0MDB109		–	√	–	Undefined
000F0692H	CAN0 message data byte 23 register 09	C0MDB2309		–	–	√	Undefined
000F0692H	CAN0 message data byte 2 register 09	C0MDB209		–	√	–	Undefined
000F0693H	CAN0 message data byte 3 register 09	C0MDB309		–	√	–	Undefined
000F0694H	CAN0 message data byte 45 register 09	C0MDB4509		–	–	√	Undefined
000F0694H	CAN0 message data byte 4 register 09	C0MDB409		–	√	–	Undefined
000F0695H	CAN0 message data byte 5 register 09	C0MDB509		–	√	–	Undefined
000F0696H	CAN0 message data byte 67 register 09	C0MDB6709		–	–	√	Undefined
000F0696H	CAN0 message data byte 6 register 09	C0MDB609		–	√	–	Undefined
000F0697H	CAN0 message data byte 7 register 09	C0MDB709		–	√	–	Undefined
000F0698H	CAN0 message data length register 09	C0MDLC09		–	√	–	0000xxxxB
000F0699H	CAN0 message configuration register 09	C0MCONF09		–	√	–	Undefined
000F069AH	CAN0 message ID register 09	C0MIDL09		–	–	√	Undefined
000F069CH		C0MIDH09		–	–	√	Undefined
000F069EH	CAN0 message control register 09	C0MCTRL09		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (7/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F06A0H	CAN0 message data byte 01 register 10	C0MDB0110	R/W	–	–	√	Undefined
000F06A0H	CAN0 message data byte 0 register 10	C0MDB010		–	√	–	Undefined
000F06A1H	CAN0 message data byte 1 register 10	C0MDB110		–	√	–	Undefined
000F06A2H	CAN0 message data byte 23 register 10	C0MDB2310		–	–	√	Undefined
000F06A2H	CAN0 message data byte 2 register 10	C0MDB210		–	√	–	Undefined
000F06A3H	CAN0 message data byte 3 register 10	C0MDB310		–	√	–	Undefined
000F06A4H	CAN0 message data byte 45 register 10	C0MDB4510		–	–	√	Undefined
000F06A4H	CAN0 message data byte 4 register 10	C0MDB410		–	√	–	Undefined
000F06A5H	CAN0 message data byte 5 register 10	C0MDB510		–	√	–	Undefined
000F06A6H	CAN0 message data byte 67 register 10	C0MDB6710		–	–	√	Undefined
000F06A6H	CAN0 message data byte 6 register 10	C0MDB610		–	√	–	Undefined
000F06A7H	CAN0 message data byte 7 register 10	C0MDB710		–	√	–	Undefined
000F06A8H	CAN0 message data length register 10	C0MDLC10		–	√	–	0000xxxxB
000F06A9H	CAN0 message configuration register 10	C0MCONF10		–	√	–	Undefined
000F06AAH	CAN0 message ID register 10	C0MIDL10		–	–	√	Undefined
000F06ACH		C0MIDH10		–	–	√	Undefined
000F06AEH	CAN0 message control register 10	C0MCTRL10		–	–	√	00x00000 000xx000B
000F06B0H	CAN0 message data byte 01 register 11	C0MDB0111		–	–	√	Undefined
000F06B0H	CAN0 message data byte 0 register 11	C0MDB011		–	√	–	Undefined
000F06B1H	CAN0 message data byte 1 register 11	C0MDB111		–	√	–	Undefined
000F06B2H	CAN0 message data byte 23 register 11	C0MDB2311		–	–	√	Undefined
000F06B2H	CAN0 message data byte 2 register 11	C0MDB211		–	√	–	Undefined
000F06B3H	CAN0 message data byte 3 register 11	C0MDB311		–	√	–	Undefined
000F06B4H	CAN0 message data byte 45 register 11	C0MDB4511		–	–	√	Undefined
000F06B4H	CAN0 message data byte 4 register 11	C0MDB411		–	√	–	Undefined
000F06B5H	CAN0 message data byte 51 register 11	C0MDB511		–	√	–	Undefined
000F06B6H	CAN0 message data byte 67 register 11	C0MDB6711		–	–	√	Undefined
000F06B6H	CAN0 message data byte 6 register 11	C0MDB611		–	√	–	Undefined
000F06B7H	CAN0 message data byte 71 register 11	C0MDB711		–	√	–	Undefined
000F06B8H	CAN0 message data length register 11	C0MDLC11		–	√	–	0000xxxxB
000F06B9H	CAN0 message configuration register 11	C0MCONF11		–	√	–	Undefined
000F06BAH	CAN0 message ID register 11	C0MIDL11		–	–	√	Undefined
000F06BCH		C0MIDH11		–	–	√	Undefined
000F06BEH	CAN0 message control register 11	C0MCTRL11		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (8/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F06C0H	CAN0 message data byte 01 register 12	C0MDB0112	R/W	–	–	√	Undefined
000F06C0H	CAN0 message data byte 0 register 12	C0MDB012		–	√	–	Undefined
000F06C1H	CAN0 message data byte 1 register 12	C0MDB112		–	√	–	Undefined
000F06C2H	CAN0 message data byte 23 register 12	C0MDB2312		–	–	√	Undefined
000F06C2H	CAN0 message data byte 2 register 12	C0MDB212		–	√	–	Undefined
000F06C3H	CAN0 message data byte 3 register 12	C0MDB312		–	√	–	Undefined
000F06C4H	CAN0 message data byte 45 register 12	C0MDB4512		–	–	√	Undefined
000F06C4H	CAN0 message data byte 4 register 12	C0MDB412		–	√	–	Undefined
000F06C5H	CAN0 message data byte 5 register 12	C0MDB512		–	√	–	Undefined
000F06C6H	CAN0 message data byte 67 register 12	C0MDB6712		–	–	√	Undefined
000F06C6H	CAN0 message data byte 6 register 12	C0MDB612		–	√	–	Undefined
000F06C7H	CAN0 message data byte 7 register 12	C0MDB712		–	√	–	Undefined
000F06C8H	CAN0 message data length register 12	C0MDLC12		–	√	–	0000xxxxB
000F06C9H	CAN0 message configuration register 12	C0MCONF12		–	√	–	Undefined
000F06CAH	CAN0 message ID register 12	C0MIDL12		–	–	√	Undefined
000F06CCH		C0MIDH12		–	–	√	Undefined
000F06CEH	CAN0 message control register 12	C0MCTRL12		–	–	√	00x00000 000xx000B
000F06D0H	CAN0 message data byte 01 register 13	C0MDB0113		–	–	√	Undefined
000F06D0H	CAN0 message data byte 0 register 13	C0MDB013		–	√	–	Undefined
000F06D1H	CAN0 message data byte 1 register 13	C0MDB113		–	√	–	Undefined
000F06D2H	CAN0 message data byte 23 register 13	C0MDB2313		–	–	√	Undefined
000F06D2H	CAN0 message data byte 2 register 13	C0MDB213		–	√	–	Undefined
000F06D3H	CAN0 message data byte 3 register 13	C0MDB313		–	√	–	Undefined
000F06D4H	CAN0 message data byte 45 register 13	C0MDB4513		–	–	√	Undefined
000F06D4H	CAN0 message data byte 4 register 13	C0MDB413		–	√	–	Undefined
000F06D5H	CAN0 message data byte 5 register 13	C0MDB513		–	√	–	Undefined
000F06D6H	CAN0 message data byte 67 register 13	C0MDB6713		–	–	√	Undefined
000F06D6H	CAN0 message data byte 6 register 13	C0MDB613		–	√	–	Undefined
000F06D7H	CAN0 message data byte 7 register 13	C0MDB713		–	√	–	Undefined
000F06D8H	CAN0 message data length register 13	C0MDLC13		–	√	–	0000xxxxB
000F06D9H	CAN0 message configuration register 13	C0MCONF13		–	√	–	Undefined
000F06DAH	CAN0 message ID register 13	C0MIDL13		–	–	√	Undefined
000F06DCH		C0MIDH13		–	–	√	Undefined
000F06DEH	CAN0 message control register 13	C0MCTRL13		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (9/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F06E0H	CAN0 message data byte 01 register 14	C0MDB0114	R/W	–	–	√	Undefined
000F06E0H	CAN0 message data byte 0 register 14	C0MDB014		–	√	–	Undefined
000F06E1H	CAN0 message data byte 1 register 14	C0MDB114		–	√	–	Undefined
000F06E2H	CAN0 message data byte 23 register 14	C0MDB2314		–	–	√	Undefined
000F06E2H	CAN0 message data byte 2 register 14	C0MDB214		–	√	–	Undefined
000F06E3H	CAN0 message data byte 3 register 14	C0MDB314		–	√	–	Undefined
000F06E4H	CAN0 message data byte 45 register 14	C0MDB4514		–	–	√	Undefined
000F06E4H	CAN0 message data byte 4 register 14	C0MDB414		–	√	–	Undefined
000F06E5H	CAN0 message data byte 5 register 14	C0MDB514		–	√	–	Undefined
000F06E6H	CAN0 message data byte 67 register 14	C0MDB6714		–	–	√	Undefined
000F06E6H	CAN0 message data byte 6 register 14	C0MDB614		–	√	–	Undefined
000F06E7H	CAN0 message data byte 7 register 14	C0MDB714		–	√	–	Undefined
000F06E8H	CAN0 message data length register 14	C0MDLC14		–	√	–	0000xxxxB
000F06E9H	CAN0 message configuration register 14	C0MCONF14		–	√	–	Undefined
000F06EAH	CAN0 message ID register 14	C0MIDL14		–	–	√	Undefined
000F06ECH		C0MIDH14		–	–	√	Undefined
000F06EEH	CAN0 message control register 14	C0MCTRL14		–	–	√	00x00000 000xx000B
000F06F0H	CAN0 message data byte 01 register 15	C0MDB0115		–	–	√	Undefined
000F06F0H	CAN0 message data byte 0 register 15	C0MDB015		–	√	–	Undefined
000F06F1H	CAN0 message data byte 1 register 15	C0MDB115		–	√	–	Undefined
000F06F2H	CAN0 message data byte 23 register 15	C0MDB2315		–	–	√	Undefined
000F06F2H	CAN0 message data byte 2 register 15	C0MDB215		–	√	–	Undefined
000F06F3H	CAN0 message data byte 3 register 15	C0MDB315		–	√	–	Undefined
000F06F4H	CAN0 message data byte 45 register 15	C0MDB4515		–	–	√	Undefined
000F06F4H	CAN0 message data byte 4 register 15	C0MDB415		–	√	–	Undefined
000F06F5H	CAN0 message data byte 5 register 15	C0MDB515		–	√	–	Undefined
000F06F6H	CAN0 message data byte 67 register 15	C0MDB6715		–	–	√	Undefined
000F06F6H	CAN0 message data byte 6 register 15	C0MDB615		–	√	–	Undefined
000F06F7H	CAN0 message data byte 7 register 15	C0MDB715		–	√	–	Undefined
000F06F8H	CAN0 message data length register 15	C0MDLC15		–	√	–	0000xxxxB
000F06F9H	CAN0 message configuration register 15	C0MCONF15		–	√	–	Undefined
000F06FAH	CAN0 message ID register 15	C0MIDL15		–	–	√	Undefined
000F06FCH		C0MIDH15		–	–	√	Undefined
000F06FEH	CAN0 message control register 15	C0MCTRL15		–	–	√	00x00000 000xx000B
000F0340H	CAN1 global module control register	C1GMCTRL		–	–	√	0000H
000F0342H	CAN1 global module clock select register	C1GMCS		–	√	–	0FH
000F0346H	CAN1 global automatic block transmission control register	C1GMABT		–	–	√	0000H
000F0348H	CAN1 global automatic block transmission delay setting register	C1GMABTD		–	√	–	00H

Table 14-16. Register Access Types (10/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0380H	CAN1 module mask 1 register L	C1MASK1L	R/W	–	–	√	Undefined
000F0382H	CAN1 module mask 1 register H	C1MASK1H		–	–	√	Undefined
000F0384H	CAN1 module mask 2 register L	C1MASK2L		–	–	√	Undefined
000F0386H	CAN1 module mask 2 register H	C1MASK2H		–	–	√	Undefined
000F0388H	CAN1 module mask 3 register L	C1MASK3L		–	–	√	Undefined
000F038AH	CAN1 module mask 3 register H	C1MASK3H		–	–	√	Undefined
000F038CH	CAN1 module mask 4 register L	C1MASK4L		–	–	√	Undefined
000F038EH	CAN1 module mask 4 register H	C1MASK4H		–	–	√	Undefined
000F0390H	CAN1 module control register	C1CTRL		–	–	√	0000H
000F0392H	CAN1 module last error code register	C1LEC		–	√	–	00H
000F0393H	CAN1 module information register	C1INFO	R	–	√	–	00H
000F0394H	CAN1 module error counter register	C1ERC		–	–	√	0000H
000F0396H	CAN1 module interrupt enable register	C1IE	R/W	–	–	√	0000H
000F0398H	CAN1 module interrupt status register	C1INTS		–	–	√	0000H
000F039AH	CAN1 module bit rate prescaler register	C1BRP		–	√	–	FFH
000F039CH	CAN1 module bit rate register	C1BTR		–	–	√	370FH
000F039EH	CAN1 module last in-pointer register	C1LIPT	R	–	√	–	Undefined
000F03A0H	CAN1 module receive history list register	C1RGPT	R/W	–	–	√	xx02H
000F03A2H	CAN1 module last out-pointer register	C1LOPT	R	–	√	–	Undefined
000F03A4H	CAN1 module transmit history list register	C1TGPT	R/W	–	–	√	xx02H
000F03A6H	CAN1 module time stamp register	C1TS		–	–	√	0000H
000F0400H	CAN1 message data byte 01 register 00	C1MDB0100		–	–	√	Undefined
000F0400H	CAN1 message data byte 0 register 00	C1MDB000		–	√	–	Undefined
000F0401H	CAN1 message data byte 1 register 00	C1MDB100		–	√	–	Undefined
000F0402H	CAN1 message data byte 23 register 00	C1MDB2300		–	–	√	Undefined
000F0402H	CAN1 message data byte 2 register 00	C1MDB200		–	√	–	Undefined
000F0403H	CAN1 message data byte 3 register 00	C1MDB300		–	√	–	Undefined
000F0404H	CAN1 message data byte 45 register 00	C1MDB4500		–	–	√	Undefined
000F0404H	CAN1 message data byte 4 register 00	C1MDB400		–	√	–	Undefined
000F0405H	CAN1 message data byte 5 register 00	C1MDB500		–	√	–	Undefined
000F0406H	CAN1 message data byte 67 register 00	C1MDB6700		–	–	√	Undefined
000F0406H	CAN1 message data byte 6 register 00	C1MDB600		–	√	–	Undefined
000F0407H	CAN1 message data byte 7 register 00	C1MDB700		–	√	–	Undefined
000F0408H	CAN1 message data length register 00	C1MDLC00		–	√	–	0000xxxxB
000F0409H	CAN1 message configuration register 00	C1MCONF00		–	√	–	Undefined
000F040AH	CAN1 message ID register 00 L	C1MIDL00		–	–	√	Undefined
000F040CH	CAN1 message ID register 00 H	C1MIDH00		–	–	√	Undefined
000F040EH	CAN1 message control register 00	C1MCTRL00		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (11/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0410H	CAN1 message data byte 01 register 01	C1MDB0101	R/W	–	–	√	Undefined
000F0410H	CAN1 message data byte 0 register 01	C1MDB001		–	√	–	Undefined
000F0411H	CAN1 message data byte 1 register 01	C1MDB101		–	√	–	Undefined
000F0412H	CAN1 message data byte 23 register 01	C1MDB2301		–	–	√	Undefined
000F0412H	CAN1 message data byte 2 register 01	C1MDB201		–	√	–	Undefined
000F0413H	CAN1 message data byte 3 register 01	C1MDB301		–	√	–	Undefined
000F0414H	CAN1 message data byte 45 register 01	C1MDB4501		–	–	√	Undefined
000F0414H	CAN1 message data byte 4 register 01	C1MDB401		–	√	–	Undefined
000F0415H	CAN1 message data byte 5 register 01	C1MDB501		–	√	–	Undefined
000F0416H	CAN1 message data byte 67 register 01	C1MDB6701		–	–	√	Undefined
000F0416H	CAN1 message data byte 6 register 01	C1MDB601		–	√	–	Undefined
000F0417H	CAN1 message data byte 7 register 01	C1MDB701		–	√	–	Undefined
000F0418H	CAN1 message data length register 01	C1MDLC01		–	√	–	0000xxxxB
000F0419H	CAN1 message configuration register 01	C1MCONF01		–	√	–	Undefined
000F041AH	CAN1 message ID register 01 L	C1MIDL01		–	–	√	Undefined
000F041CH	CAN1 message ID register 01 H	C1MIDH01		–	–	√	Undefined
000F041EH	CAN1 message control register 01	C1MCTRL01		–	–	√	00x00000 000xx000B
000F0420H	CAN1 message data byte 01 register 02	C1MDB0102		–	–	√	Undefined
000F0420H	CAN1 message data byte 0 register 02	C1MDB002		–	√	–	Undefined
000F0421H	CAN1 message data byte 1 register 02	C1MDB102		–	√	–	Undefined
000F0422H	CAN1 message data byte 23 register 02	C1MDB2302		–	–	√	Undefined
000F0422H	CAN1 message data byte 2 register 02	C1MDB202		–	√	–	Undefined
000F0423H	CAN1 message data byte 3 register 02	C1MDB302		–	√	–	Undefined
000F0424H	CAN1 message data byte 45 register 02	C1MDB4502		–	–	√	Undefined
000F0424H	CAN1 message data byte 4 register 02	C1MDB402		–	√	–	Undefined
000F0425H	CAN1 message data byte 5 register 02	C1MDB502		–	√	–	Undefined
000F0426H	CAN1 message data byte 67 register 02	C1MDB6702		–	–	√	Undefined
000F0426H	CAN1 message data byte 6 register 02	C1MDB602		–	√	–	Undefined
000F0427H	CAN1 message data byte 7 register 02	C1MDB702		–	√	–	Undefined
000F0428H	CAN1 message data length register 02	C1MDLC02		–	√	–	0000xxxxB
000F0429H	CAN1 message configuration register 02	C1MCONF02		–	√	–	Undefined
000F042AH	CAN1 message ID register 02 L	C1MIDL02		–	–	√	Undefined
000F042CH	CAN1 message ID register 02 H	C1MIDH02		–	–	√	Undefined
000F042EH	CAN1 message control register 02	C1MCTRL02		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (12/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0430H	CAN1 message data byte 01 register 03	C1MDB0103	R/W	–	–	√	Undefined
000F0430H	CAN1 message data byte 0 register 03	C1MDB003		–	√	–	Undefined
000F0431H	CAN1 message data byte 1 register 03	C1MDB103		–	√	–	Undefined
000F0432H	CAN1 message data byte 23 register 03	C1MDB2303		–	–	√	Undefined
000F0432H	CAN1 message data byte 2 register 03	C1MDB203		–	√	–	Undefined
000F0433H	CAN1 message data byte 3 register 03	C1MDB303		–	√	–	Undefined
000F0434H	CAN1 message data byte 45 register 03	C1MDB4503		–	–	√	Undefined
000F0434H	CAN1 message data byte 4 register 03	C1MDB403		–	√	–	Undefined
000F0435H	CAN1 message data byte 5 register 03	C1MDB503		–	√	–	Undefined
000F0436H	CAN1 message data byte 67 register 03	C1MDB6703		–	–	√	Undefined
000F0436H	CAN1 message data byte 6 register 03	C1MDB603		–	√	–	Undefined
000F0437H	CAN1 message data byte 7 register 03	C1MDB703		–	√	–	Undefined
000F0438H	CAN1 message data length register 03	C1MDLC03		–	√	–	0000xxxxB
000F0439H	CAN1 message configuration register 03	C1MCONF03		–	√	–	Undefined
000F043AH	CAN1 message ID register 03 L	C1MIDL03		–	–	√	Undefined
000F043CH	CAN1 message ID register 03 H	C1MIDH03		–	–	√	Undefined
000F043EH	CAN1 message control register 03	C1MCTRL03		–	–	√	00x00000 000xx000B
000F0440H	CAN1 message data byte 01 register 04	C1MDB0104		–	–	√	Undefined
000F0440H	CAN1 message data byte 0 register 04	C1MDB004		–	√	–	Undefined
000F0441H	CAN1 message data byte 1 register 04	C1MDB104		–	√	–	Undefined
000F0442H	CAN1 message data byte 23 register 04	C1MDB2304		–	–	√	Undefined
000F0442H	CAN1 message data byte 2 register 04	C1MDB204		–	√	–	Undefined
000F0443H	CAN1 message data byte 3 register 04	C1MDB304		–	√	–	Undefined
000F0444H	CAN1 message data byte 45 register 04	C1MDB4504		–	–	√	Undefined
000F0444H	CAN1 message data byte 4 register 04	C1MDB404		–	√	–	Undefined
000F0445H	CAN1 message data byte 5 register 04	C1MDB504		–	√	–	Undefined
000F0446H	CAN1 message data byte 67 register 04	C1MDB6704		–	–	√	Undefined
000F0446H	CAN1 message data byte 6 register 04	C1MDB604		–	√	–	Undefined
000F0447H	CAN1 message data byte 7 register 04	C1MDB704		–	√	–	Undefined
000F0448H	CAN1 message data length register 04	C1MDLC04		–	√	–	0000xxxxB
000F0449H	CAN1 message configuration register 04	C1MCONF04		–	√	–	Undefined
000F044AH	CAN1 message ID register 04 L	C1MIDL04		–	–	√	Undefined
000F044CH	CAN1 message ID register 04 H	C1MIDH04		–	–	√	Undefined
000F044EH	CAN1 message control register 04	C1MCTRL04		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (13/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0450H	CAN1 message data byte 01 register 05	C1MDB0105	R/W	–	–	√	Undefined
000F0450H	CAN1 message data byte 0 register 05	C1MDB005		–	√	–	
000F0451H	CAN1 message data byte 1 register 05	C1MDB105		–	√	–	
000F0452H	CAN1 message data byte 23 register 05	C1MDB2305		–	–	√	Undefined
000F0452H	CAN1 message data byte 2 register 05	C1MDB205		–	√	–	
000F0453H	CAN1 message data byte 3 register 05	C1MDB305		–	√	–	
000F0454H	CAN1 message data byte 45 register 05	C1MDB4505		–	–	√	Undefined
000F0454H	CAN1 message data byte 4 register 05	C1MDB405		–	√	–	
000F0455H	CAN1 message data byte 5 register 05	C1MDB505		–	√	–	
000F0456H	CAN1 message data byte 67 register 05	C1MDB6705		–	–	√	Undefined
000F0456H	CAN1 message data byte 6 register 05	C1MDB605		–	√	–	
000F0457H	CAN1 message data byte 7 register 05	C1MDB705		–	√	–	
000F0458H	CAN1 message data length register 05	C1MDLC05		–	√	–	0000xxxxB
000F0459H	CAN1 message configuration register 05	C1MCONF05		–	√	–	Undefined
000F045AH	CAN1 message ID register 05 L	C1MIDL05		–	–	√	Undefined
000F045CH	CAN1 message ID register 05 H	C1MIDH05		–	–	√	Undefined
000F045EH	CAN1 message configuration register 05	C1MCTRL05		–	–	√	00x00000 000xx000B
000F0460H	CAN1 message data byte 01 register 06	C1MDB0106		–	–	√	Undefined
000F0460H	CAN1 message data byte 0 register 06	C1MDB006		–	√	–	Undefined
000F0461H	CAN1 message data byte 1 register 06	C1MDB106		–	√	–	Undefined
000F0462H	CAN1 message data byte 23 register 06	C1MDB2306		–	–	√	Undefined
000F0462H	CAN1 message data byte 2 register 06	C1MDB206		–	√	–	Undefined
000F0463H	CAN1 message data byte 3 register 06	C1MDB306		–	√	–	Undefined
000F0464H	CAN1 message data byte 45 register 06	C1MDB4506		–	–	√	Undefined
000F0464H	CAN1 message data byte 4 register 06	C1MDB406		–	√	–	Undefined
000F0465H	CAN1 message data byte 5 register 06	C1MDB506		–	√	–	Undefined
000F0466H	CAN1 message data byte 67 register 06	C1MDB6706		–	–	√	Undefined
000F0466H	CAN1 message data byte 6 register 06	C1MDB606		–	√	–	Undefined
000F0467H	CAN1 message data byte 7 register 06	C1MDB706		–	√	–	Undefined
000F0468H	CAN1 message data length register 06	C1MDLC06		–	√	–	0000xxxxB
000F0469H	CAN1 message configuration register 06	C1MCONF06		–	√	–	Undefined
000F046AH	CAN1 message ID register 06 L	C1MIDL06		–	–	√	Undefined
000F046CH	CAN1 message ID register 06 H	C1MIDH06		–	–	√	Undefined
000F046EH	CAN1 message control register 06	C1MCTRL06		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (14/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0470H	CAN1 message data byte 01 register 07	C1MDB0107	R/W	–	–	√	Undefined
000F0470H	CAN1 message data byte 0 register 07	C1MDB007		–	√	–	Undefined
000F0471H	CAN1 message data byte 1 register 07	C1MDB107		–	√	–	Undefined
000F0472H	CAN1 message data byte 23 register 07	C1MDB2307		–	–	√	Undefined
000F0472H	CAN1 message data byte 2 register 07	C1MDB207		–	√	–	Undefined
000F0473H	CAN1 message data byte 3 register 07	C1MDB307		–	√	–	Undefined
000F0474H	CAN1 message data byte 45 register 07	C1MDB4507		–	–	√	Undefined
000F0474H	CAN1 message data byte 4 register 07	C1MDB407		–	√	–	Undefined
000F0475H	CAN1 message data byte 5 register 07	C1MDB507		–	√	–	Undefined
000F0476H	CAN1 message data byte 67 register 07	C1MDB6707		–	–	√	Undefined
000F0476H	CAN1 message data byte 6 register 07	C1MDB607		–	√	–	Undefined
000F0477H	CAN1 message data byte 7 register 07	C1MDB707		–	√	–	Undefined
000F0478H	CAN1 message data length register 07	C1MDLC07		–	√	–	0000xxxxB
000F0479H	CAN1 message configuration register 07	C1MCONF07		–	√	–	Undefined
000F047AH	CAN1 message ID register 07 L	C1MIDL07		–	–	√	Undefined
000F047CH	CAN1 message ID register 07 H	C1MIDH07		–	–	√	Undefined
000F047EH	CAN1 message control register 07	C1MCTRL07		–	–	√	00x00000 000xx000B
000F0480H	CAN1 message data byte 01 register 08	C1MDB0108		–	–	√	Undefined
000F0480H	CAN1 message data byte 0 register 08	C1MDB008		–	√	–	Undefined
000F0481H	CAN1 message data byte 1 register 08	C1MDB108		–	√	–	Undefined
000F0482H	CAN1 message data byte 23 register 08	C1MDB2308		–	–	√	Undefined
000F0482H	CAN1 message data byte 2 register 08	C1MDB208		–	√	–	Undefined
000F0483H	CAN1 message data byte 3 register 08	C1MDB308		–	√	–	Undefined
000F0484H	CAN1 message data byte 45 register 08	C1MDB4508		–	–	√	Undefined
000F0484H	CAN1 message data byte 4 register 08	C1MDB408		–	√	–	Undefined
000F0485H	CAN1 message data byte 5 register 08	C1MDB508		–	√	–	Undefined
000F0486H	CAN1 message data byte 67 register 08	C1MDB6708		–	–	√	Undefined
000F0486H	CAN1 message data byte 6 register 08	C1MDB608		–	√	–	Undefined
000F0487H	CAN1 message data byte 7 register 08	C1MDB708		–	√	–	Undefined
000F0488H	CAN1 message data length register 08	C1MDLC08		–	√	–	0000xxxxB
000F0489H	CAN1 message configuration register 08	C1MCONF08		–	√	–	Undefined
000F048AH	CAN1 message ID register 08 H	C1MIDL08		–	–	√	Undefined
000F048CH	CAN1 message ID register 08 L	C1MIDH08		–	–	√	Undefined
000F048EH	CAN1 message control register 08	C1MCTRL08		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (15/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F0490H	CAN1 message data byte 01 register 09	C1MDB0109	R/W	–	–	√	Undefined
000F0490H	CAN1 message data byte 0 register 09	C1MDB009		–	√	–	Undefined
000F0491H	CAN1 message data byte 1 register 09	C1MDB109		–	√	–	Undefined
000F0492H	CAN1 message data byte 23 register 09	C1MDB2309		–	–	√	Undefined
000F0492H	CAN1 message data byte 2 register 09	C1MDB209		–	√	–	Undefined
000F0493H	CAN1 message data byte 3 register 09	C1MDB309		–	√	–	Undefined
000F0494H	CAN1 message data byte 45 register 09	C1MDB4509		–	–	√	Undefined
000F0494H	CAN1 message data byte 4 register 09	C1MDB409		–	√	–	Undefined
000F0495H	CAN1 message data byte 5 register 09	C1MDB509		–	√	–	Undefined
000F0496H	CAN1 message data byte 67 register 09	C1MDB6709		–	–	√	Undefined
000F0496H	CAN1 message data byte 6 register 09	C1MDB609		–	√	–	Undefined
000F0497H	CAN1 message data byte 7 register 09	C1MDB709		–	√	–	Undefined
000F0498H	CAN1 message data length register 09	C1MDLC09		–	√	–	0000xxxxB
000F0499H	CAN1 message configuration register 09	C1MCONF09		–	√	–	Undefined
000F049AH	CAN1 message ID register 09 L	C1MIDL09		–	–	√	Undefined
000F049CH	CAN1 message ID register 09 H	C1MIDH09		–	–	√	Undefined
000F049EH	CAN1 message control register 09	C1MCTRL09		–	–	√	00x00000 000xx000B
000F04A0H	CAN1 message data byte 01 register 10	C1MDB0110		–	–	√	Undefined
000F04A0H	CAN1 message data byte 0 register 10	C1MDB010		–	√	–	Undefined
000F04A1H	CAN1 message data byte 1 register 10	C1MDB110		–	√	–	Undefined
000F04A2H	CAN1 message data byte 23 register 10	C1MDB2310		–	–	√	Undefined
000F04A2H	CAN1 message data byte 2 register 10	C1MDB210		–	√	–	Undefined
000F04A3H	CAN1 message data byte 3 register 10	C1MDB310		–	√	–	Undefined
000F04A4H	CAN1 message data byte 45 register 10	C1MDB4510		–	–	√	Undefined
000F04A4H	CAN1 message data byte 4 register 10	C1MDB410		–	√	–	Undefined
000F04A5H	CAN1 message data byte 5 register 10	C1MDB510		–	√	–	Undefined
000F04A6H	CAN1 message data byte 67 register 10	C1MDB6710		–	–	√	Undefined
000F04A6H	CAN1 message data byte 6 register 10	C1MDB610		–	√	–	Undefined
000F04A7H	CAN1 message data byte 7 register 10	C1MDB710		–	√	–	Undefined
000F04A8H	CAN1 message data length register 10	C1MDLC10		–	√	–	0000xxxxB
000F04A9H	CAN1 message configuration register 10	C1MCONF10		–	√	–	Undefined
000F04AAH	CAN1 message ID register 10 L	C1MIDL10		–	–	√	Undefined
000F04ACH	CAN1 message ID register 10 H	C1MIDH10		–	–	√	Undefined
000F04AEH	CAN1 message control register 10	C1MCTRL10		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (16/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F04B0H	CAN1 message data byte 01 register 11	C1MDB0111	R/W	–	–	√	Undefined
000F04B0H	CAN1 message data byte 0 register 11	C1MDB011		–	√	–	Undefined
000F04B1H	CAN1 message data byte 1 register 11	C1MDB111		–	√	–	Undefined
000F04B2H	CAN1 message data byte 23 register 11	C1MDB2311		–	–	√	Undefined
000F04B2H	CAN1 message data byte 2 register 11	C1MDB211		–	√	–	Undefined
000F04B3H	CAN1 message data byte 3 register 11	C1MDB311		–	√	–	Undefined
000F04B4H	CAN1 message data byte 45 register 11	C1MDB4511		–	–	√	Undefined
000F04B4H	CAN1 message data byte 4 register 11	C1MDB411		–	√	–	Undefined
000F04B5H	CAN1 message data byte 51 register 11	C1MDB511		–	√	–	Undefined
000F04B6H	CAN1 message data byte 67 register 11	C1MDB6711		–	–	√	Undefined
000F04B6H	CAN1 message data byte 6 register 11	C1MDB611		–	√	–	Undefined
000F04B7H	CAN1 message data byte 71 register 11	C1MDB711		–	√	–	Undefined
000F04B8H	CAN1 message data length register 11	C1MDLC11		–	√	–	0000xxxxB
000F04B9H	CAN1 message configuration register 11	C1MCONF11		–	√	–	Undefined
000F04BAH	CAN1 message ID register 11 L	C1MIDL11		–	–	√	Undefined
000F04BCH	CAN1 message ID register 11 H	C1MIDH11		–	–	√	Undefined
000F04BEH	CAN1 message control register 11	C1MCTRL11		–	–	√	00x00000 000xx000B
000F04C0H	CAN1 message data byte 01 register 12	C1MDB0112		–	–	√	Undefined
000F04C0H	CAN1 message data byte 0 register 12	C1MDB012		–	√	–	Undefined
000F04C1H	CAN1 message data byte 1 register 12	C1MDB112		–	√	–	Undefined
000F04C2H	CAN1 message data byte 23 register 12	C1MDB2312		–	–	√	Undefined
000F04C2H	CAN1 message data byte 2 register 12	C1MDB212		–	√	–	Undefined
000F04C3H	CAN1 message data byte 3 register 12	C1MDB312		–	√	–	Undefined
000F04C4H	CAN1 message data byte 45 register 12	C1MDB4512		–	–	√	Undefined
000F04C4H	CAN1 message data byte 4 register 12	C1MDB412		–	√	–	Undefined
000F04C5H	CAN1 message data byte 5 register 12	C1MDB512		–	√	–	Undefined
000F04C6H	CAN1 message data byte 67 register 12	C1MDB6712		–	–	√	Undefined
000F04C6H	CAN1 message data byte 6 register 12	C1MDB612		–	√	–	Undefined
000F04C7H	CAN1 message data byte 7 register 12	C1MDB712		–	√	–	Undefined
000F04C8H	CAN1 message data length register 12	C1MDLC12		–	√	–	0000xxxxB
000F04C9H	CAN1 message configuration register 12	C1MCONF12		–	√	–	Undefined
000F04CAH	CAN1 message ID register 12 L	C1MIDL12		–	–	√	Undefined
000F04CCH	CAN1 message ID register 12 H	C1MIDH12		–	–	√	Undefined
000F04CEH	CAN1 message control register 12	C1MCTRL12		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (17/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F04D0H	CAN1 message data byte 01 register 13	C1MDB0113	R/W	–	–	√	Undefined
000F04D0H	CAN1 message data byte 0 register 13	C1MDB013		–	√	–	Undefined
000F04D1H	CAN1 message data byte 1 register 13	C1MDB113		–	√	–	Undefined
000F04D2H	CAN1 message data byte 23 register 13	C1MDB2313		–	–	√	Undefined
000F04D2H	CAN1 message data byte 2 register 13	C1MDB213		–	√	–	Undefined
000F04D3H	CAN1 message data byte 3 register 13	C1MDB313		–	√	–	Undefined
000F04D4H	CAN1 message data byte 45 register 13	C1MDB4513		–	–	√	Undefined
000F04D4H	CAN1 message data byte 4 register 13	C1MDB413		–	√	–	Undefined
000F04D5H	CAN1 message data byte 5 register 13	C1MDB513		–	√	–	Undefined
000F04D6H	CAN1 message data byte 67 register 13	C1MDB6713		–	–	√	Undefined
000F04D6H	CAN1 message data byte 6 register 13	C1MDB613		–	√	–	Undefined
000F04D7H	CAN1 message data byte 7 register 13	C1MDB713		–	√	–	Undefined
000F04D8H	CAN1 message data length register 13	C1MDLC13		–	√	–	0000xxxxB
000F04D9H	CAN1 message configuration register 13	C1MCONF13		–	√	–	Undefined
000F04DAH	CAN1 message ID register 13 L	C1MIDL13		–	–	√	Undefined
000F04DCH	CAN1 message ID register 13 H	C1MIDH13		–	–	√	Undefined
000F04DEH	CAN1 message control register 13	C1MCTRL13		–	–	√	00x00000 000xx000B
000F04E0H	CAN1 message data byte 01 register 14	C1MDB0114		–	–	√	Undefined
000F04E0H	CAN1 message data byte 0 register 14	C1MDB014		–	√	–	Undefined
000F04E1H	CAN1 message data byte 1 register 14	C1MDB114		–	√	–	Undefined
000F04E2H	CAN1 message data byte 23 register 14	C1MDB2314		–	–	√	Undefined
000F04E2H	CAN1 message data byte 2 register 14	C1MDB214		–	√	–	Undefined
000F04E3H	CAN1 message data byte 3 register 14	C1MDB314		–	√	–	Undefined
000F04E4H	CAN1 message data byte 45 register 14	C1MDB4514		–	–	√	Undefined
000F04E4H	CAN1 message data byte 4 register 14	C1MDB414		–	√	–	Undefined
000F04E5H	CAN1 message data byte 5 register 14	C1MDB514		–	√	–	Undefined
000F04E6H	CAN1 message data byte 67 register 14	C1MDB6714		–	–	√	Undefined
000F04E6H	CAN1 message data byte 6 register 14	C1MDB614		–	√	–	Undefined
000F04E7H	CAN1 message data byte 7 register 14	C1MDB714		–	√	–	Undefined
000F04E8H	CAN1 message data length register 14	C1MDLC14		–	√	–	0000xxxxB
000F04E9H	CAN1 message configuration register 14	C1MCONF14		–	√	–	Undefined
000F04EAH	CAN1 message ID register 14 L	C1MIDL14		–	–	√	Undefined
000F04ECH	CAN1 message ID register 14 H	C1MIDH14		–	–	√	Undefined
000F04EEH	CAN1 message control register 14	C1MCTRL14		–	–	√	00x00000 000xx000B

Table 14-16. Register Access Types (18/18)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
000F04F0H	CAN1 message data byte 01 register 15	C1MDB0115	R/W	–	–	√	Undefined
000F04F0H	CAN1 message data byte 0 register 15	C1MDB015		–	√	–	Undefined
000F04F1H	CAN1 message data byte 1 register 15	C1MDB115		–	√	–	Undefined
000F04F2H	CAN1 message data byte 23 register 15	C1MDB2315		–	–	√	Undefined
000F04F2H	CAN1 message data byte 2 register 15	C1MDB215		–	√	–	Undefined
000F04F3H	CAN1 message data byte 3 register 15	C1MDB315		–	√	–	Undefined
000F04F4H	CAN1 message data byte 45 register 15	C1MDB4515		–	–	√	Undefined
000F04F4H	CAN1 message data byte 4 register 15	C1MDB415			√	–	Undefined
000F04F5H	CAN1 message data byte 5 register 15	C1MDB515		–	√	–	Undefined
000F04F6H	CAN1 message data byte 67 register 15	C1MDB6715		–	–	√	Undefined
000F04F6H	CAN1 message data byte 6 register 15	C1MDB615		–	√	–	Undefined
000F04F7H	CAN1 message data byte 7 register 15	C1MDB715		–	√	–	Undefined
000F04F8H	CAN1 message data length register 15	C1MDLC15		–	√	–	0000xxxxB
000F04F9H	CAN1 message configuration register 15	C1MCONF15		–	√	–	Undefined
000F04FAH	CAN1 message ID register 15 L	C1MIDL15		–	–	√	Undefined
000F04FCH	CAN1 message ID register 15 H	C1MIDH15		–	–	√	Undefined
000F04FEH	CAN1 message control register 15	C1MCTRL15		–	–	√	00x00000 000xx000B

14.5.3 Register bit configuration

Table 14-17. Bit Configuration of CAN Global 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
000F05C0H	C0GMCTRL(W)	0	0	0	0	0	0	0	Clear GOM
000F05C1H		0	0	0	0	0	0	Set EFSD	Set GOM
000F05C0H	C0GMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
000F05C1H		MBON	0	0	0	0	0	0	0
000F05C6H	C0GMABT(W)	0	0	0	0	0	0	0	Clear ABTTRG
000F05C7H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
000F05C6H	C0GMABT(R)	0	0	0	0	0	0	ABTCLR	ABTTRG
000F05C7H		0	0	0	0	0	0	0	0
000F05C8H	C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0
000F05CEH	C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

Table 14-17. Bit Configuration of CAN Global 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
000F0340H	C1GMCTRL(W)	0	0	0	0	0	0	0	Clear GOM
000F0341H		0	0	0	0	0	0	Set EFSD	Set GOM
000F0340H	C1GMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
000F0341H		MBON	0	0	0	0	0	0	0
000F0346H	C1GMABT(W)	0	0	0	0	0	0	0	Clear ABTTRG
000F0347H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
000F0346H	C1GMABT(R)	0	0	0	0	0	0	ABTCLR	ABTTRG
000F0347H		0	0	0	0	0	0	0	0
000F0348H	C1GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0
000F0342H	C1GMCS	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 14-18. Bit Configuration of CAN Module Registers (1/4)

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	C0MASK1L	CM1ID [7:0]							
000F05D1H		CM1ID [15:8]							
000F05D2H	C0MASK1H	CM1ID [23:16]							
000F05D3H		0	0	0	CM1ID [28:24]				
000F05D4H	C0MASK2L	CM2ID [7:0]							
000F05D5H		CM2ID [15:8]							
000F05D6H	C0MASK2H	CM2ID [23:16]							
000F05D7H		0	0	0	CM2ID [28:24]				
000F05D8H	C0MASK3L	CM3ID [7:0]							
000F05D9H		CM3ID [15:8]							
000F05DAH	C0MASK3H	CM3ID [23:16]							
000F05DBH		0	0	0	CM3ID [28:24]				
000F05DCH	C0MASK4L	CM4ID [7:0]							
000F05DDH		CM4ID [15:8]							
000F05DEH	C0MASK4H	CM4ID [23:16]							
000F05DFH		0	0	0	CM4ID [28:24]				
000F05E0H	C0CTRL(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	C0CTRL(R)	CCERC	AL	VALID	PSMODE 1	PSMODE 0	OPMODE 2	OPMODE 1	OPMODE 0
000F05E1H		0	0	0	0	0	0	RSTAT	TSTAT
000F05E2H	C0LEC(W)	0	0	0	0	0	0	0	0
000F05E2H	C0LEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
000F05E3H	C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
000F05E4H	C0ERC	TEC [7:0]							
000F05E5H		REPS	REC [7:0]						
000F05E6H	C0IE(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	C0IE(R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
000F05E7H		0	0	0	0	0	0	0	0
000F05E8H	C0INTS(W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear 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 14-18. Bit Configuration of CAN Module Registers (2/4)

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
000F05E8H	C0INTS(R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
000F05E9H		0	0	0	0	0	0	0	0
000F05EAH	C0BRP	TQPRS [7:0]							
000F05ECH	C0BTR	0	0	0	0	TSEG1 [3:0]			
000F05EDH		0	0	SJW [1:0]		0	TSEG2 [2:0]		
000F05EEH	C0LIPT	LIPT [7:0]							
000F05F0H	C0RGPT(W)	0	0	0	0	0	0	0	Clear ROVF
000F05F1H		0	0	0	0	0	0	0	0
000F05F0H	C0RGPT(R)	0	0	0	0	0	0	RHPM	ROVF
000F05F1H		RGPT [7:0]							
000F05F2H	C0LOPT	LOPT [7:0]							
000F05F4H	C0TGPT(W)	0	0	0	0	0	0	0	Clear TOVF
000F05F5H		0	0	0	0	0	0	0	0
000F05F4H	C0TGPT(R)	0	0	0	0	0	0	THPM	TOVF
000F05F5H		TGPT [7:0]							
000F05F6H	C0TS(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	C0TS(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 14-18. Bit Configuration of CAN Module Registers (3/4)

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
000F0380H	C1MASK1L	CM1ID [7:0]							
000F0381H		CM1ID [15:8]							
000F0382H	C1MASK1H	CM1ID [23:16]							
000F0383H		0	0	0	CM1ID [28:24]				
000F0384H	C1MASK2L	CM2ID [7:0]							
000F0385H		CM2ID [15:8]							
000F0386H	C1MASK2H	CM2ID [23:16]							
000F0387H		0	0	0	CM2ID [28:24]				
000F0388H	C1MASK3L	CM3ID [7:0]							
000F0389H		CM3ID [15:8]							
000F038AH	C1MASK3H	CM3ID [23:16]							
000F038BH		0	0	0	CM3ID [28:24]				
000F038CH	C1MASK4L	CM4ID [7:0]							
000F038DH		CM4ID [15:8]							
000F038EH	C1MASK4H	CM4ID [23:16]							
000F038FH		0	0	0	CM4ID [28:24]				
000F0390H	C1CTRL(W)	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE 1	Clear PSMODE 0	Clear OPMODE 2	Clear OPMODE 1	Clear OPMODE 0
000F0391H		Set CCERC	Set AL	0	Set PSMODE 1	Set PSMODE 0	Set OPMODE 2	Set OPMODE 1	Set OPMODE 0
000F0390H	C1CTRL(R)	CCERC	AL	VALID	PSMODE 1	PSMODE 0	OPMODE 2	OPMODE 1	OPMODE 0
000F0391H		0	0	0	0	0	0	RSTAT	TSTAT
000F0392H	C1LEC(W)	0	0	0	0	0	0	0	0
000F0392H	C1LEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
000F0393H	C1INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
000F0394H	C1ERC	TEC [7:0]							
000F0395H		REPS	REC [7:0]						
000F0396H	C1IE(W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
000F0397H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
000F0396H	C1IE(R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
000F0397H		0	0	0	0	0	0	0	0
000F0398H	C1INTS(W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
000F0399H		0	0	0	0	0	0	0	0
000F0398H	C1INTS(R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
000F0399H		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 14-18. Bit Configuration of CAN Module Registers (4/4)

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
000F039AH	C1BRP	TQPRS [7:0]							
000F039CH	C1BTR	0	0	0	0	TSEG1 [3:0]			
000F039DH		0	0	SJW [1:0]		0	TSEG2 [2:0]		
000F039EH	C1LIPT	LIPT [7:0]							
000F03A0H	C1RGPT(W)	0	0	0	0	0	0	0	Clear ROVF
000F03A1H		0	0	0	0	0	0	0	0
000F03A0H	C1RGPT(R)	0	0	0	0	0	0	RHPM	ROVF
000F03A1H		RGPT [7:0]							
000F03A2H	C1LOPT	LOPT [7:0]							
000F03A4H	C1TGPT(W)	0	0	0	0	0	0	0	Clear TOVF
000F03A5H		0	0	0	0	0	0	0	0
000F03A4H	C1TGPT(R)	0	0	0	0	0	0	THPM	TOVF
000F03A5H		TGPT [7:0]							
000F03A6H	C1TS(W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
000F03A7H		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
000F03A6H	C1TS(R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
000F03A7H		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 14-19. Bit Configuration of Message Buffer 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
000F06x0H	C0MDB01m	Message data (byte 0)							
000F06x1H		Message data (byte 1)							
000F06x0H	C0MDB0m	Message data (byte 0)							
000F06x1H	C0MDB1m	Message data (byte 1)							
000F06x2H	C0MDB23m	Message data (byte 2)							
000F06x3H		Message data (byte 3)							
000F06x2H	C0MDB2m	Message data (byte 2)							
000F06x3H	C0MDB3m	Message data (byte 3)							
000F06x4H	C0MDB45m	Message data (byte 4)							
000F06x5H		Message data (byte 5)							
000F06x4H	C0MDB4m	Message data (byte 4)							
000F06x5H	C0MDB5m	Message data (byte 5)							
000F06x6H	C0MDB67m	Message data (byte 6)							
000F06x7H		Message data (byte 7)							
000F06x6H	C0MDB6m	Message data (byte 6)							
000F06x7H	C0MDB7m	Message data (byte 7)							
000F06x8H	C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
000F06x9H	C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
000F06xAH	C0MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
000F06xBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
000F06xCH	C0MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
000F06xDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
000F06xEH	C0MCTRLm (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	C0MCTRLm (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
3. x = 0 to F

Table 14-19. Bit Configuration of Message Buffer 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
000F04x0H	C1MDB01m	Message data (byte 0)							
000F04x1H		Message data (byte 1)							
000F04x0H	C1MDB0m	Message data (byte 0)							
000F04x1H	C1MDB1m	Message data (byte 1)							
000F04x2H	C1MDB23m	Message data (byte 2)							
000F04x3H		Message data (byte 3)							
000F04x2H	C1MDB2m	Message data (byte 2)							
000F04x3H	C1MDB3m	Message data (byte 3)							
000F04x4H	C1MDB45m	Message data (byte 4)							
000F04x5H		Message data (byte 5)							
000F04x4H	C1MDB4m	Message data (byte 4)							
000F04x5H	C1MDB5m	Message data (byte 5)							
000F04x6H	C1MDB67m	Message data (byte 6)							
000F04x7H		Message data (byte 7)							
000F04x6H	C1MDB6m	Message data (byte 6)							
000F04x7H	C1MDB7m	Message data (byte 7)							
000F04x8H	C1MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
000F04x9H	C1MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
000F04xAH	C1MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
000F04xBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
000F04xCH	C1MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
000F04xDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
000F04xEH	C1MCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
000F04xFH		0	0	0	0	Set IE	0	Set TRQ	Set RDY
000F04xEH	C1MCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
000F04xFH		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
3. x = 0 to F

14.6 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 (C0GMCTRL, C1GMCTRL)
- CAN global automatic block transmission control register (C0GMABT, C1GMABT)
- CAN module control register (C0CTRL, C1CTRL)
- CAN module interrupt enable register (C0IE, C1IE)
- CAN module interrupt status register (C0INTS, C1INTS)
- CAN module receive history list register (C0RGPT, C1RGPT)
- CAN module transmit history list register (C0TGPT, C1TGPT)
- CAN module time stamp register (C0TS, C1TS)
- CAN message control register (C0MCTRLm, C1MCTRLm)

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 14-23 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 (refer to the 16-bit data after a write operation in **Figure 14-24**). **Figure 14-23** shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 14-23. Example of Bit Setting/Clearing Operations

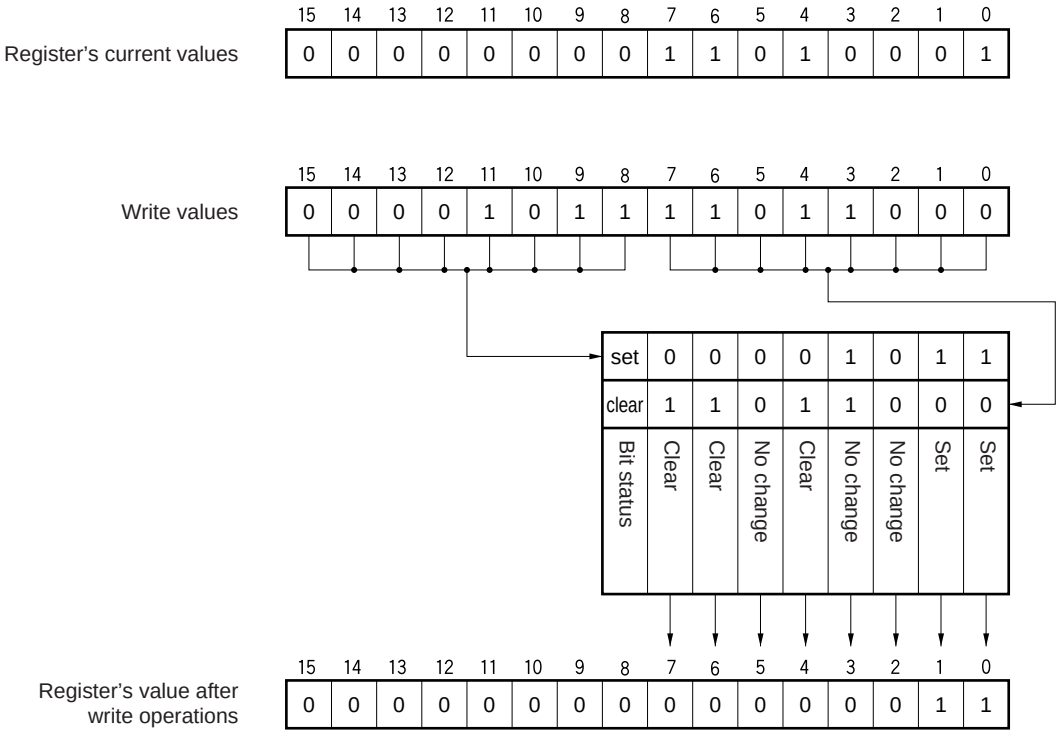


Figure 14-24. 16-Bit Data during Write Operation

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

14.7 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 14-25. 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	CAN MCKE1	CAN MCK1	CAN MCKE0	CAN MCK0	0	0	SGCLK SEL

CANMCKEn	CANMCKn	aFCANn input clock control
0	X	STOPS input clock supply. Writing to SFR to be used with aFCANn is disabled
1	0	f_{MAIN} is supplied Reading from and writing to SFR to be used with aFCANn is enable
1	1	f_{MP} is supplied Reading from and writing to SFR to be used with aFCANn is enable

n = 0, 1

(2) CAN global module control register (C0GMCTRL, C1GMCTRL)

The C0GMCTRL, C1GMCTRL register is used to control the operation of the CAN module.

Figure 14-26. Format of CAN Global Module Control Register (C0GMCTRL, C1GMCTRL) (1/2)

Address: F05C0H (C0GMCTRL), F0340H (C1GMCTRL) After reset: 0000H R/W

(a) Read

C0GMCTRL, C1GMCTRL

15	14	13	12	11	10	9	8
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

C0GMCTRL, C1GMCTRL

15	14	13	12	11	10	9	8
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 (C0MDB0m, C1MDB0m, C0MDB1m, C1MDB1m, C0MDB01m, C1MDB01m, C0MDB2m, C1MDB2m, C0MDB3m, C1MDB3m, C0MDB23m, C1MDB23m, C0MDB4m, C1MDB4m, C0MDB5m, C1MDB5m, C0MDB45m, C1MDB45m, C0MDB6m, C1MDB6m, C0MDB7m, C1MDB7m, C0MDB67m, C1MDB67m, C0MDLCm, C1MDLCm, C0MCONFm, C1MCONFm, C0MIDLm, C1MIDLm, C0MIDHm, C1MIDHm, and C0MCTRLm, C1MCTRLm), or registers related to transmit history or receive history (C0LOPT, C1LOPT, C0TGPT, C1TGPT, C0LIPT, C1LIPT, and C0RGPT, C1RGPT) 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 14-26. Format of CAN Global Module Control Register (C0GMCTRL, C1GMCTRL) (2/2)

EFSD	Bit Enabling Forced Shut Down
0	Forced shut down by GOM = 0 disabled.
1	Forced shut down by GOM = 0 enabled.

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. If access to another register (including reading the C0GMCTRL, C1GMCTRL register) is 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.

When DMA is being performed, a request for a forced shut down might be ignored. Be sure to read the EFSD bit and confirm that forced shut down is enabled before issuing a forced shut down request. If forced shut down cannot be enabled because DMA is being performed, it is recommended to temporarily stop DMA.

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 the above		No change in GOM bit.

Caution Set GOM bit and EFSD bit always separately.

(3) CAN global module clock select register (C0GMCS, C1GMCS)

The C0GMCS, C1GMCS register is used to select the CAN module system clock.

Figure 14-27. Format of CAN Global Module Clock Select Register (C0GMCS, C1GMCS)

Address: F05CEH (C0GMCS), F0342H (C1GMCS) After reset: 0FH R/W

	7	6	5	4	3	2	1	0
C0GMCS, C1GMCS	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 (C0GMABT, C1GMABT)

The C0GMABT, C1GMABT register is used to control the automatic block transmission (ABT) operation.

Figure 14-28. Format of CAN Global Automatic Block Transmission Control Register (C0GMABT, C1GMABT) (1/2)

Address: F05C6H (C0GMABT), F0346H (C1GMABT) After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
C0GMABT, C1GMABT	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
C0GMABT, C1GMABT	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 C0GMABT, C1GMABT register to the default value (0000H) and confirm the C0GMABT, C1GMABT 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 14-28. Format of CAN Global Automatic Block Transmission Control Register (C0GMABT, C1GMABT) (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 C0CTRL.TSTAT.TSTAT, C1CTRL.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 the 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 (C0GMABTD, C1GMABTD)

The C0GMABTD, C1GMABTD 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 14-29. Format of CAN Global Automatic Block Transmission Delay Setting Register (C0GMABTD, C1GMABTD)

Address: F05C8H (C0GMABTD), F0348H (C1GMABTD) After reset: 00H R/W

	7	6	5	4	3	2	1	0
C0GMABTD, C1GMABTD	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 the above				Setting prohibited

Cautions 1. Do not change the contents of the C0GMABTD, C1GMABTD 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 (C0MASKaL, C1MASKaL, C0MASKaH, C1MASKaH)
(a = 1, 2, 3, or 4)

The C0MASKaL, C1MASKaL and C0MASKaH, C1MASKaH 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 14-30. Format of CAN Module Mask Register
(C0MASKaL, C1MASKaL, C0MASKaH, C1MASKaH) (a = 1, 2, 3, or 4) (1/2)

- CAN Module Mask 1 Register
(C0MASK1L, C1MASK1L, C0MASK1H, C1MASK1H)

Address: F05D0H (C0MASK1L), F05D2H (C0MASK1H) After reset: Undefined R/W
Address: F0380H (C1MASK1L), F0382H (C1MASK1H) After reset: Undefined R/W

C0MASK1L, C1MASK1L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

C0MASK1H/ MASK1H, C1MASK1H

15	14	13	12	11	10	9	8
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
(C0MASK2L, C1MASK2L, C0MASK2H, C1MASK2H)

Address: F05D4H (C0MASK2L), F05D6H (C0MASK2H) After reset: Undefined R/W
Address: F0384H (C1MASK2L), F0386H (C1MASK2H) After reset: Undefined R/W

C0MASK2L, C1MASK2L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

C0MASK2H, C1MASK2H

15	14	13	12	11	10	9	8
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 14-30. Format of CAN Module Mask Register (C0MASKaL, C1MASKaL, C0MASKaH, C1MASKaH) (a = 1, 2, 3, or 4) (2/2)

- CAN Module Mask 3 Register
  (C0MASK3L, C1MASK3L, C0MASK3H, C1MASK3H)

Address: F05D8H (C0MASK3L), F05DAH (C0MASK3H) After reset: Undefined R/W

Address: F0388H (C1MASK3L), F038AH (C1MASK3H) After reset: Undefined R/W

C0MASK3L, C1MASK3L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

C0MASK3H, C1MASK3H

15	14	13	12	11	10	9	8
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
  (C0MASK4L, C1MASK4L, C0MASK4H, C1MASK4H)

Address: F05DCH (C0MASK4L), F05DEH (C0MASK4H) After reset: Undefined R/W

Address: F038CH (C1MASK4L), F038EH (C1MASK4H) After reset: Undefined R/W

C0MASK4L, C1MASK4L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

C0MASK4H, C1MASK4H

15	14	13	12	11	10	9	8
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-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 (C0CTRL, C1CTRL)

The C0CTRL, C1CTRL register is used to control the operation mode of the CAN module.

Figure 14-31. Format of CAN Module Control Register (C0CTRL, C1CTRL) (1/4)

Address: F05E0H (C0CTRL), F0390H (C1CTRL) After reset: 0000H R/W

(a) Read

C0CTRL, C1CTRL	15	14	13	12	11	10	9	8
	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

C0CTRL, C1CTRL	15	14	13	12	11	10	9	8
	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 14-31. Format of CAN Module Control Register (C0CTRL, C1CTRL) (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 C0ERC, C1ERC and C0INFO, C1INFO registers are not cleared in the initialization mode.
1	The C0ERC, C1ERC and C0INFO, C1INFO registers are cleared in the initialization mode.

Remarks 1. The CCERC bit is used to clear the C0ERC, C1ERC and C0INFO, C1INFO 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 C0ERC, C1ERC and C0INFO, C1INFO 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 14-31. Format of CAN Module Control Register (C0CTRL, C1CTRL) (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 C0GMCTRL, C1GMCTRL 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.

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 the above			Setting prohibited

Caution 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.

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.
0	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 the above		AL bit is not changed.

Figure 14-31. Format of CAN Module Control Register (C0CTRL, C1CTRL) (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 the 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 the 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 the 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 the 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 the above		OPMODE2 bit is not changed.

(8) CAN module last error code register (C0LEC, C1LEC)

The C0LEC, C1LEC register provides the error information of the CAN protocol.

Figure 14-32. Format of CAN Module Last Error Code Register (C0LEC, C1LEC)

Address: F05E2H (C0LEC), F0392H (C1LEC) After reset: 00H R/W

	7	6	5	4	3	2	1	0
C0LEC, C1LEC	0	0	0	0	0	LEC2	LEC1	LEC0

Remarks 1. The contents of the C0LEC, C1LEC 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 C0LEC, C1LEC 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 (C0INFO, C1INFO)

The C0INFO, C1INFO register indicates the status of the CAN module.

Figure 14-33. Format of CAN Module Information Register (C0INFO, C1INFO)

Address: F05E3H (C0INFO), F0393H (C1INFO) After reset: 00H R

	7	6	5	4	3	2	1	0
C0INFO, C1INFO	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 (C0ERC, C1ERC)

The C0ERC, C1ERC register indicates the count value of the transmission/reception error counter.

Figure 14-34. Format of CAN Module Error Counter Register (C0ERC, C1ERC)

Address: F05E4H (C0ERC), F0394H (C1ERC) After reset: 0000H R

C0ERC, C1ERC

15	14	13	12	11	10	9	8
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-REC0	Reception Error Counter Bit
0-127	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-TEC0	Transmission Error Counter Bit
0-255	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 (C0IE, C1IE)

The C0IE, C1IE register is used to enable or disable the interrupts of the CAN module.

Figure 14-35. Format of CAN Module Interrupt Enable Register (C0IE, C1IE) (1/2)

Address: F05E6H (C0IE), F0396H (C1IE) After reset: 0000H R/W

(a) Read

C0IE, C1IE	15	14	13	12	11	10	9	8
	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

C0IE, C1IE	15	14	13	12	11	10	9	8
	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-CIE0	CAN Module Interrupt Enable Bit
0	Output of the interrupt corresponding to interrupt status register C0INTS, C1INTS [5:0] bits is disabled.
1	Output of the interrupt corresponding to interrupt status register C0INTS, C1INTS [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 the 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 the above		CIE4 bit is not changed.

Figure 14-35. Format of CAN Module Interrupt Enable Register (C0IE, C1IE) (2/2)

Set CIE3	Clear CIE3	Setting of CIE3 Bit
0	1	CIE3 bit is cleared to 0.
1	0	CIE3 bit is set to 1.
Other than the 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 the 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 the 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 the above		CIE0 bit is not changed.

(12) CAN module interrupt status register (C0INTS, C1INTS)

The C0INTS, C1INTS register indicates the interrupt status of the CAN module.

Figure 14-36. Format of CAN Module Interrupt Status Register (C0INTS, C1INTS)

Address: F05E8H (C0INTS), F0398H (C1INTS) After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
C0INTS, C1INTS	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
C0INTS, C1INTS	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-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 (C0BRP, C1BRP)

The C0BRP, C1BRP register is used to select the CAN protocol layer basic clock (f_{TQ}). The communication baud rate is set to the C0BTR, C1BTR register.

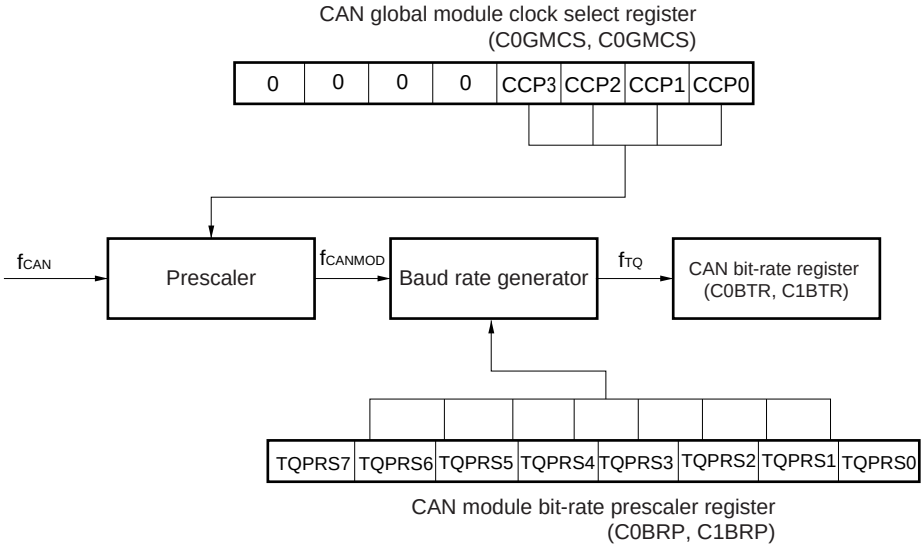
Figure 14-37. Format of CAN Module Bit Rate Prescaler Register (C0BRP, C1BRP)

Address: F05EAH (C0BRP), F039AH (C1BRP) After reset: FFH R/W

	7	6	5	4	3	2	1	0
C0BRP, C1BRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7-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 14-38. CAN Global Clock



Caution The CPBRP 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 (C0BTR, C1BTR)

The C0BTR, C1BTR register is used to control the data bit time of the communication baud rate.

Figure 14-39. Format of CAN Module Bit Rate Register (C0BTR, C1BTR) (1/2)

Address: F05ECH (C0BTR), F039CH (C1BTR) After reset: 370FH R/W

C0BTR, C1BTR

15	14	13	12	11	10	9	8
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)

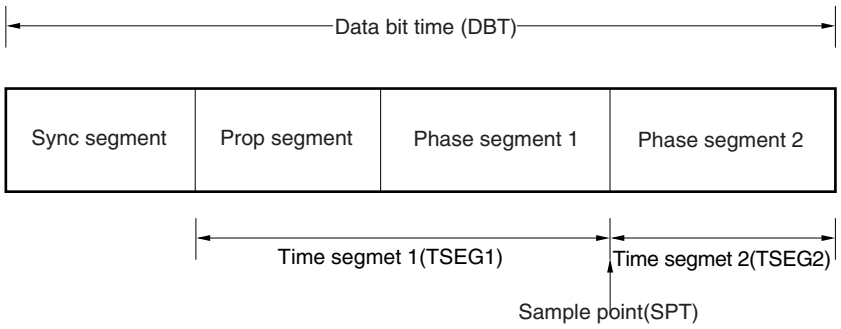
Figure 14-39. Format of CAN Module Bit Rate Register (C0BTR, C1BTR) (2/2)

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 C0BRP, C1BRP register = 00H.

Remark TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)

Figure 14-40. Data Bit Time



(15) CAN module last in-pointer register (C0LIPT, C1LIPT)

The C0LIPT, C1LIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

Figure 14-41. Format of CAN Module Last In-pointer Register (C0LIPT, C1LIPT)

Address: F05EEH (C0LIPT), F039EH (C1LIPT) After reset: Undefined R

	7	6	5	4	3	2	1	0
C0LIPT, C1LIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7-LIPT0	Last In-Pointer Register (C0LIPT, C1LIPT)
0 to 15	When the C0LIPT, C1LIPT 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 C0LIPT, C1LIPT 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 C0RGPT, C1RGPT 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 C0LIPT, C1LIPT register is undefined.

(16) CAN module receive history list register (C0RGPT, C1RGPT)

The C0RGPT, C1RGPT register is used to read the receive history list.

Figure 14-42. Format of CAN Module Receive History List Register (C0RGPT, C1RGPT) (1/2)

Address: F05F0H (C0RGPT), F03A0H (C1RGPT) After reset: xx02H R/W

(a) Read

C0RGPT, C1RGPT

15	14	13	12	11	10	9	8
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

C0RGPT, C1RGPT

15	14	13	12	11	10	9	8
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-RGPT0	Receive History List Get Pointer
0 to 15	When the C0RGPT, C1RGPT 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 C0RGPT, C1RGPT). 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 C0RGPT, C1RGPT are read by software.

Figure 14-42. Format of CAN Module Receive History List Register (C0RGPT, C1RGPT) (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 (C0LOPT, C1LOPT)

The C0LOPT, C1LOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

Figure 14-43. Format of CAN Module Last Out-pointer Register (C0LOPT, C1LOPT)

Address: F05F2H (C0L0PT), F03A2H (C1LOPT) After reset: Undefined R

	7	6	5	4	3	2	1	0
C0LOPT, C1LOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7-LOPT0	Last Out-Pointer of Transmit History List (LOPT)
0 to 15	When the C0LOPT, C1LOPT 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 C0LOPT, C1LOPT 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 C0LOPT, C1LOPT register is undefined.

(18) CAN module transmit history list register (C0TGPT, C1TGPT)

The C0TGPT, C1TGPT register is used to read the transmit history list.

Figure 14-44. Format of CAN Module Transmit History List Register (C0TGPT, C1TGPT) (1/2)

Address: F05F4H (C0TGPT), F03A4H (C1TGPT) After reset: xx02H R/W

(a) Read

	15	14	13	12	11	10	9	8
C0TGPT, C1TGPT	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
C0TGPT, C1TGPT	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-TGPT0	Transmit History List Read Pointer
0 to 15	When the C0TGPT, C1TGPT 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 C0TGPT, C1TGPT). 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 14-44. Format of CAN Module Transmit History List Register (C0TGPT, C1TGPT) (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 (C0TS, C1TS)

The C0TS, C1TS register is used to control the time stamp function.

Figure 14-45. Format of CAN Module Time Stamp Register (C0TS, C1TS) (1/2)

Address: F05F6H (C0TS), F03A6H (C1TS) After reset: 0000H R/W

(a) Read

	15	14	13	12	11	10	9	8
C0TS, C1TS	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
C0TS, C1TS	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 14-45. Format of CAN Module Time Stamp Register (C0TS, C1TS) (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. Refer to **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 the 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 the 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 the above		TSEN bit is not changed.

(20) CAN message data byte register (C0MDBxm, C1MDBxm) (x = 0 to 7), (C1MDBzm) (z = 01, 23, 45, 67)

The C0MDBxm, C1MDBxm, C1MDBzm registers are used to store the data of a transmit/receive message. The C0MDBxm, C1MDBxm registers can access in 8-bit units. The C1MDBzm registers can access the C0MDBxm, C1MDBxm registers in 16-bit units.

Figure 14-46. Format of CAN Message Data Byte Register (C0MDBxm, C1MDBxm) (x = 0 to 7), (C1MDBzm) (z = 01, 23, 45, 67) (1/2)

Address: See **Table 14-16** After reset: Undefined R/W

- C0MDBxm, C1MDBxm Register

	7	6	5	4	3	2	1	0
C0MDB0m, C1MDB0m	MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00
C0MDB1m, C1MDB1m	MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10
C0MDB2m, C1MDB2m	MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20
C0MDB3m, C1MDB3m	MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30
C0MDB4m, C1MDB4m	MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40
C0MDB5m, C1MDB5m	MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50
C0MDB6m, C1MDB6m	MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60
C0MDB7m, C1MDB7m	MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

Remark m = 0 to 15

Figure 14-46. Format of CAN Message Data Byte Register (C0MDBxm, C1MDBxm) (x = 0 to 7), (C1MDBzm) (z = 01, 23, 45, 67) (2/2)

- C1MDBzm Register

C0MDB01m, C1MDB01m

15	14	13	12	11	10	9	8
MDATA 0115	MDATA 0114	MDATA 0113	MDATA 0112	MDATA 0111	MDATA 0110	MDATA 019	MDATA 018
7	6	5	4	3	2	1	0
MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

C0MDB23m, C1MDB23m

15	14	13	12	11	10	9	8
MDATA 2315	MDATA 2314	MDATA 2313	MDATA 2312	MDATA 2311	MDATA 2310	MDATA 239	MDATA 238
7	6	5	4	3	2	1	0
MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

C0MDB45m, C1MDB45m

15	14	13	12	11	10	9	8
MDATA 4515	MDATA 4514	MDATA 4513	MDATA 4512	MDATA 4511	MDATA 4510	MDATA 459	MDATA 458
7	6	5	4	3	2	1	0
MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

C0MDB67m, C1MDB67m

15	14	13	12	11	10	9	8
MDATA 6715	MDATA 6714	MDATA 6713	MDATA 6712	MDATA 6711	MDATA 6710	MDATA 679	MDATA 678
7	6	5	4	3	2	1	0
MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

Remark m = 0 to 15

(21) CAN message data length register m (C0MDLCm, C1MDLCm)

The C0MDLCm, C1MDLCm register is used to set the number of bytes of the data field of a message buffer.

Figure 14-47. Format of CAN Message Data Length Register m (C0MDLCm, C1MDLCm)

Address: See **Table 14-16** After reset: 0000xxxxB R/W

	7	6	5	4	3	2	1	0
C0MDLCm, C1MDLCm	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 C0MDBxm, C1MDBxm as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. C0MDBxm, C1MDBxm in which no data is stored is undefined.

Remark m = 0 to 15

(22) CAN message configuration register (C0MCONFm, C1MCONFm)

The C0MCONFm, C1MCONFm register is used to specify the type of the message buffer and to set a mask.

Figure 14-48. Format of CAN Message Configuration Register (C0MCONFm, C1MCONFm) (1/2)

Address: See **Table 14-16** After reset: Undefined R/W

	7	6	5	4	3	2	1	0
C0MCONFm, C1MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0

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 the above			Setting prohibited

Remark m = 0 to 15

Figure 14-48. Format of CAN Message Configuration Register (C0MCONFm, C1MCONFm) (2/2)

MA0	Message Buffer Assignment Bit
0	Message buffer not used.
1	Message buffer used.

Caution Be sure to write 0 to bits 2 and 1.

Remark m = 0 to 15

(23) CAN message ID register m (C0MIDLm, C1MIDLm and C0MIDHm, C1MIDHm)

The C0MIDLm, C1MIDLm and C0MIDHm, C1MIDHm registers are used to set an identifier (ID).

Figure 14-49. Format of CAN Message ID Register m (C0MIDLm, C1MIDLm and C0MIDHm, C1MIDHm)

Address: See **Table 14-16** After reset: Undefined R/W

C0MIDLm, C1MIDLm

15	14	13	12	11	10	9	8
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
7	6	5	4	3	2	1	0
ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

C0MIDHm, C1MIDHm

15	14	13	12	11	10	9	8
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 C0MIDHm, C1MIDHm 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.

Remark m = 0 to 15

(24) CAN message control register m (C0MCTRLm, C1MCTRLm)

The C0MCTRLm, C1MCTRLm register is used to control the operation of the message buffer.

Figure 14-50. Format of CAN Message Control Register m (C0MCTRLm, C1MCTRLm) (1/3)

Address: See **Table 14-16**. After reset: 00x00000 R/W

000xx000B

(a) Read

C0MCTRLm, C1MCTRLm	15	14	13	12	11	10	9	8
	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

C0MCTRLm, C1MCTRLm	15	14	13	12	11	10	9	8
	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.

Remark m = 0 to 15

Figure 14-50. Format of CAN Message Control Register m (C0MCTRLm, C1MCTRLm) (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 C0MCTRLm, C1MCTRLm 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 the 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.

Caution Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.

Remark m = 0 to 15

Figure 14-50. Format of CAN Message Control Register m (C0MCTRLm, C1MCTRLm) (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 the 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 the above		RDY bit is not changed.

Caution Set IE bit and RDY bit always separately.

Remark m = 0 to 15

(25) Serial communication pin select register1 (STSEL1)

The STSEL1 register is used to switch the input source to the timer array unit and the CAN communication pins.

This register can be read or written in 1-bits unit or 8-bit units.

The STSEL1 register can be used to select which set of CTxD, CRxD pins provided at two different ports to use.

Figure 14-51. Format of Serial Communication Pin Select Register1 (STSEL1)

Address: FFF3D After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STSEL1	SIIC1	SIIC0	0	0	SCAN1	SCAN0	TMCAN1	TMCAN0

SCAN0	Communication pin selection of aFCAN0	
	CTxD0	CRxD0
0	P71	P70
1	P00	P01

SCAN1	Communication pin selection of aFCAN1	
	CTxD1	CRxD1
0	P62	P63
1	P134	P135

(26) Port mode registers 0, 6, 7, 13 (PM0, PM6, PM7, PM13)

The PM0, PM6, PM7 and PM13 registers are used to set ports 0, 6, 7 and 13 to input or output in 1-bit units.

When using the P00/CTxD0 or P71/CTxD0 pins for serial data output, clear the PM00 or PM71 bits to “0”, and set the output latches of P00 or P71 to “1”. When using the P62/CTxD1 or P134/CTxD1 pins for serial data output, clear the PM62 or PM134 bits to “0”, and set the output latches of P62 or P134 to “1”.

When using the P01/CRxD0 or P70/CRxD0 pins for serial data input, set the PM01 or PM70 bits to “1”. At this time, the output latches of P01 or P70 may be “0” or “1”. When using the P63/CRxD1 or P135/CRxD1 pins for serial data input, set the PM63 or PM135 bits to “1”. At this time, the output latches of P63 or P135 may be “0” or “1”.

The PM1, PM6, PM7, PM13 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 serial data output and serial data input are provided at two ports per channel. Select either port by using the corresponding register.

Remark The pins mounted depend on the product. See **1.4 Pin Configuration** and **2.1 Pin Function List**.

Figure 14-52. Format of Port Mode Registers 0, 6, 7, 13 (PM0, PM6, PM7, PM13)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	PM66	PM65	PM64	PM63	PM62	PM61	PM60

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF2DH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	PM136	PM135	PM134	PM133	PM132	PM131	0

PMmn	PMmn pin I/O mode selection (m = 1, 6, 7, 13 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.8 CAN Controller Initialization

This chapter is explaining the case of CAN channel 0 for register symbol to the example. In the case of CAN channel 1, please transpose 0 to 1, and read it (ex.: C0GMCS → C1GMCS).

14.8.1 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 C0GMCS 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 C0GMCTRL register.

For the procedure of initializing the CAN module, refer to **14.16 Operation Of CAN Controller.**

14.8.2 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 C0MCTRLm register to 0.
- Clear the MA0 bit of the C0MCONFm register to 0.

Remark m = 0 to 15

14.8.3 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.

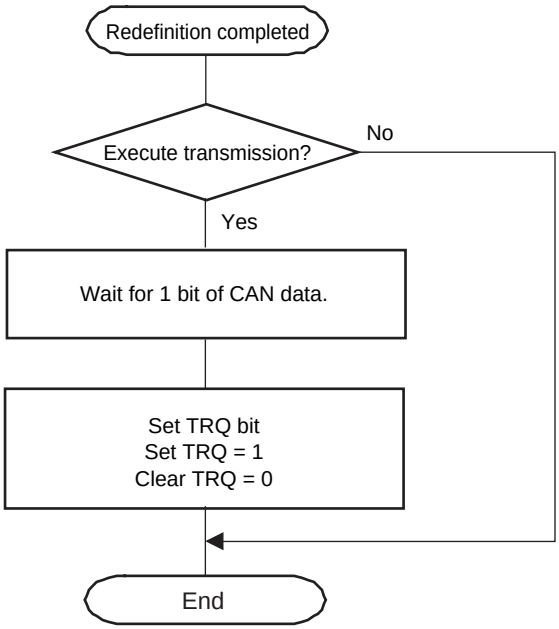
(2) To redefine message buffer during reception

Perform redefinition as shown in Figure 14-66.

(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 (refer to **14.10.4 (1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)** and **14.10.4 (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 14-53. Setting Transmission Request (TRQ) to Transmit Message Buffer After Redefining



Cautions1. 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 14-66 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.

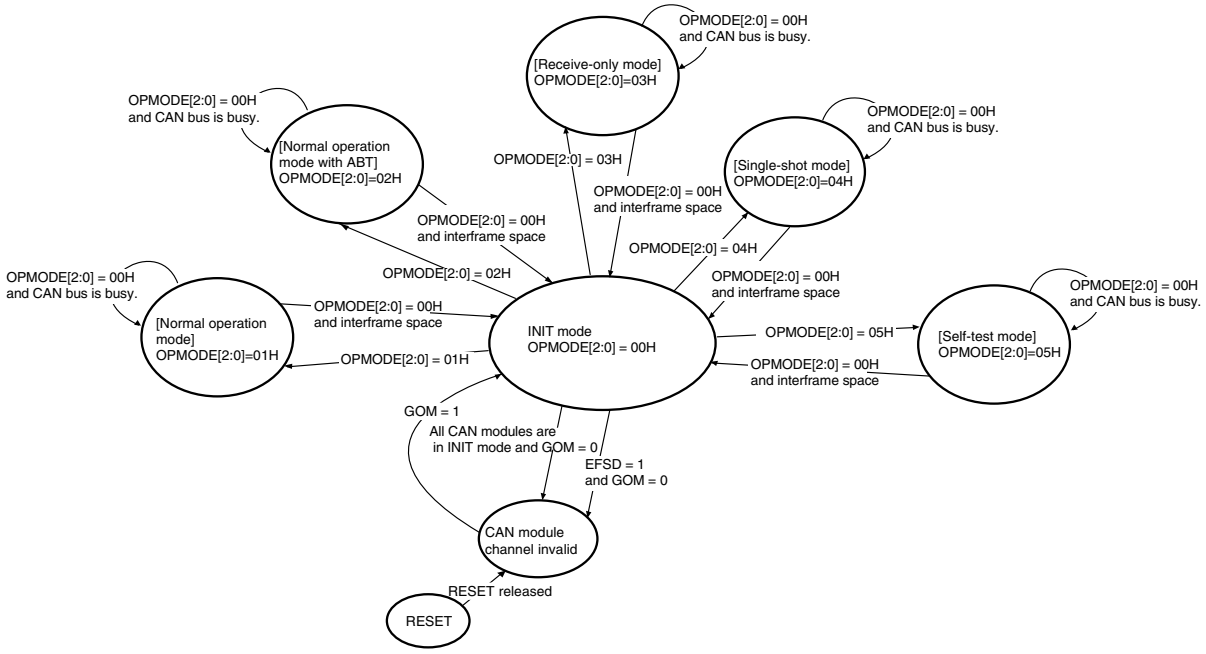
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 14-53 is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

14.8.4 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 14-54. 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 C0CTRL 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 (refer to **Figure 14-64**).

14.8.5 Resetting error counter C0ERC of CAN module

If it is necessary to reset the CAN module error counter C0ERC and the CAN module information register C0INFO when re-initialization or forced recovery from the bus-off state is made, set the CCERC bit of the C0CTRL register to 1 in the initialization mode. When this bit is set to 1, the CAN module error counter C0ERC and the CAN module information register C0INFO are cleared to their default values.

14.9 Message Reception

14.9.1 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 C0MCONFm register set to 1B.)
- Set as a receive message buffer
  (MT [2:0] bits of C0MCONFm register set to 001B, 010B, 011B, 100B, or 101B.)
- Ready for reception
  (RDY bit of C0MCTRLm 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

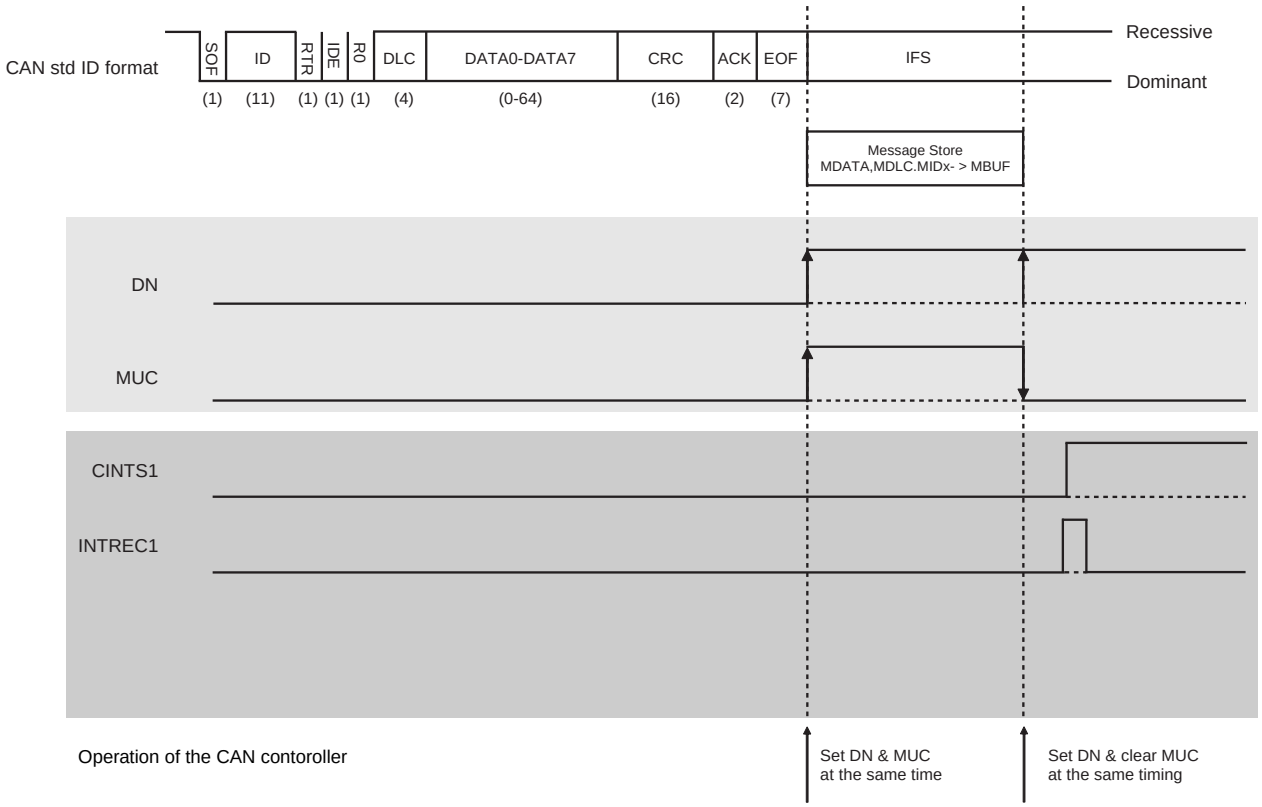
14.9.2 Receive Data Read

To keep data consistency when reading CAN message buffers, perform the data reading according to Figure 14-76 to 14-78.

During message reception, the CAN module sets DN of the C0MCTRLm 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 C0MCTRLm register of the message buffer is set. (Refer to **Figure 14-55**.)

The receive history list is also updated just before the storage process. In addition, during storage process (MUC = 1), the RDY bit of the C0MCTRLm 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.

Figure 14-55. DN and MUC Bit Setting Period (for Standard ID Format)



Remark m = 0 to 15

14.9.3 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 C0LIPT register and the receive history list get pointer (RGPT) with the corresponding C0RGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The C0LIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the C0LIPT 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 C0RGPT 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 C0RGPT register, the RGPT pointer is automatically incremented.

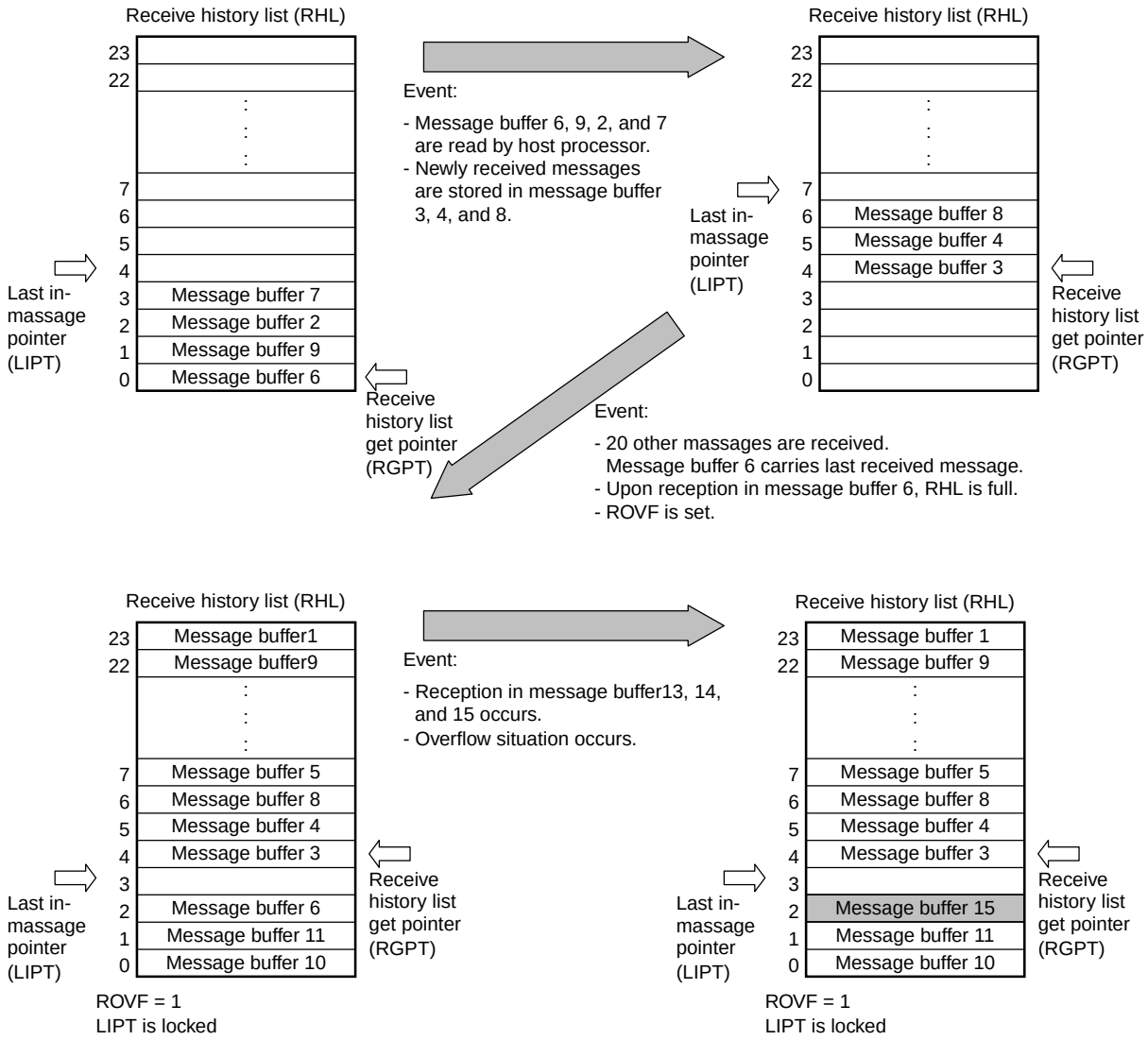
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the C0RGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer 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 C0RGPT 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 14-56. Receive History List



ROVF = 1 denotes that LIPT equals RGPT−1 while message buffer number stored to element indicated by LIPT−1.

14.9.4 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 CAN0 message ID registers L14 and H14 (C0MIDL14 and C0MIDH14))

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 C0MCONF14 register are set to 010B).

<3> Mask setting for CAN module 1 (mask 1) (Example)

(Using CAN0 module mask 1 registers L and H (C0MASK1L and C0MASK1H))

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".

14.9.5 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 C0MCTRLm 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 14.9.1 Message Reception.

Remark m = 0 to 15

14.9.6 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 C0MCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT [2:0] bits in C0MCONFm register set to 000B)
- Ready for reception
  (RDY bit of C0MCTRLm register set to 1.)
- Set to transmit message
  (RTR bit of C0MCONFm register is cleared to 0.)
- Transmission request is not set.
  (TRQ bit of C0MCTRLm 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 C0MDLCm register stores the received DLC value.
- C0MDATA0m to C0MDATA7m in the data area are not updated (data before reception is saved).
- The DN bit of the C0MCTRLm register is set to 1.
- The CINTS1 bit of the C0INTS register is set to 1 (if the IE bit in the C0MCTRLm 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 C0MCTRLm register of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the C0IE 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 C0MCONFm register of the message buffer and the DN bit of the C0MCTRLm 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

14.10 Message Transmission

14.10.1 Message transmission

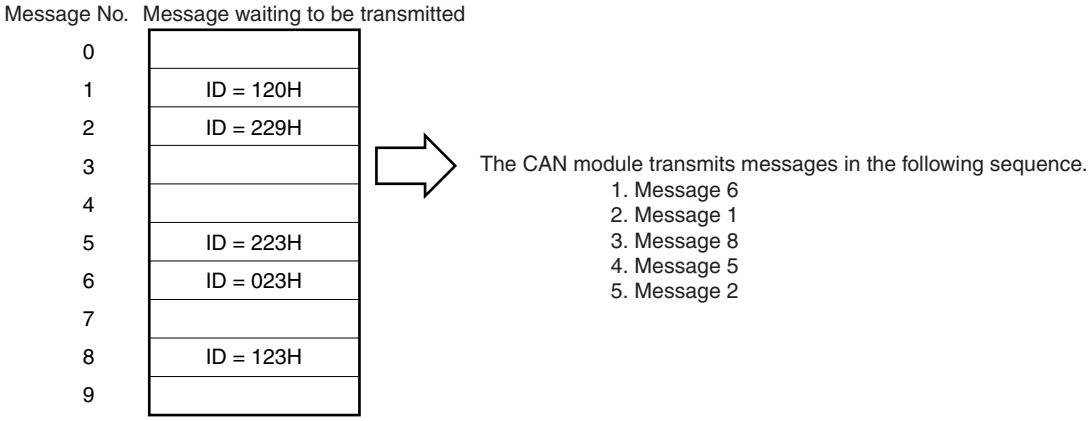
In all the operation modes, if the TRQ bit is set to 1 in a message buffer that satisfies the following conditions, the message buffer that is to transmit a message is searched.

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT [2:0] bits of C0MCONFm register set to 000B.)
- Ready for transmission
  (RDY bit of C0MCTRLm 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 14-57. Message Processing Example



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 C0INTS 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 C0IE 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

14.10.2 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 C0LOPT register, and the transmit history list get pointer (TGPT) with the corresponding C0TGPT register.

The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The C0LOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the C0LOPT 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 C0TGPT 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 C0TGPT register, the TGPT pointer is automatically incremented.

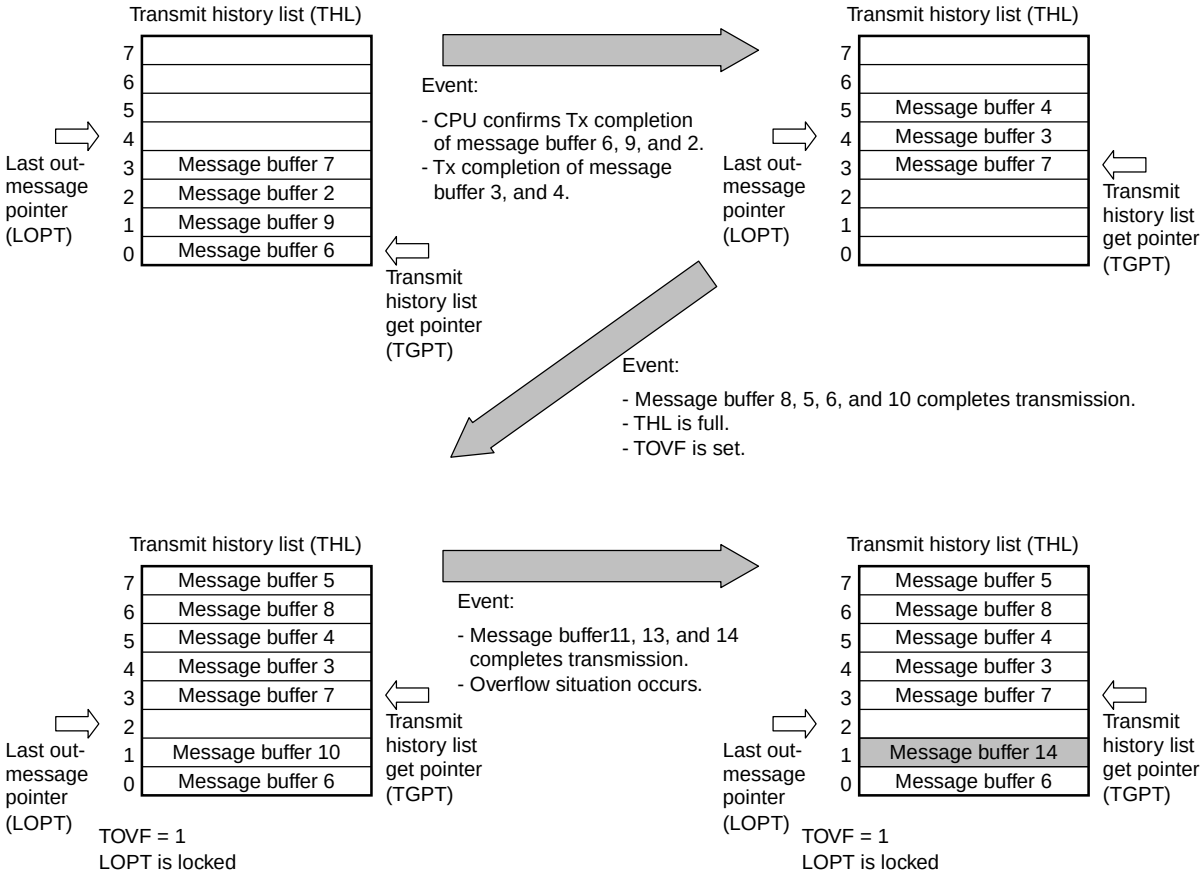
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the C0TGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT 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 C0TGPT 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 14-58. Transmit History List



TOVF = 1 denotes that LOPT equals TGPT−1 while message buffer number stored to element indicated by LOPT−1.

14.10.3 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 C0CTRL 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 C0MIDLm and C0MIDHm registers. Set the C0MDLCm and C0MDATA0m to C0MDATA7m 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 C0GMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the C0BRP and C0BTR 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 C0MCTRLm 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 C0GMABTD 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 (C0GMABTD = 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 C0GMABTD register was set up with 00H.

Remark m = 0 to 15

14.10.4 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 C0MCTRLm 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 C0CTRL register and the C0TGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 14-72**).

(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 C0GMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the C0GMABT register = 0, clear the TRQ bit of the C0MCTRLm 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 C0CTRL register and the C0TGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 14-74**).

(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 C0GMABT 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, refer to the process in **Figure 14-73**). 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, refer to the process in **Figure 14-74**).

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

14.10.5 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 C0MCONFm register. Setting (1) the RTR bit sets remote frame transmission.

Remark m = 0 to 15

14.11 Power Save Modes

14.11.1 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 C0CTRL register.

This transition request is only acknowledged only under the following conditions.

- 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
- The CAN bus state is bus idle (the 4th bit in the interframe space is recessive) Note
- 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 (C0CTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for C0LIPT, C0RGPT, C0LOPT, and C0TGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN0 Global Control register (C0GMCTRL) 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 C0CTRL 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, refer to the processing in Figure 14-81).

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 C0CTRL 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 C0INTS register is set to 1, regardless of the CIE bit of the C0IE 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

14.11.2 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 C0CTRL 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 C0CTRL 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 (C0CTRL), but nothing can be written to other CAN0 module registers or bits.
- The CAN0 module registers can be read, except for C0LIPT, C0RGPT, C0LOPT, and C0TGPT.
- The CAN0 message buffer registers cannot be written or read.
- MBON bit of the CAN0 Global Control register (C0GMCTRL) 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 C0CTRL 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

14.11.3 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, put the CPU in the power saving mode. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRxD) in this status, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the C0CTRL register is set to 1, a wakeup interrupt (INTC0WUP) is generated. The CAN module is automatically released from the CAN sleep mode (PSMODE = 00B) and returns to the normal operation mode (However, in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software after a bus change is detected at the CAN reception pin (CRxD)). The CPU, in response to INTC0WUP, can release its own power saving mode and return to the normal operation mode.

To further reduce the power consumption of the CPU, the internal clocks, including that of the CAN module, may be stopped. In this case, the operating clock supplied to the CAN module is stopped after the CAN module is put in the CAN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRxD) in this status, the CAN module can set the CINTS5 bit to 1 and generate the wakeup interrupt (INTC0WUP) even if it is not supplied with the clock. The other functions, however, do not operate because clock supply to the CAN module is stopped, and the module remains in the CAN sleep mode. 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).

14.12 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 14-20. List of CAN Module Interrupt Sources

No.	Interrupt Status Bit		Interrupt Enable Bit		Interrupt Request Signal	Interrupt Source Description
	Name	Register	Name	Register		
1	CINTS0Note	C0INTS	CIE0Note	C0IE	INTC0TRX	Message frame successfully transmitted from message buffer m
2	CINTS1Note	C0INTS	CIE1Note	C0IE	INTC0REC	Valid message frame reception in message buffer m
3	CINTS2	C0INTS	CIE2	C0IE	INTC0ERR	CAN module error state interrupt (Supplement 1)
4	CINTS3	C0INTS	CIE3	C0IE		CAN module protocol error interrupt (Supplement 2)
5	CINTS4	C0INTS	CIE4	C0IE		CAN module arbitration loss interrupt
6	CINTS5	C0INTS	CIE5	C0IE	INTC0WUP	CAN module wakeup interrupt from CAN sleep mode (Supplement 3)

Note The IE bit (message buffer interrupt enable bit) in the C0MCTRLm 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

14.13 Diagnosis Functions and Special Operational Modes

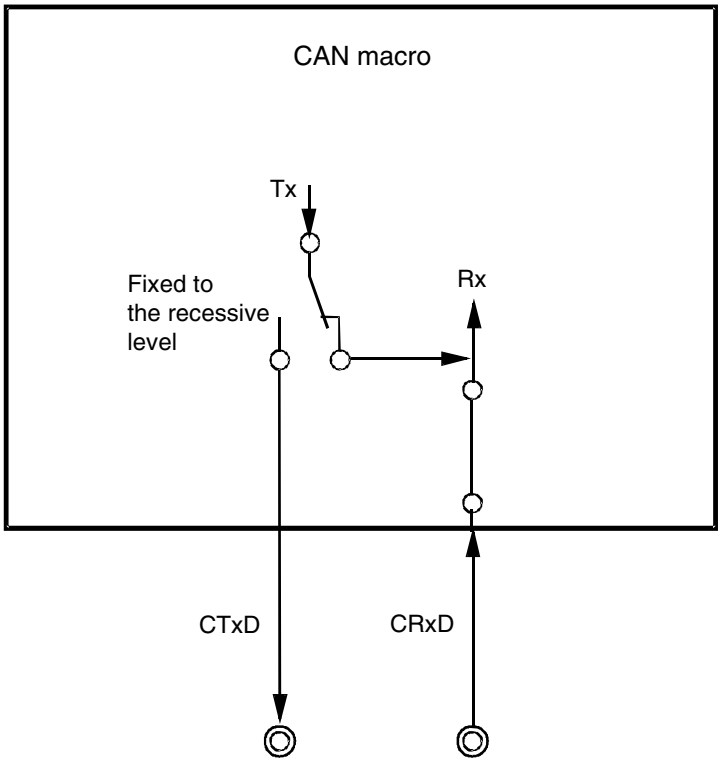
The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of specific CAN communication methods.

14.13.1 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 C0CTRL register (1).

Figure 14-59. 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.

14.13.2 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 C0CTRL 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 C0INTS register respectively, and the type of the error can be identified by reading the LEC[2:0] bits of the C0LEC register.

Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the C0INTS register is set to 1. If the CIE0 bit of the C0IE 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.

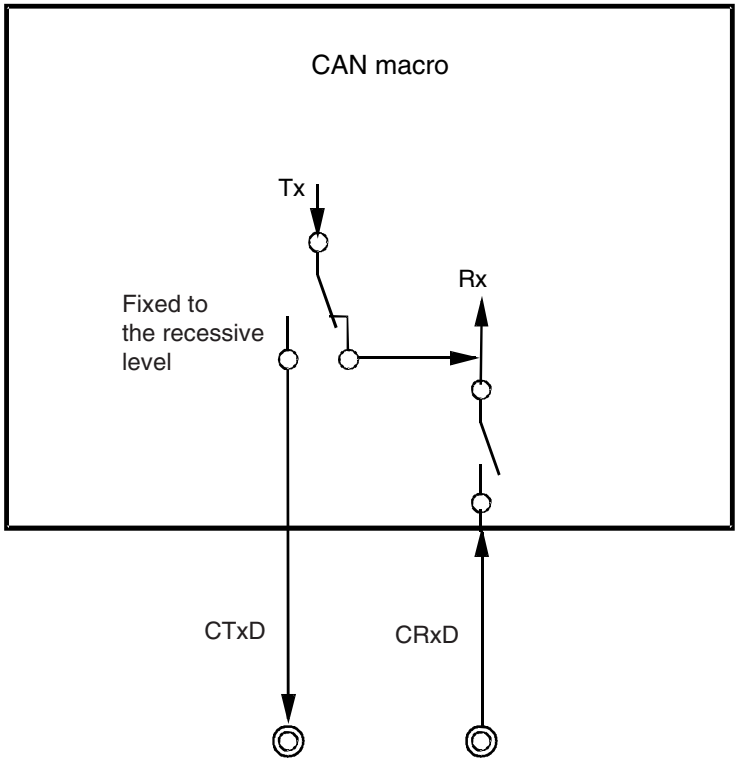
14.13.3 Self-test mode

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 14-60. CAN Module Terminal Connection in Self-test Mode



14.13.4 Receive/Transmit Operation in Each Operation Mode

Table 14-21 shows outline of the receive/transmit operation in each operation mode.

Table 14-21. Outline of the Receive/Transmit in Each Operation Mode

Operation Mode	Transmission of data/ remote frame	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 C0CTRL register.

2. Each signals are not generated to outside, but generated into the CAN module.

14.14 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.

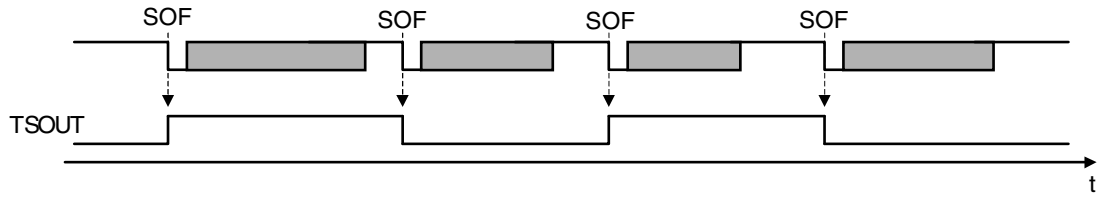
14.14.1 Time stamp function

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 C0TS 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 C0TS register to 1.

Figure 14-61. Timing Diagram of Capture Signal TSOUT



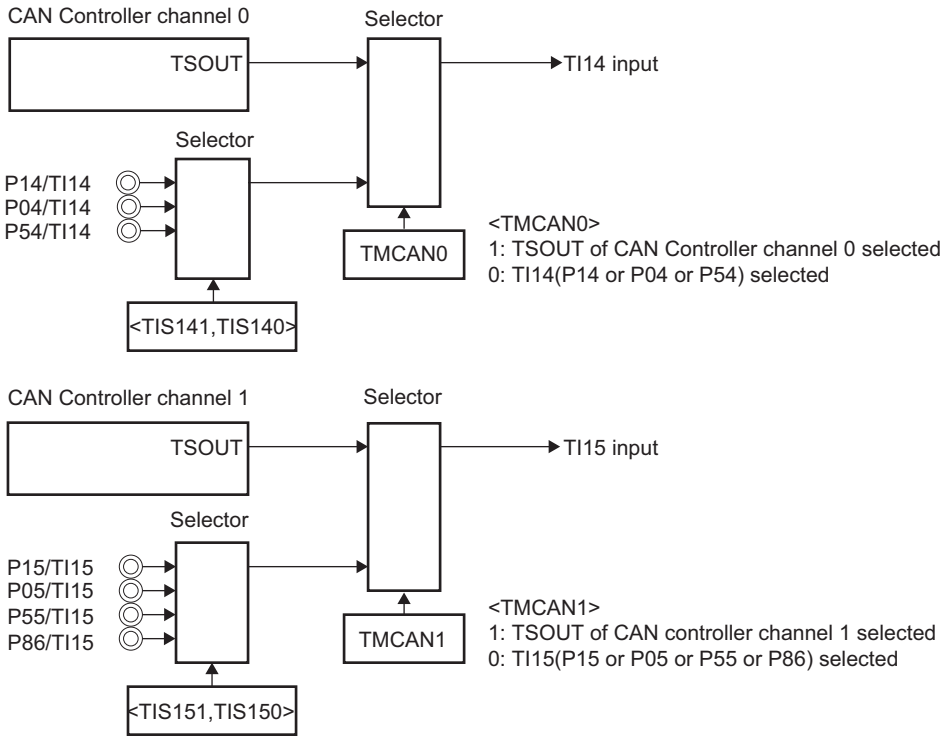
TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in the above timing diagram, the SOF is used as the trigger event source). To capture a timer value by using TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

This time stamp function is controlled by the TSLOCK bit of the C0TS 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 TMCAN0), the capture trigger signal (TSOUT of CAN Controller channel 0) can be input to channel 4 of timer array unit 1 without connecting TSOUT of CAN controller channel 0 and TI14 externally. By switching the input source (by using TMCAN1), the capture trigger signal (TSOUT of CAN Controller channel 1) can be input to channel 5 of timer array unit 1 without connecting TSOUT of CAN Controller channel 1 and TI15 externally.

Figure 14-62. Switching source of input



Remarks 1. TMCAN0, TMCAN1: Bit 0, 1 of the serial communication pin selection register (STSEL1) (see **Figure 14-51**).
TIS140, TIS141: Bit 0, 1 of timer input selection register 11 (TIS11) (see **CHAPTER 6 TIMER ARRAY UNIT**).
TIS150, TIS151: Bit 2, 3 of timer input selection register 11 (TIS11) (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**.

14.15 Baud Rate Settings

14.15.1 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 CAN0 bit rate register (C0BTR)
TSEG2 [2:0]: Bits 10 to 8 of CAN0 bit rate register (C0BTR)

Table 14-22 shows the combinations of bit rates that satisfy the above conditions.

Table 14-22. Settable Bit Rate Combinations (1/3)

Valid Bit Rate Setting					C0BTR 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 14-22. Settable Bit Rate Combinations (2/3)

Valid Bit Rate Setting					C0BTR 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 14-22. Settable Bit Rate Combinations (3/3)

Valid Bit Rate Setting					C0BTR 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
7Note	1	2	2	2	0011	001	71.4
7Note	1	4	1	1	0100	000	85.7
6Note	1	1	2	2	0010	001	66.7
6Note	1	3	1	1	0011	000	83.3
5Note	1	2	1	1	0010	000	80.0
4Note	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 C0BRP register is other than 00H.

Caution The values in Table 14-22 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.

14.15.2 Representative examples of baud rate settings

Tables 14-23 and **14-24** show representative examples of baud rate setting.

Table 14-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	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 14-23 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 14-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	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 14-23 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 14-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	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 14-24 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 14-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	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

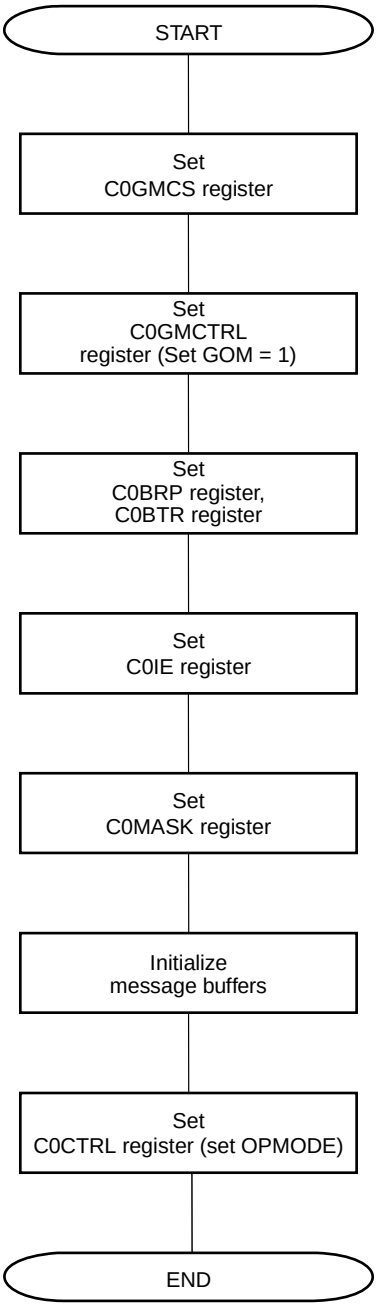
Caution The values in Table 14-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

14.16 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.

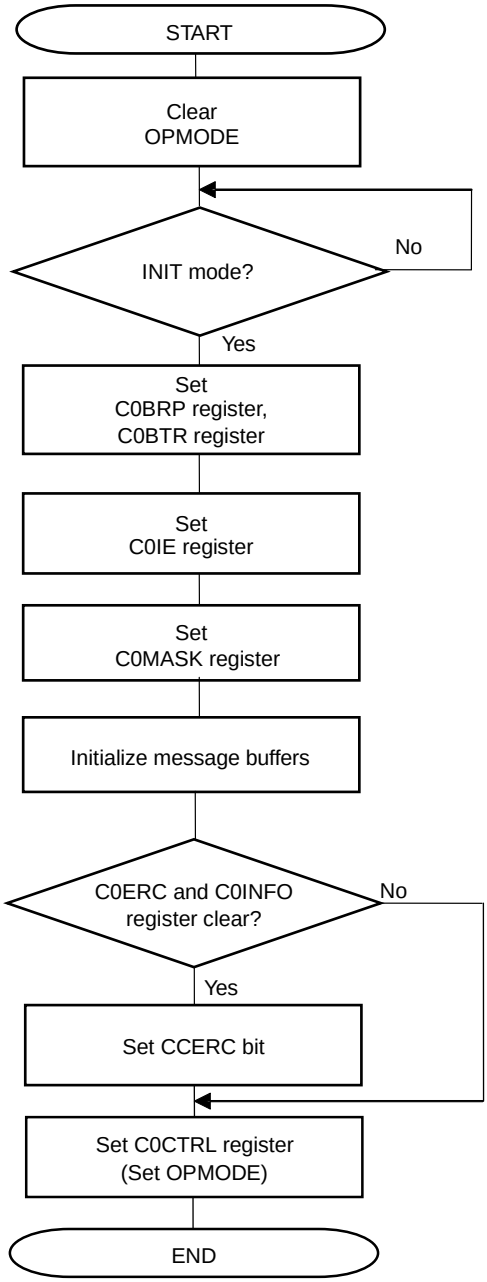
Remark m = 0 to 15

Figure 14-63. Initialization



Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

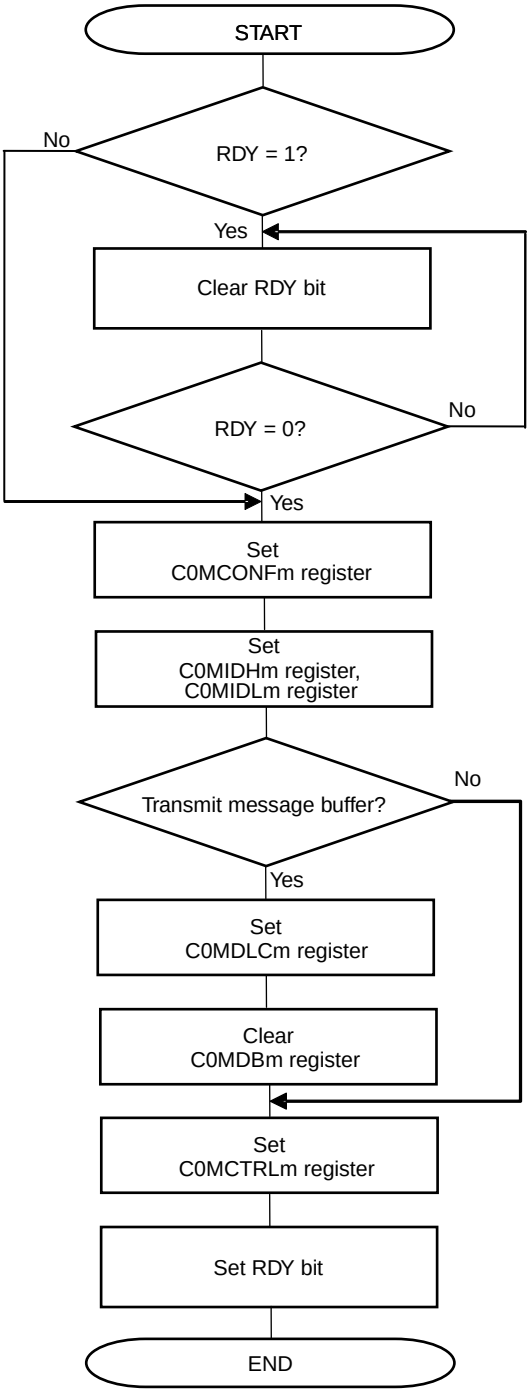
Figure 14-64. Re-initialization



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 C0CTRL and C0GMCTRL 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 14-65. Message Buffer Initialization



Cautions 1. Before a message buffer is initialized, the RDY bit must be cleared.

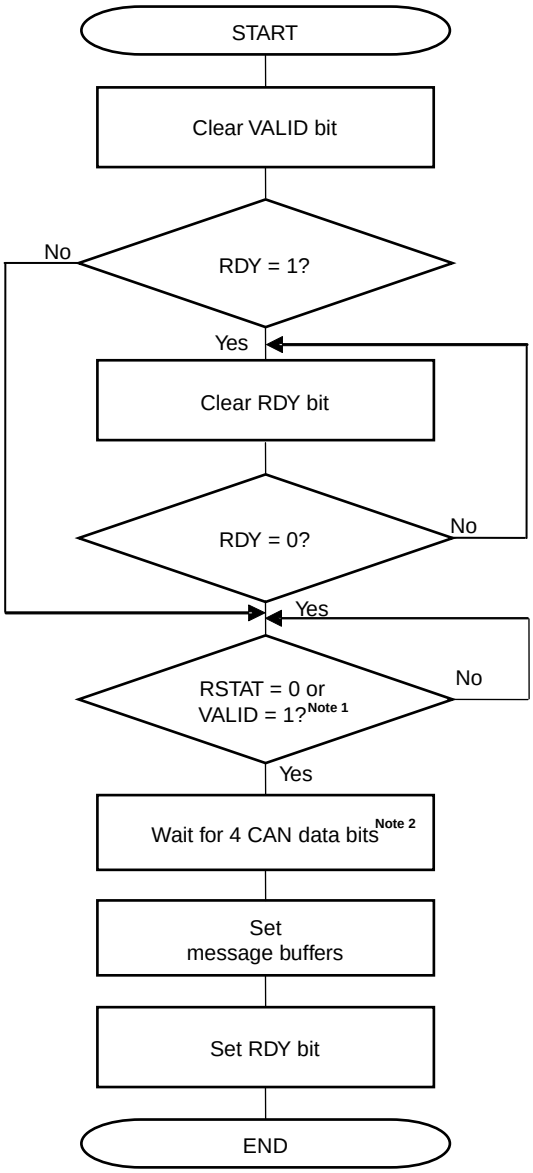
2. Make the following settings for message buffers not used by the application.

- Clear the RDY, TRQ, and DN bits of the C0MCTRLm register to 0.

- Clear the MA0 bit of the C0MCONFm register to 0.

Figure 14-66 shows the processing for a receive message buffer (MT [2:0] bits of C0MCONFm register = 001B to 101B).

Figure 14-66. 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 14-67 shows the processing for a transmit message buffer during transmission (MT [2:0] bits of C0MCONFm register = 000B).

Figure 14-67. Message Buffer Redefinition during Transmission

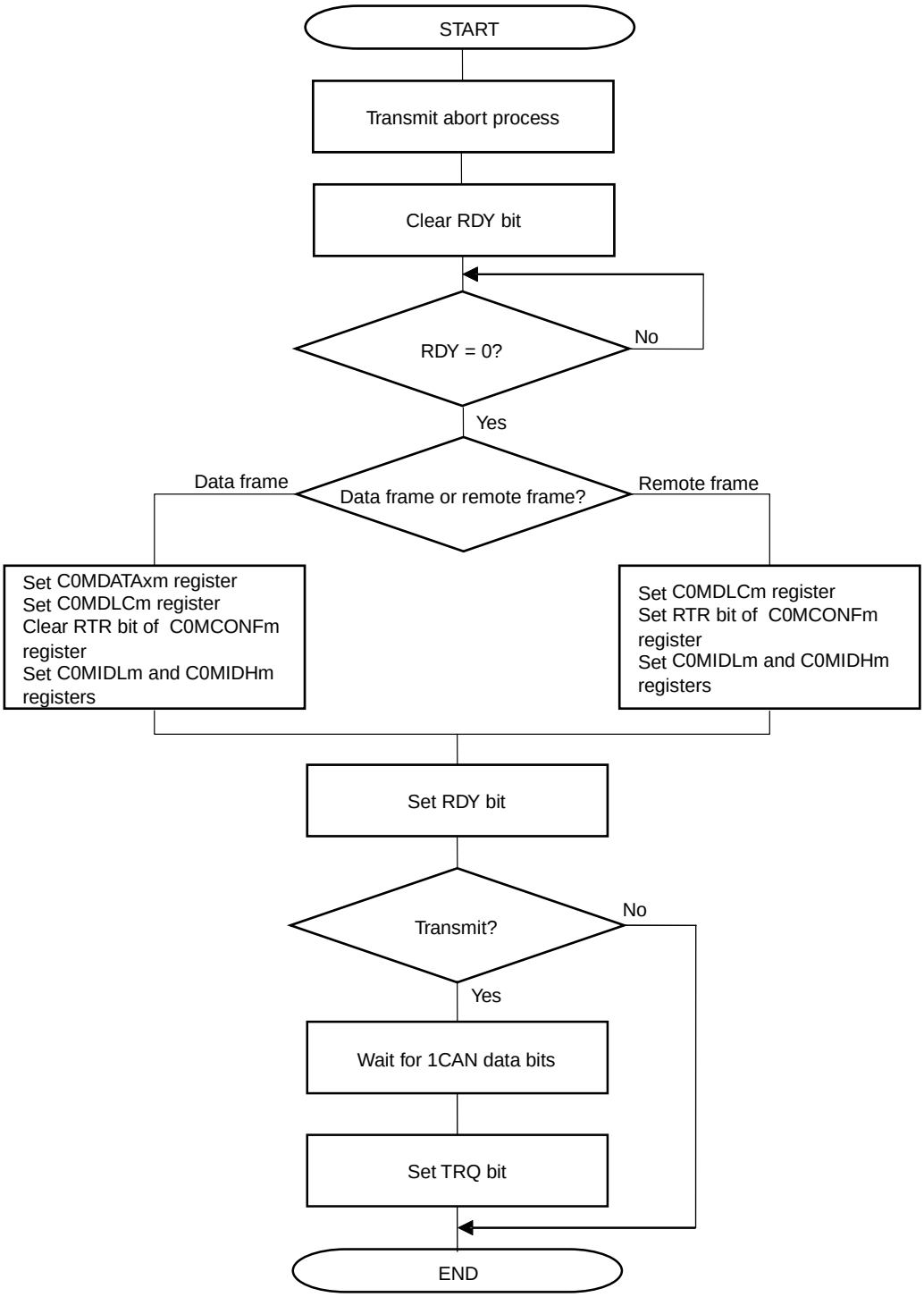
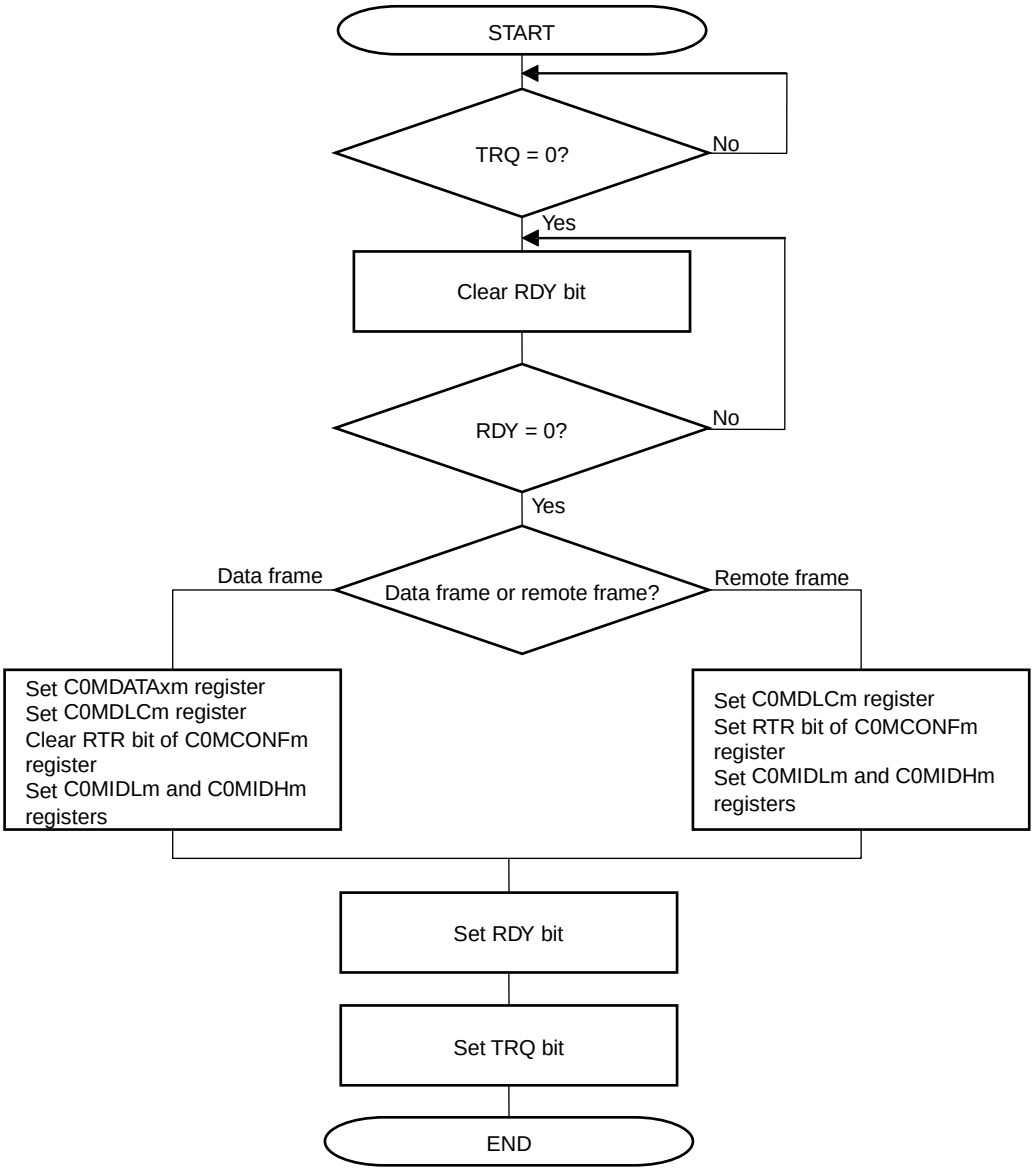


Figure 14-68 shows the processing for a transmit message buffer (MT [2:0] bits of C0MCONFm register = 000B).

Figure 14-68. Message Transmit Processing

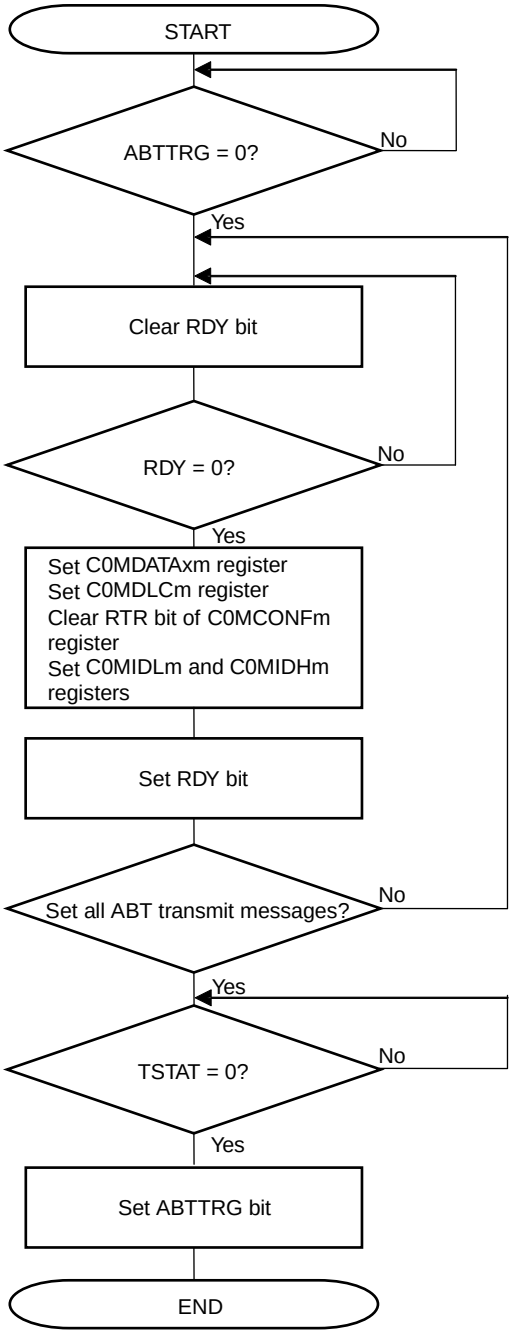


Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Figure 14-69 shows the processing for a transmit message buffer (MT [2:0] bits of C0MCONFm register = 000B).

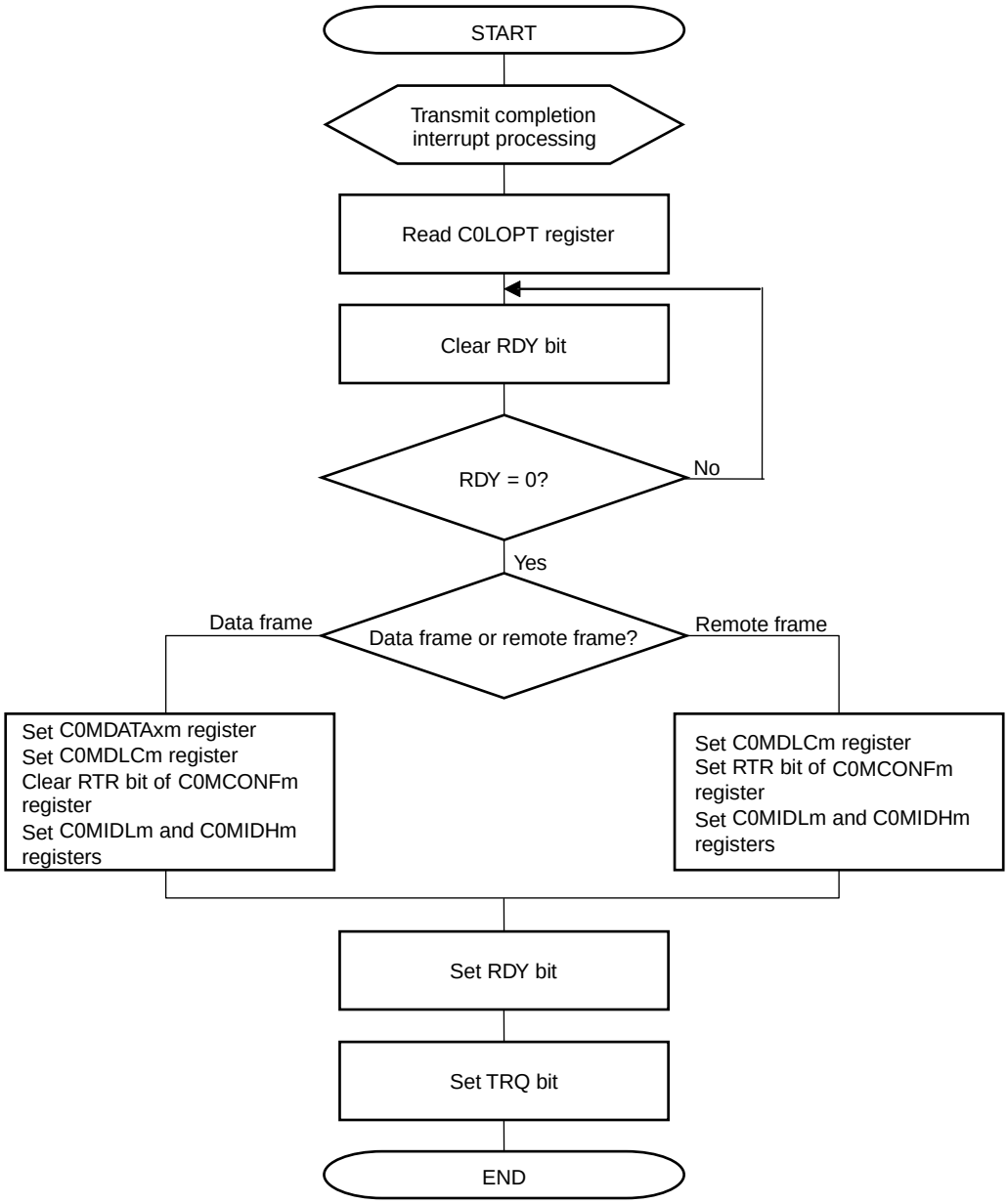
Figure 14-69. ABT Message Transmit Processing



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, refer to **Figure 14-68**.

Figure 14-70. Transmission via Interrupt (Using C0LOPT register)

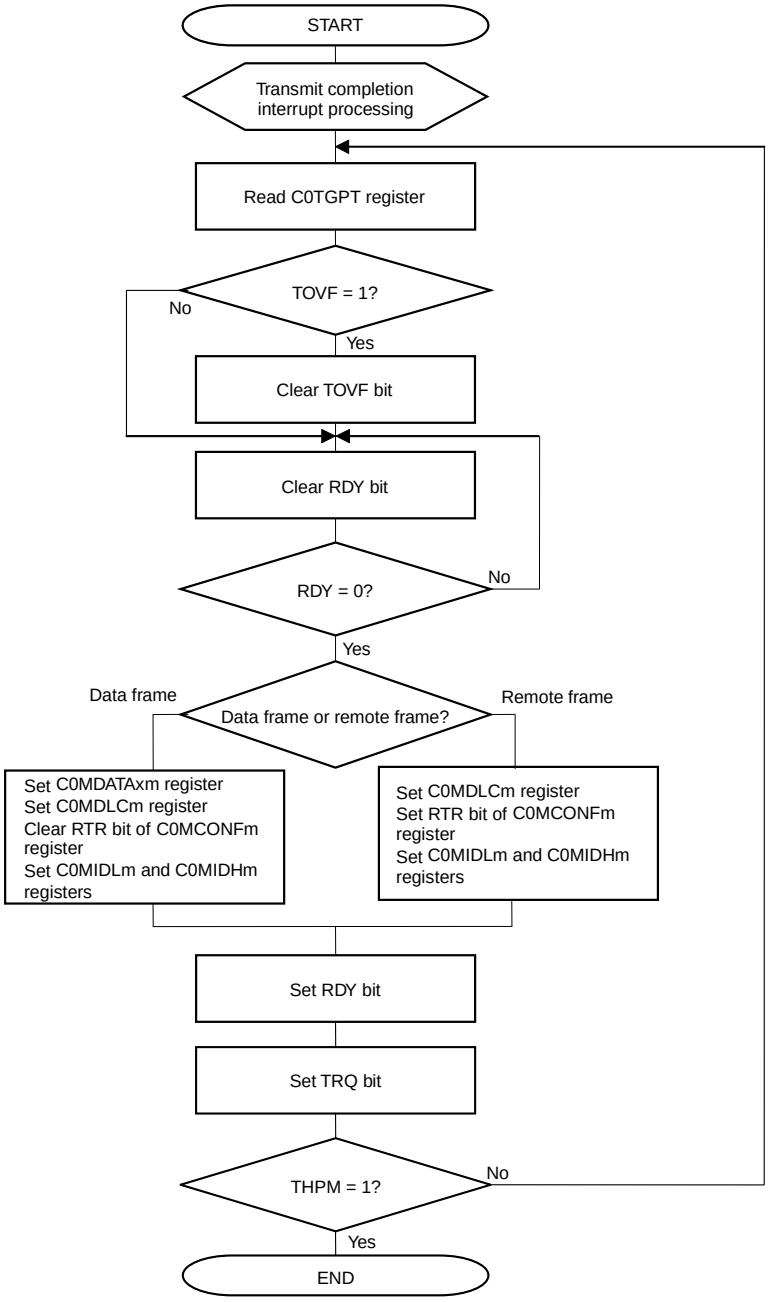


Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

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 14-71. Transmit via Interrupt (Using C0TGPT register)



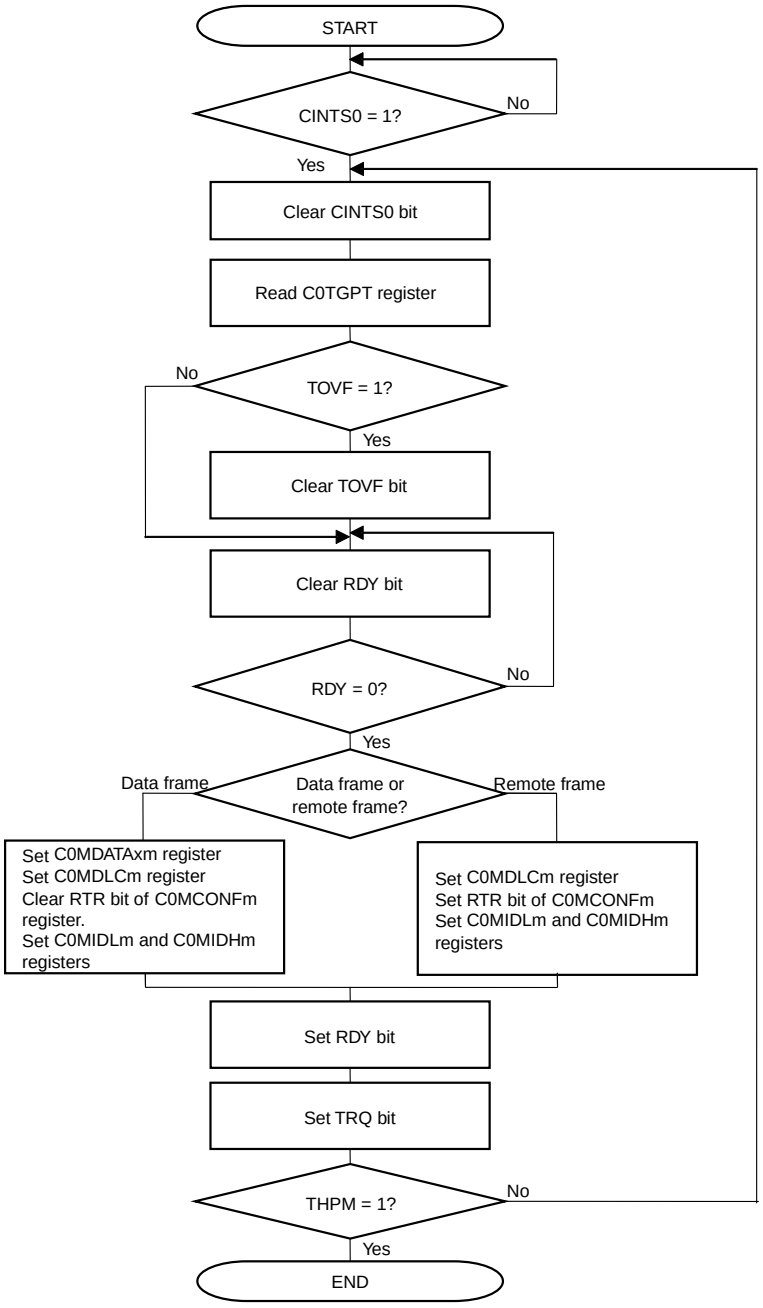
Cautions 1. The TRQ bit should be set after the RDY bit is set.

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 14-72. Transmission via Software Polling



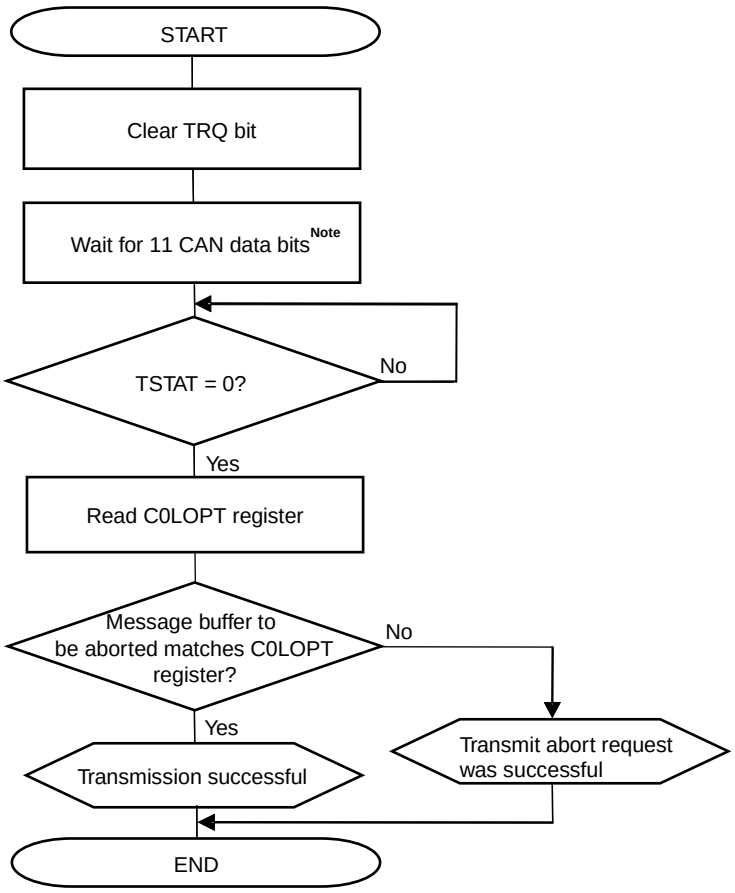
Cautions 1. The TRQ bit should be set after the RDY bit is set.

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 14-73. Transmission Abort Processing (Except Normal Operation Mode with ABT)



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.

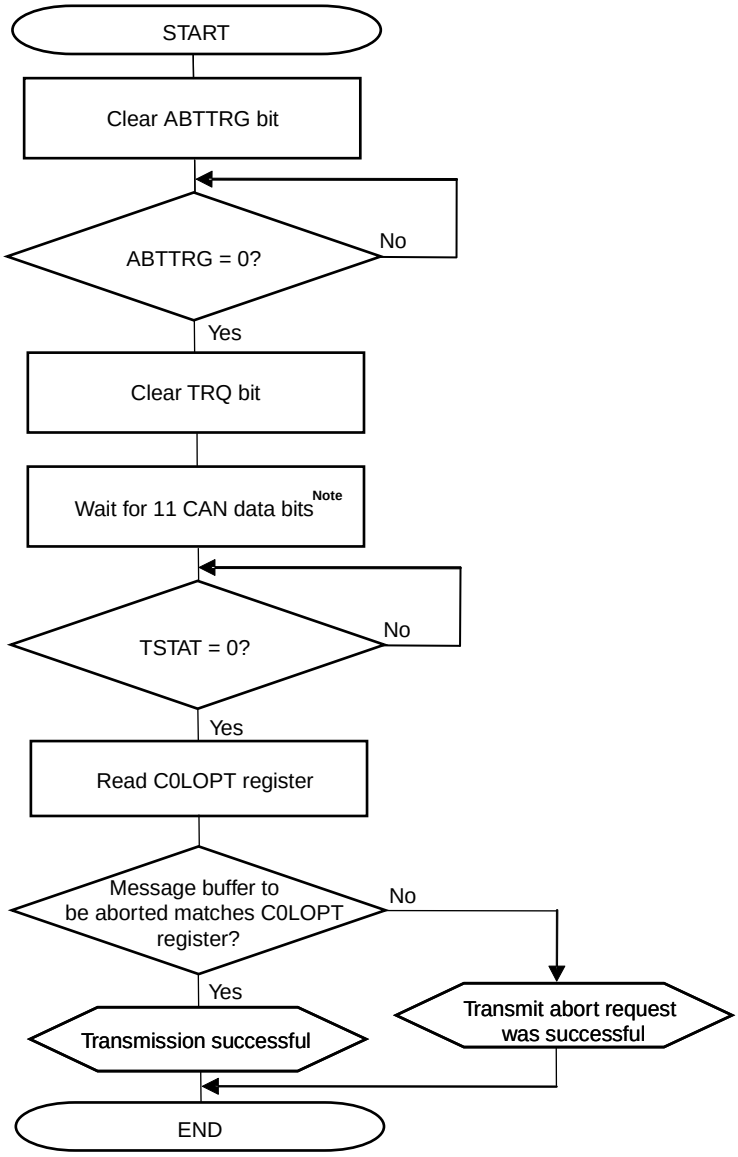
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

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 C0TGPT register indicates.

Figure 14-74. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)



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.

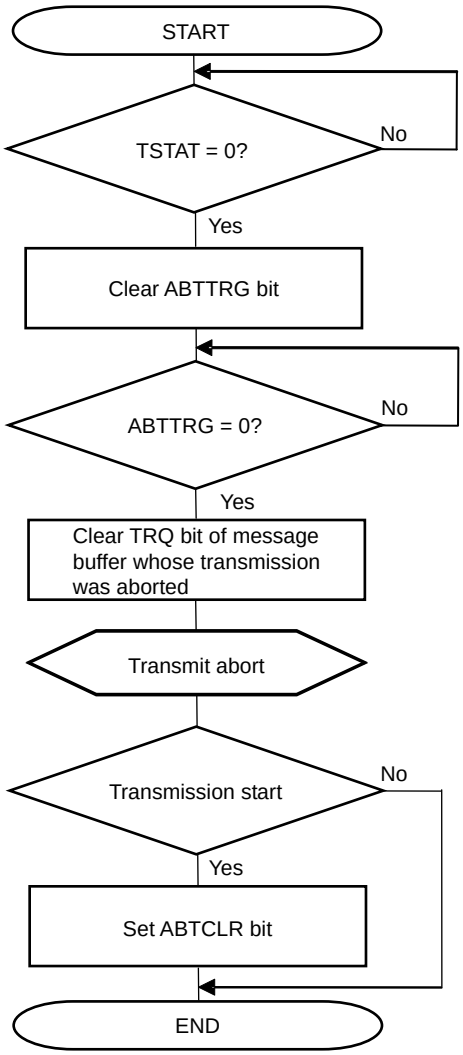
3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

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 C0TGPT register indicates.

Figure 14-75 shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 14-75. ABT Transmission Abort Processing (Normal Operation Mode with ABT)

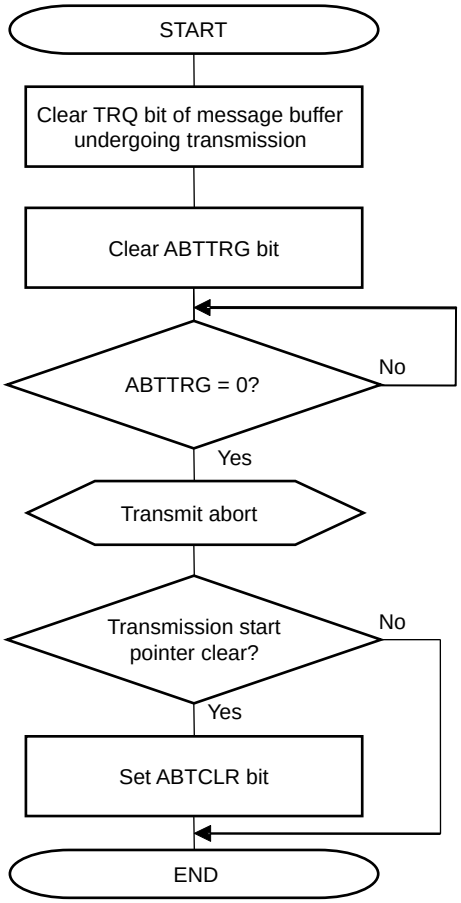


Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

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 14-75 or 14-76. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 14-74.

Figure 14-76 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

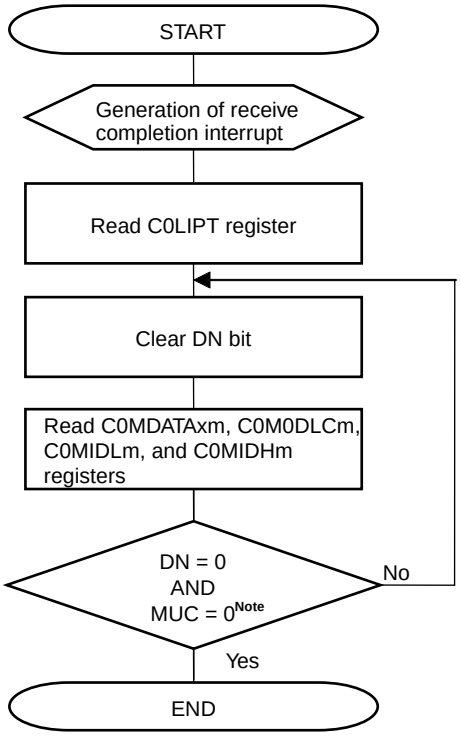
Figure 14-76. ABT Transmission Request Abort Processing (Normal Operation Mode with ABT)



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 request after ABTTRG is cleared (after ABT mode is stopped) following the procedure shown in Figure 14-75 or 14-76. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 14-74.

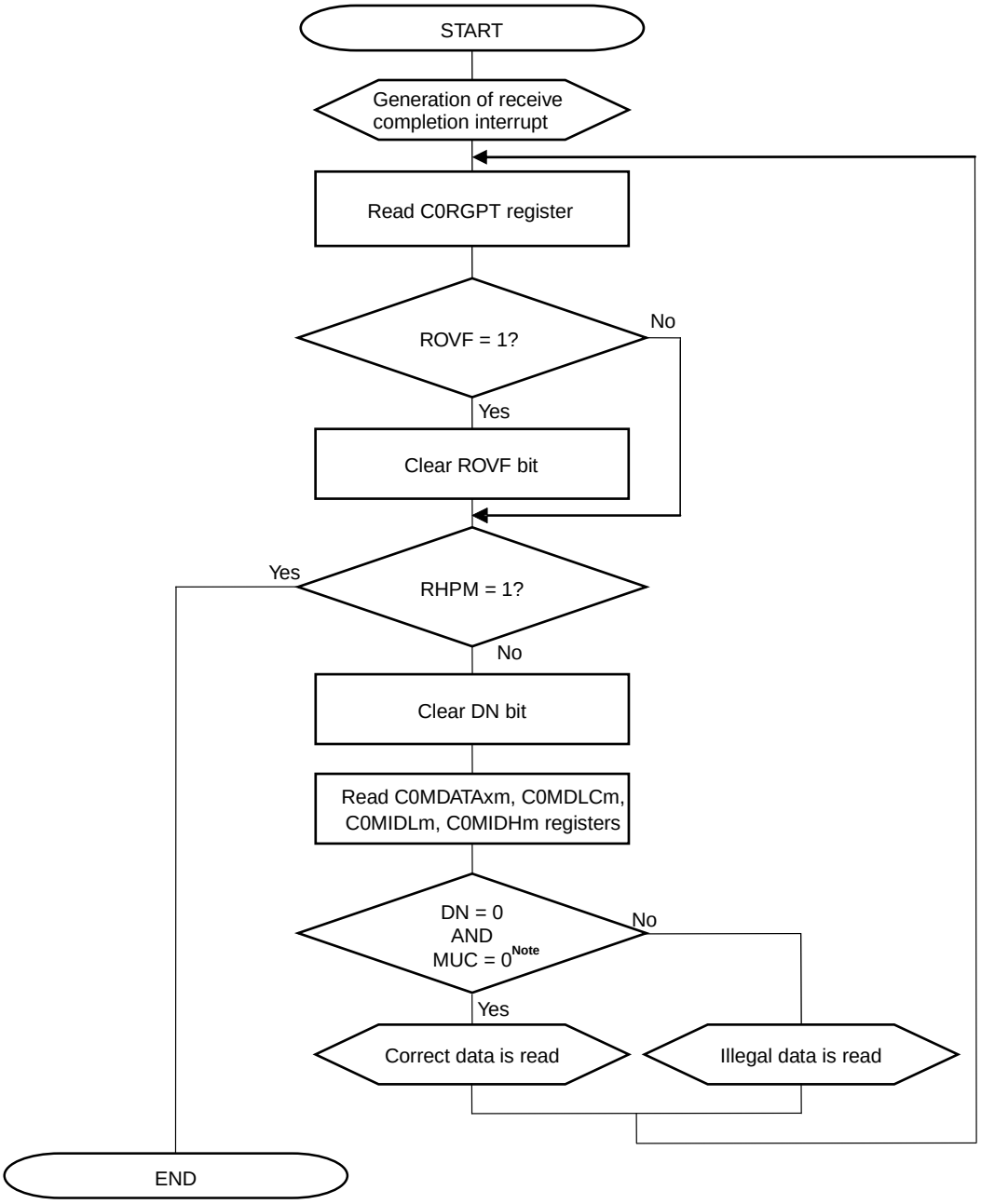
Figure 14-77. Reception via Interrupt (Using C0LIPT Register)



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 14-78. Reception via Interrupt (Using C0RGPT Register)



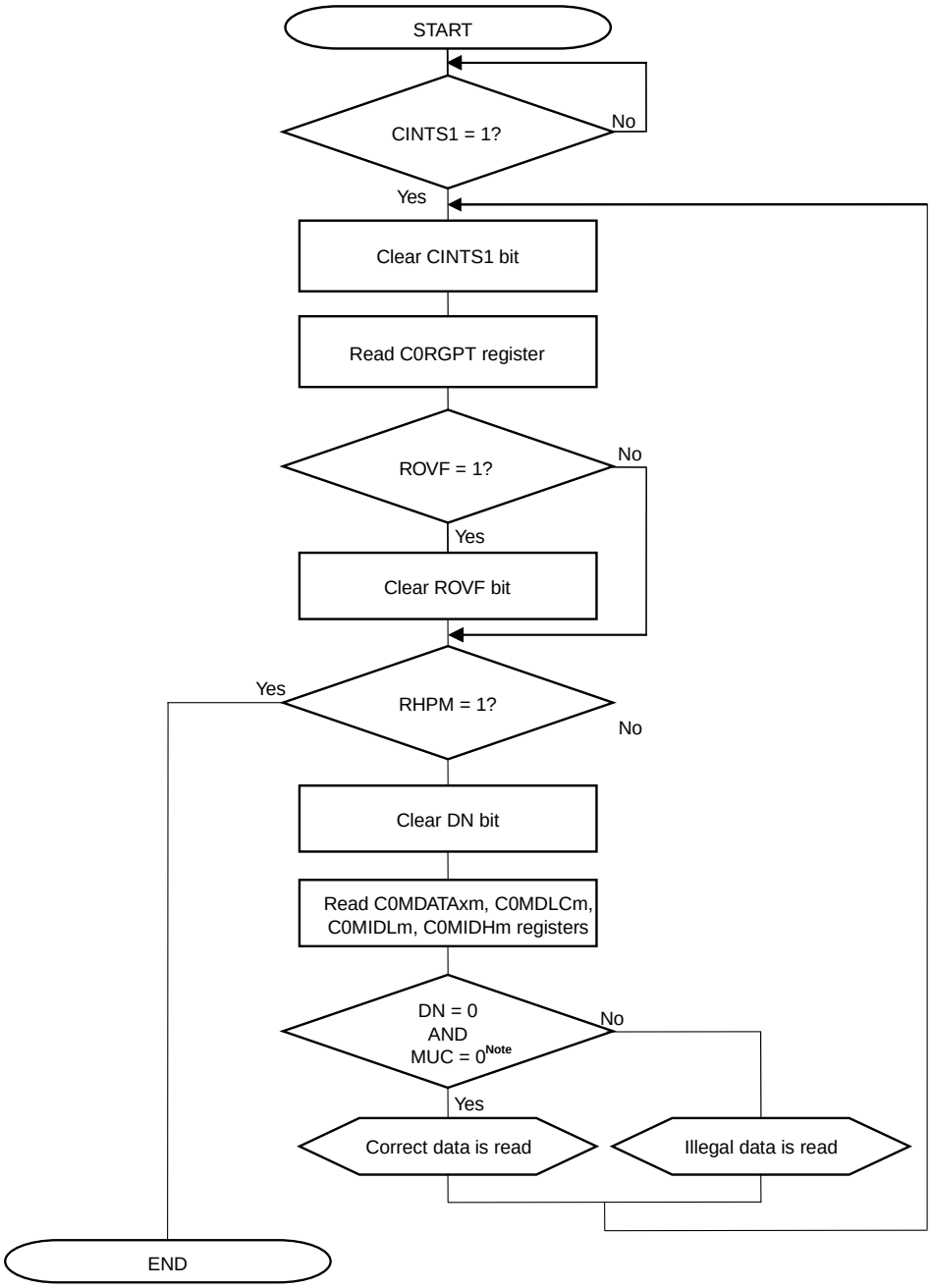
Note Check the MUC and DN bits using one read access.

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 14-79. Reception via Software Polling

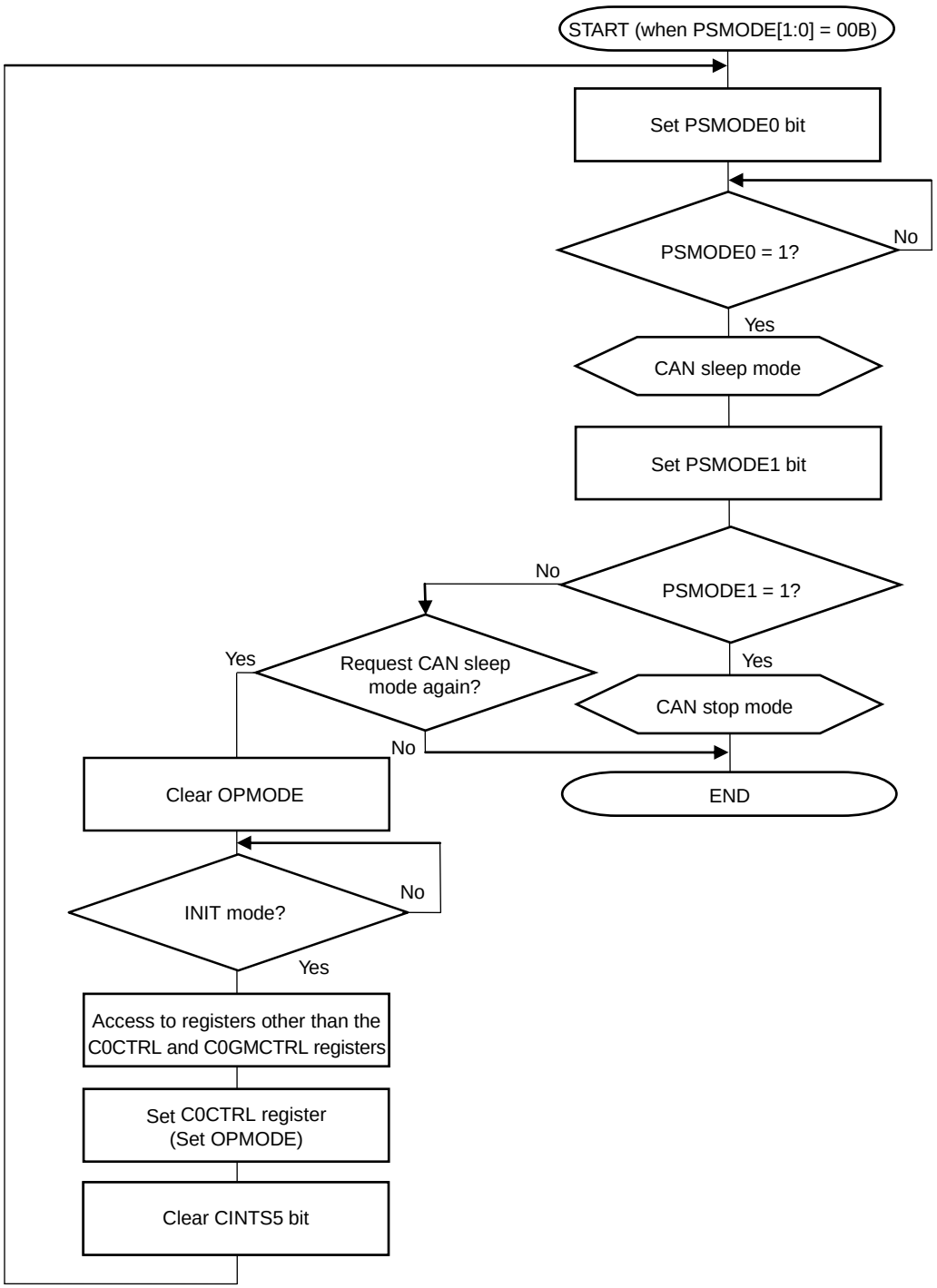


Note Check the MUC and DN bits using one read access.

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.

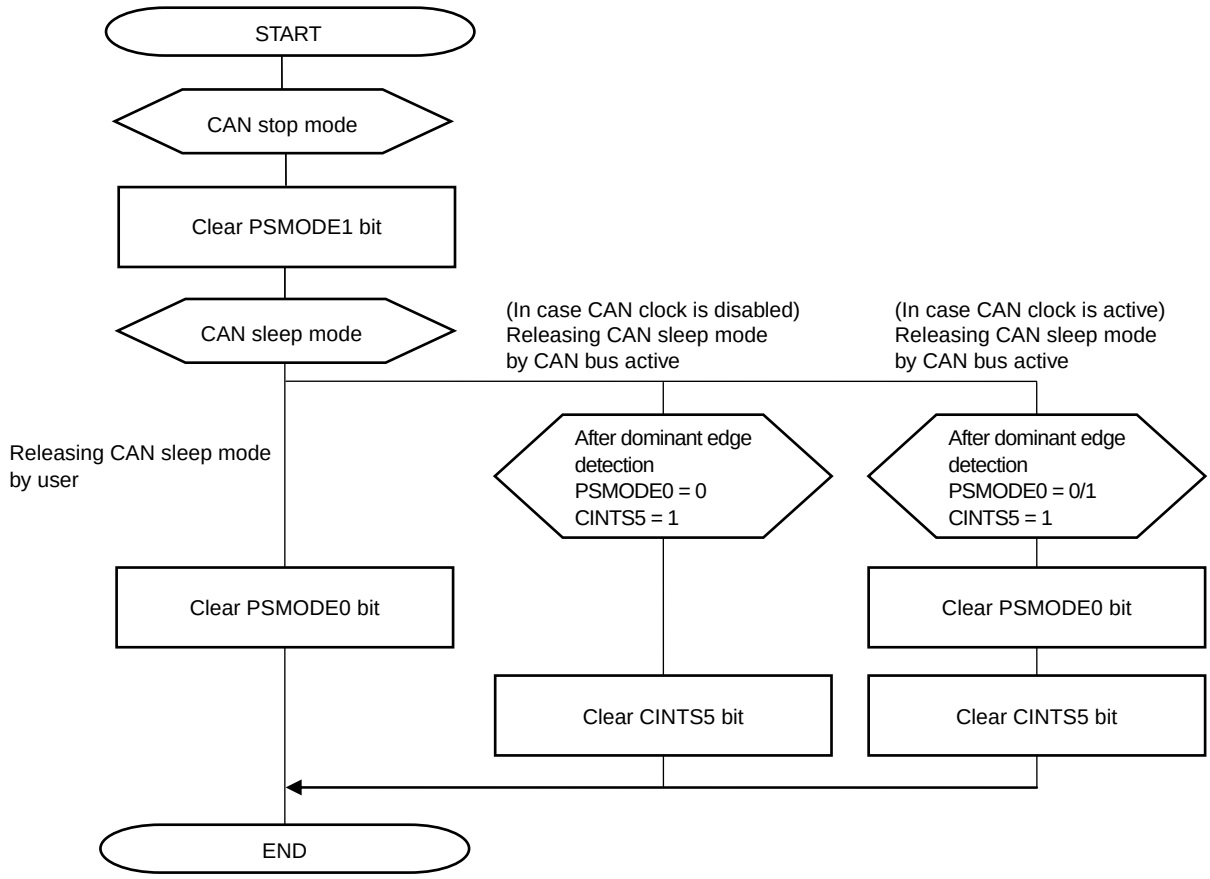
2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

Figure 14-80. Setting CAN Sleep Mode/Stop Mode



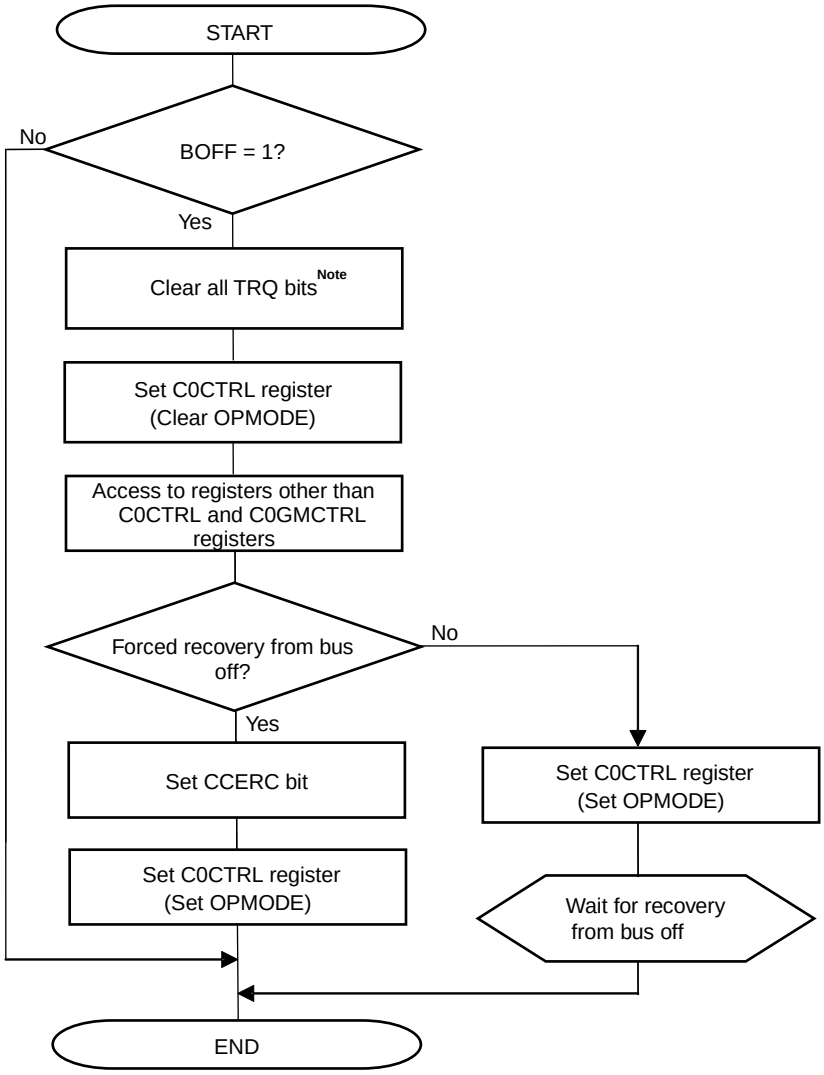
Caution To abort transmission before making a request for the CAN sleep mode, perform processing according to Figures 14-73 or 14-74.

Figure 14-81. 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 14-82. Bus-Off Recovery (Expect Normal Operation Mode with ABT)

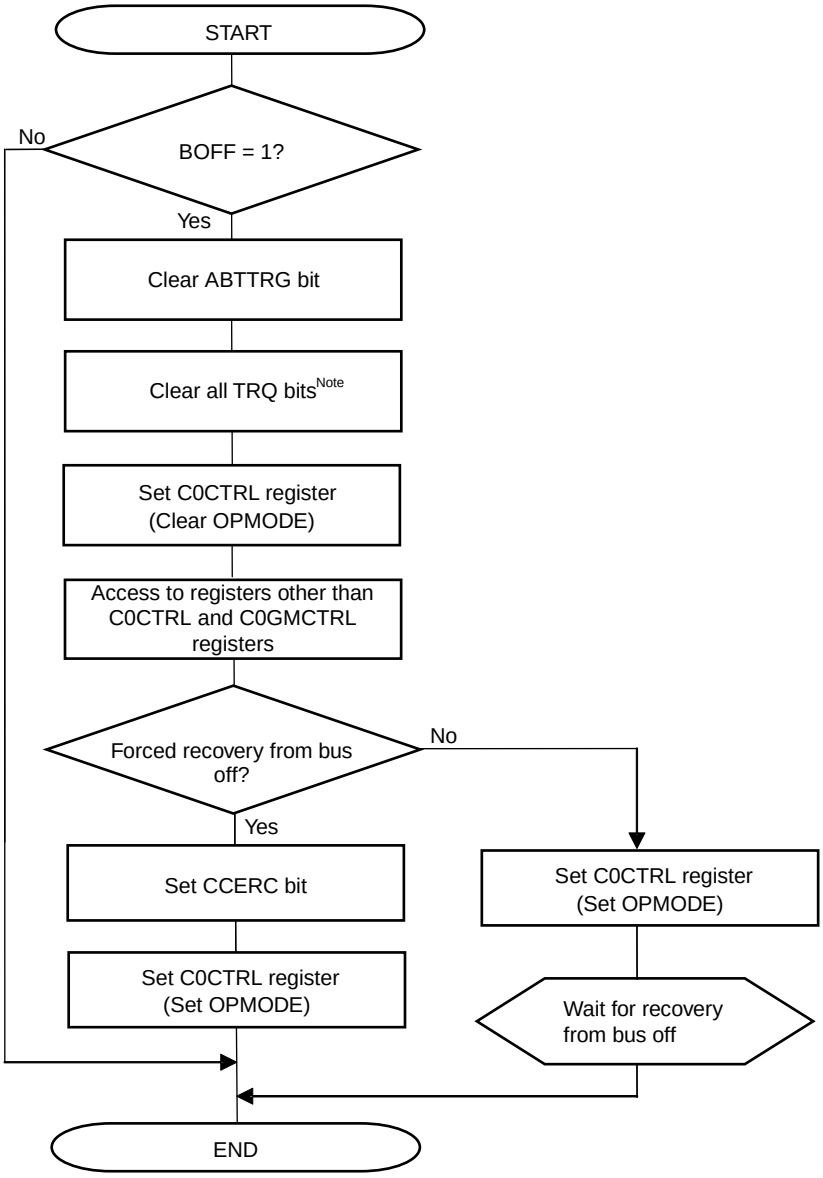


Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

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 14-83. Bus-Off Recovery (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 14-84. Normal Shutdown Process

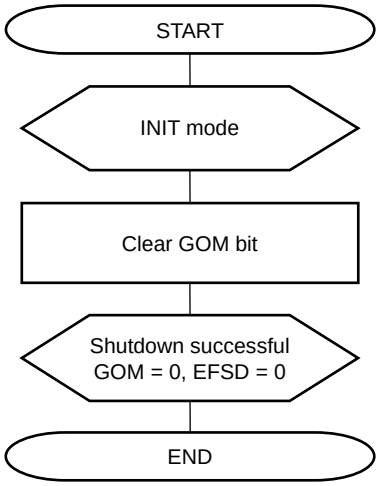
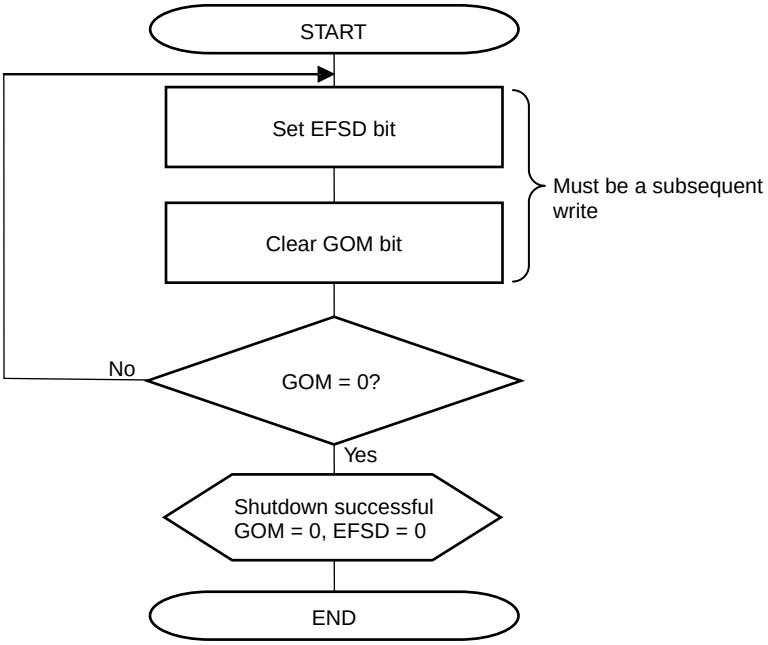


Figure 14-85. Forced Shutdown Process



Caution Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.
Note that, if interrupt or DMA occurs, it is not regarded as a sequential access and the forced shutdown request is invalidated.

Figure 14-86. Error Handling

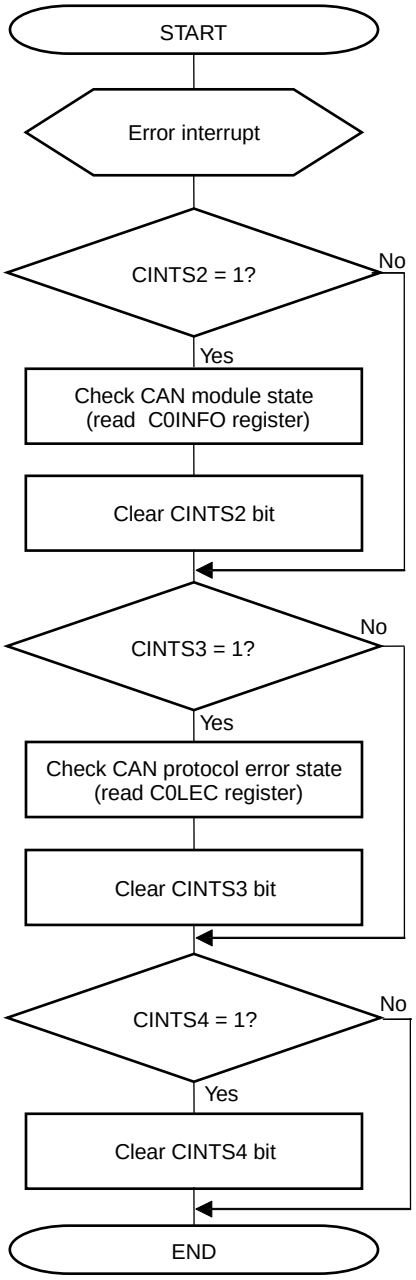
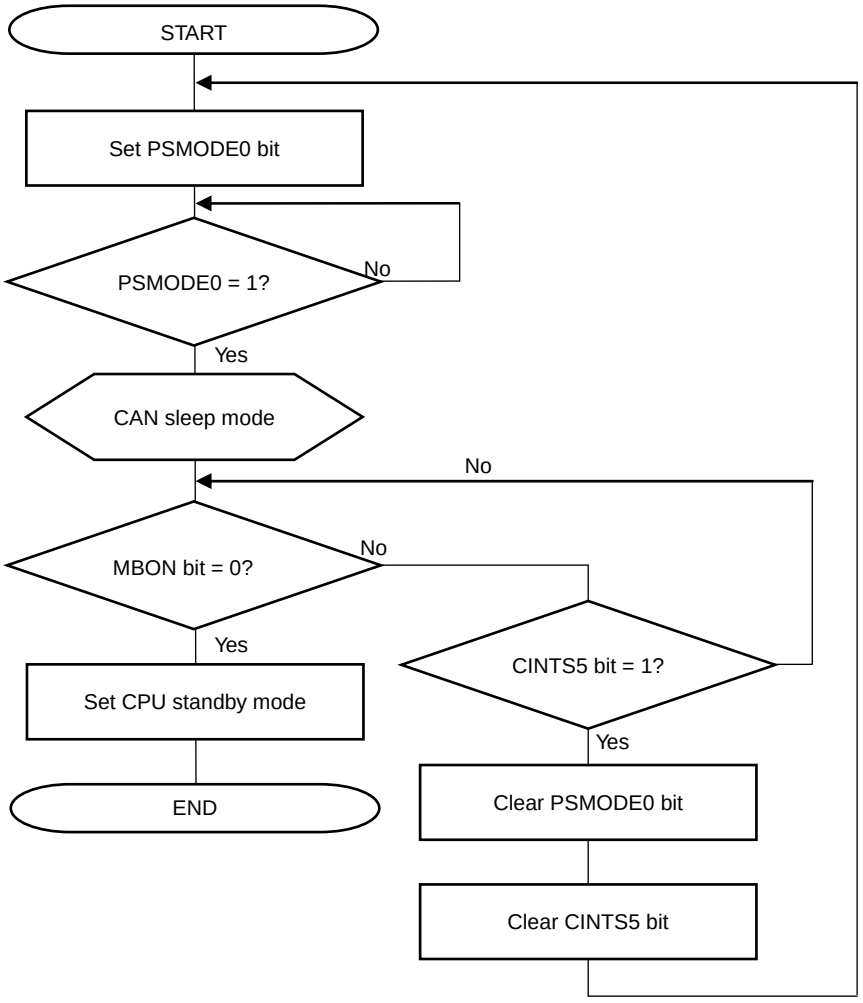
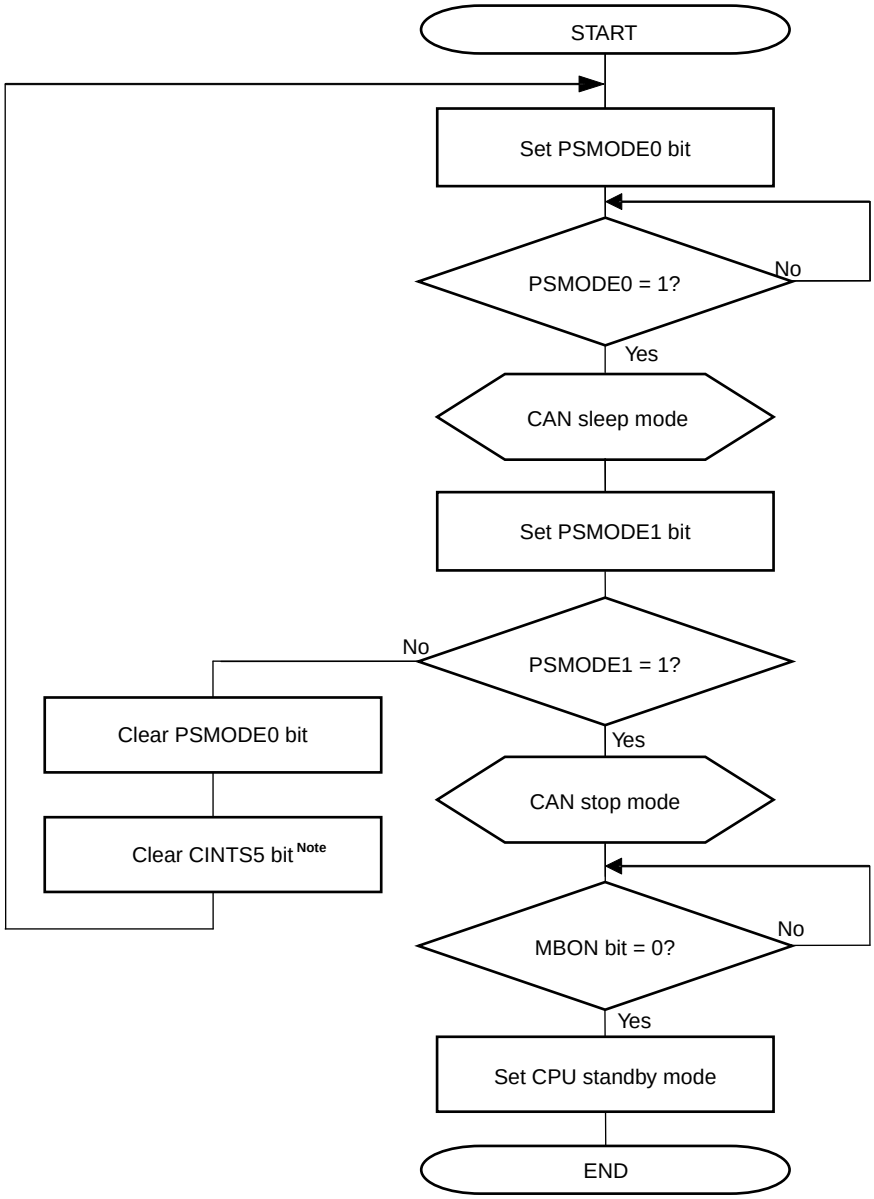


Figure 14-87. Setting CPU Standby (from CAN Sleep Mode)



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 14-88. 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 C0CTRL register and not by a change in the CAN bus state.

CHAPTER 15 STEPPER MOTOR CONTROLLER/DRIVER

<R>

	48-pin	64-pin	80-pin	100-pin	128-pin
	R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
Stepper Motor Contoller/Driver	1 channel	2 channels	4 channels		

The Stepper Motor Controller/Driver module is comprised of four drivers (k = 1 to 4) for external 360° type meters or for bipolar and unipolar stepper motors.

15.1 Overview

The Stepper Motor Controller/Driver module generates pulse width modulated (PWM) output signals. Each driver generates up to four output signals.

Features

- 8-bit precision pulse width can be specified
- Pseudo 9-bit precision pulse width can be specified by using the 1-bit addition function
- PWM output is possible at frequencies up to 20 kHz
- Automatic PWM phase shift for reducing fluctuation on power supply and for reducing the susceptibility to electromagnetic interference
- Zero Point Detection (ZPD) function

15.1.1 Driver overview

A stepper motor is driven by PWM signals. The PWM signals are generated by comparing the contents of compare registers with the actual value of a free running up counter. The Stepper Motor Controller/Driver module contains one counter and assigned compare registers and control registers.

Figure 15-1 shows the main components of the Stepper Motor Controller/Driver. The Stepper Motor Controller/Driver includes a free running up counter (CNT0). The counter is controlled by a timer mode control register (MCNTC0). Each of the four drivers consists of two compare registers, MCMPk0 and MCMPk1, respectively. Their contents define the pulse widths for the sine and the cosine side of the meters. The MCMPk0/MCMPk1 registers comprise a master-slave register combination. This allows to re-write the master register while the slave register is currently used for comparison with the counter CNT0.

The compare control register MCMPCk defines whether or not enhanced pulse width precision by one-bit addition is enabled, and it routes the output signals to the corresponding output pins (SMk1 to SMk4).

15.1.2 ZPD introduction

Zero Point Detection (ZPD) enables calibration of meter needle without additional pins.

When the needle reaches the zero-point of the meter, the induction voltage generated to this point by back electromotive force drops. ZPD function has the circuit to detect the induced voltage which its reference voltage could be configured in advance. Each of the four drivers has one ZPD circuit.

For reliable results of the ZPD circuit, digital noise removal is applied.

For details on the internal reference voltage, refer to **CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT) (TARGET)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT) (TARGET)**.

15.1.3 ZPD input pins

When Zero Point Detection is enabled and the MCMPCk.TWIN bit is set, every driver can be used for calibration. Dedicated pins of each driver are then used as input pins to allow for Zero Point Detection of the connected meter.

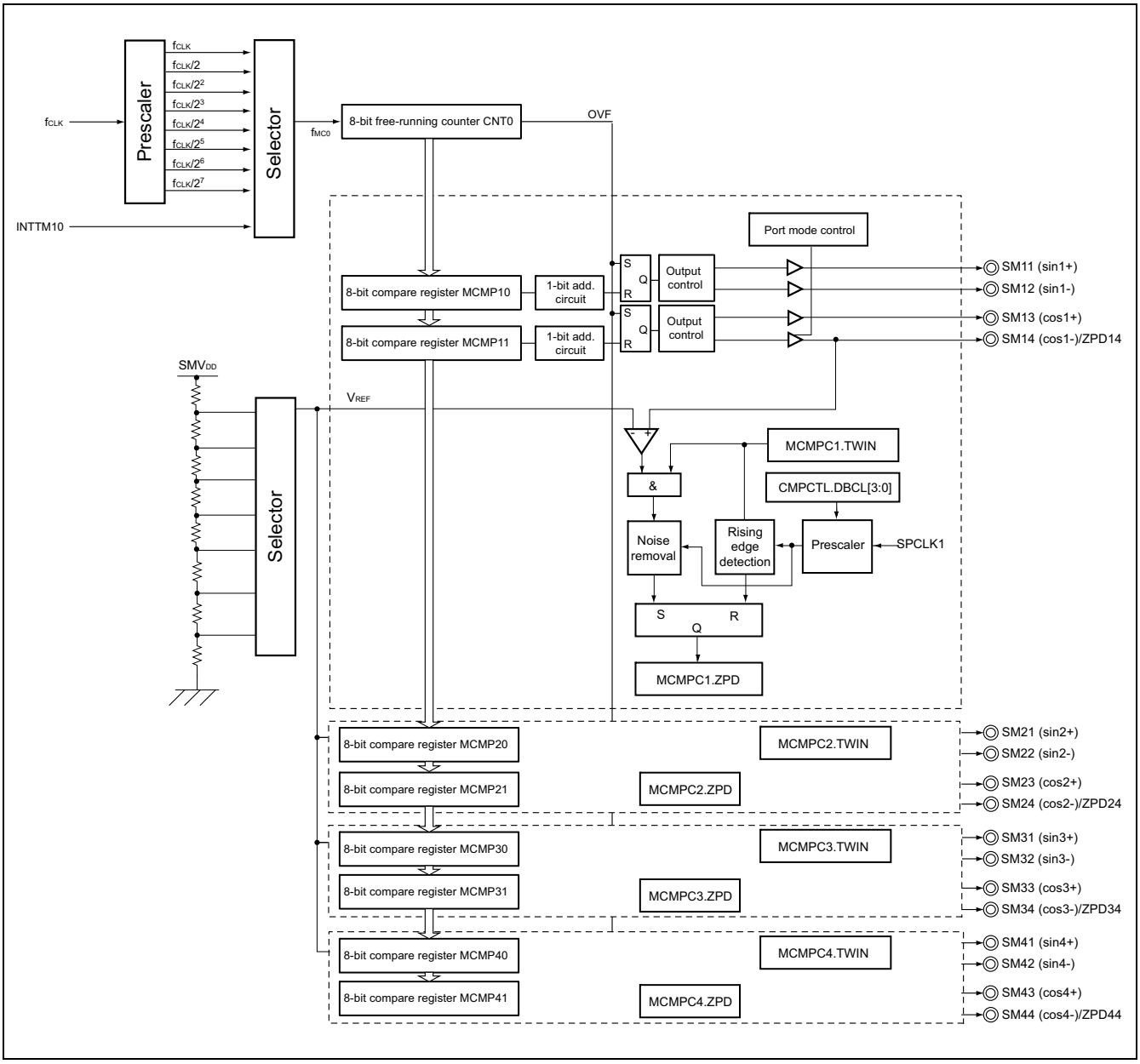
- At Stepper Motor Controller/Driver, the SMk4 pin (k = 1 to 4) can be used as ZPD input.

Note that these pins have to be configured as input pins.

The result of a voltage comparison is reflected in the MCMPCk.ZPD bit.
Properties of the ZPD can be set in the ZPD flag detection clock setting register CMPCTL.
Figure 15-1 shows the ZPD circuit of the Stepper Motor Controller/Driver.

Figure 15-1. Stepper Motor Controller/Driver Block Diagram

<R>



The external signals are listed in the following table.

Table 15-1. Stepper Motor Controller/Driver External Connections

Signal name	I/O	Active level	Reset level	Pins	Function
SM[1:4]1	O	–	L	SM11 to SM41	Driver signal, sine side (+)
SM[1:4]2	O	–	L	SM12 to SM42	Driver signal, sine side (–)
SM[1:4]3	O	–	L	SM13 to SM43	Driver signal, cosine side (+)
SM[1:4]4	O	–	L	SM14 to SM44	Driver signal, cosine side (–)

15.2 Stepper Motor Controller/Driver Registers

The Stepper Motor Controller/Driver is controlled and operated by means of the following registers:

Table 15-2. Stepper Motor Controller/Driver Registers Overview

Register name	Shortcut
Timer mode control registers	MCNTC0
Compare registers	MCMPk0 (k = 1 to 4)
	MCMPk1 (k = 1 to 4)
	MCMPkHW (k = 1 to 4)
Compare control registers	MCMPCk (k = 1 to 4)
Stepper motor port mode control register	SMPC
ZPD detection voltage setting registers	ZPDS0, ZPDS1
ZPD flag detection clock setting register	CMPCTL
ZPD operational control register	ZPDEN
Peripheral enable registers 1	PER1

(1) Timer mode control register (MCNTC0)

The 8-bit MCNTC0 register controls the operation of the free running up counters CNT0.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 15-2. Format of Timer Mode Control Register (MCNTC0) (1/2)

Address: F0160H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCNTC0	CAE	0	FULL	PCE	PCS	SMCL2	SMCL1	SMCL0

CAE	Stepper Motor Controller/Driver control
0	Stepper Motor Controller/Driver operation is disabled.
1	Stepper Motor Controller/Driver operation is enabled.

FULL	Count range of the timer counter
0	Count range from 01H to FFH
1	Count range from 00H to FFH
The initial start value is 00H in both cases. For the impact of this bit on duty factor and PWM cycle time, see also **15.3.1 (3) Duty factor**.	

PCE	Timer operation control
0	Timer counter is stopped.
1	Timer counter is enabled.

PCS	Timer count clock
0	Count clock specified by SMCL2 to SMCL0
1	INTTM10

Figure 15-2. Format of Timer Mode Control Register (MCNTC0) (2/2)

SMCL2	SMCL1	SMCL0	Selected timer count clock
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/2$
0	1	0	$f_{CLK}/2^2$
0	1	1	$f_{CLK}/2^3$
1	0	0	$f_{CLK}/2^4$
1	0	1	$f_{CLK}/2^5$
1	1	0	$f_{CLK}/2^6$
1	1	1	$f_{CLK}/2^7$

Caution Bit 6 must be 0.

Power save mode preparation

Before entering any power save mode the Stepper-C/D must be shut down in advance in order to minimize power consumption.

Apply following sequence to shut down the Stepper-C/D:

1. Stop the counter CNT0 by setting MCNTC0.PCE = 0.
2. Disable the Stepper-C/D operation by setting MCNTC0.CAE = 0.

Remark Note that the MCNTC0.PCE and MCNTC0.CAE bits must not be cleared to 0 by a single write instruction. Perform two write instructions as shown above.

(2) Compare registers for sine side (MCMPk0) (k = 1 to 4)

The 8-bit MCMPk0 registers hold the values that define the PWM pulse width for the sine side of the connected meters.

The contents of the registers are continuously compared to the timer counter value:

- Registers MCMP10 to MCMP40 are compared to CNT0.

When the register contents match the timer counter contents, a match signal is generated. Thus a PWM pulse with a pulse width corresponding to the MCMPk0 register contents is output to the sine side of the connected meter.

These registers can be read/written in 8-bit units.

This register is cleared by any reset.

Address: F0162H, F0164H, F0166H, F0168H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPk0	sine DATA							

Remarks 1. New data must only be written to registers MCMPk0 if the corresponding bit MCMPCk.TEN = 0.

2. Don't write to the compare register MCMPk0, until the corresponding bit MCMPCk.TEN has been reset to 0 automatically.

3. To enable master-to-slave register copy upon next CNTm overflow set MCMPCk.TEN = 1.

(3) Compare registers for cosine side (MCMPk1) (k = 1 to 4)

The 8-bit MCMPk1 registers hold the values that define the PWM pulse width for the cosine side of the connected meters.

The contents of the registers are continuously compared to the timer counter value:

- Registers MCMP11 to MCMP41 are compared to CNT0.

When the register contents match the timer counter contents, a match signal is generated. Thus a PWM pulse with a pulse width corresponding to the MCMPk1 register contents is output to the cosine side of the connected meter.

These registers can be read/written in 8-bit units.

This register is cleared by any reset.

Address: F0163H, F0165H, F0167H, F0169H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPk1	cosine DATA							

Remarks 1. New data must only be written to registers MCMPk1 if the corresponding bit MCMPCk.TEN = 0.

2. Don't write to the compare register MCMPk1, until the corresponding bit MCMPCk.TEN has been reset to 0 automatically.

3. To enable master-to-slave register copy upon next CNTm overflow set MCMPCk.TEN = 1.

(4) Combined compare registers (MCMPkHW) (k = 1 to 4)

The 16-bit MCMPkHW registers combine the sine and cosine registers MCMPk0 and MCMPk1. Via these registers it is possible to read or write the contents of MCMPk0 and MCMPk1 in a single instruction.

These registers can be read/written in 16-bit units.

This register is cleared by any reset.

Address: F0162H, F0164H, F0166H, F0168H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMPkHW	cosine DATA								sine DATA							

Remarks 1. New data must only be written to registers MCMPkHW if the corresponding bit MCMPCk.TEN = 0.

2. Don't write to the compare register MCMPkHW, until the corresponding bit MCMPCk.TEN has been reset to 0 automatically.

3. To enable master-to-slave register copy upon next CNTm overflow set MCMPCk.TEN = 1.

(5) Compare control registers (MCMPCk) (k = 1 to 4)

The 8-bit MCMPCk registers control the operation of the corresponding compare registers and the output direction of the PWM pin.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 15-3. Format of Compare Control Registers (MCMPCk) (1/2)

Address: F016AH, F016CH, F016EH, F0158H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPCk	AOUTk	TWINk	ZPDk[Note]	TENk	ADBk1	ADBk0	DIRk1	DIRk0

Note This bit may be written, but writing is ignored.

AOUTk[Note 1]	Output pins for sine and cosine signals
0	The PWM signals for sine and cosine side are output to those pins that are selected by bits DIRk0 and DIRk1. At all other pins, the output signal is 0 (SMV_{SS} level).
1	The PWM signal for the sine side is output to pins SMk1 and SMk2. The PWM signal for the cosine side is output to pins SMk3 and SMk4.

TWINk[Note 1]	0-point detection timing window
0	Disable writing to ZPD bit from the comparator (No 0-point detection)
1	Enable writing to ZPD bit from the comparator (0-point detection)

ZPDk	Induced voltage detection bit for 0-point detection (Read only)
0	No induced voltage detection (0-point detection) The ZPDk bit is cleared to 0 when the value of the TWINk bit is changed from 0 to 1.
1	Induced voltage detection (No 0-point detection)

TENk	Transfer enable control bit
0	MCMPk0/MCMPk1 master-to-slave register copy is disabled. New data can be written to compare registers MCMPk0 or MCMPk1.
1	MCMPk0/MCMPk1 master-to-slave register copy is enabled. The copy process will take place when CNT0 overflows. Don't write to compare registers MCMPk0 or MCMPk1 while MCMPCk.TEN = 1.
Remark This bit functions as a control bit and status flag. It is automatically reset to zero upon the next timer counter overflow.	

ADBk1	1-bit addition circuit control (cosine side)
0	No 1-bit addition to PWM signal
1	1-bit addition to PWM signal

ADBk0	1-bit addition circuit control (sine side)
0	No 1-bit addition to PWM signal
1	1-bit addition to PWM signal

Figure 15-3. Format of Compare Control Registers (MCMPCk) (2/2)

DIRk1 Note 2	DIRk0 Note 2	Selected output pins
0	0	Quadrant 1: SMk1 (sin +), SMk3 (cos +)
0	1	Quadrant 2: SMk1 (sin +), SMk4 (cos –)
1	0	Quadrant 3: SMk2 (sin –), SMk4 (cos –)
1	1	Quadrant 4: SMk2 (sin –), SMk3 (cos +)

Selects the output pins for the PWM signals.

Bits DIR1 and DIR0 address the quadrant to be activated by sine and cosine. The PWM signal is routed to the specific pin with respect to the sin/cos of each quadrant.

At the other output pins, the output level is SMV_{SS}.

Remark These bits are only considered if bit AOUT is set to 0.

Notes 1. This bit can be rewritten when the TENk bit is 0.

2. This bit can be rewritten when the TENk bit is 0. DIRk1 and DIRk0 can also be rewritten when the TENk bit changes from 0 to 1.

(6) Stepper motor port mode control register (SMPC)

The 8-bit SMPC register controls output mode of SMkm pins (k = 1 to 4, m = 1 to 4).

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 15-4. Format of Stepper Motor Port Mode Control Register (SMPC)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SMPC	MOD4	MOD3	MOD2	MOD1	EN4	EN3	EN2	EN1

ENk	MODk	Port mode selection (k = 1 to 4)
0	–	Port mode All SMkm (m = 1 to 4) are set to port function.
1	0	PWM full bridge mode SMkm (m = 1 to 4) are set to full bridge output control mode.
1	1	PWM half bridge mode Depending on the DIRkn (n = 0, 1) bit of the compare control registers (MCMPCk), SMkm (m = 1 to 4) are set to PWM output control mode or port mode.

An example of settings when k = 1 is as follows:

EN1	MOD1	DIR11	DIR10	PWM Output Pin Control				Output Mode
				SM11 (sin+)	SM12 (sin–)	SM13 (cos+)	SM14 (cos–)	
0	–	–	–	port	port	port	port	Port mode
1	0	0	0	PWM	0	PWM	0	PWM full bridge mode
1	0	0	1	PWM	0	0	PWM	
1	0	1	0	0	PWM	0	PWM	
1	0	1	1	0	PWM	PWM	0	
1	1	0	0	PWM	port	PWM	port	PWM half bridge mode
1	1	0	1	PWM	port	port	PWM	
1	1	1	0	port	PWM	port	PWM	
1	1	1	1	port	PWM	PWM	port	

Caution Set port registers (Pn) and port mode registers (PMn) whose pins are not in the PWM mode but 0 in the table to 00H in the PWM full bridge mode.

(7) ZPD detection voltage setting registers (ZPDS0, ZPDS1)

The 8-bit ZPDS0, ZPDS1 registers set ZPD detection voltage and control ZPD analog input.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 15-5. Format of ZPD detection voltage setting registers (ZPDS0, ZPDS1)

Address: F015CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ZPDS0	ZPD2PC	ZPD2S2	ZPD2S1	ZPD2S0	ZPD1PC	ZPD1S2	ZPD1S1	ZPD1S0

Address: F015DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ZPDS1	ZPD4PC	ZPD4S2	ZPD4S1	ZPD4S0	ZPD3PC	ZPD3S2	ZPD3S1	ZPD3S0

ZPDkS2	ZPDkS1	ZPDkS0	0-point detection voltage setting for ZPDk (k = 1 to 4)
0	0	0	SMV_{DD}×6/200 = 0.15 V
0	0	1	SMV_{DD}×10/200 = 0.25 V
0	1	0	SMV_{DD}×14/200 = 0.35 V
0	1	1	SMV_{DD}×18/200 = 0.45 V
1	0	0	SMV_{DD}×22/200 = 0.55 V
1	0	1	SMV_{DD}×9/200 = 0.225 V
1	1	0	SMV_{DD}×11/200 = 0.275 V
Other than the above			Setting prohibited

ZPDkPC	Analog input/digital port selection
0	Digital port/SM pin
1	ZPD analog input

(8) ZPD flag detection clock setting register (CMPCTL)

The 8-bit CMPCTL register controls the clock for noise elimination.

This register can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 15-6. Format of ZPD Flag Detection Clock Setting Register (CMPCTL)

Address: F015EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPCTL	0	0	0	0	DBCL3	DBCL2	DBCL1	DBCL0

DBCL3	DBCL2	DBCL1	DBCL0	Selected clock
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
Other than the above				Setting prohibited

(9) ZPD operational control register (ZPDEN)

The 8-bit ZPDEN register controls ZPD operation.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 15-7. Format of ZPD Operational Control Register (ZPDEN)

Address: F015FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ZPDEN	0	0	0	0	ZPD4EN	ZPD3EN	ZPD2EN	ZPD1EN

ZPDkEN	ZPDk comparator operation (k = 1 to 4)
0	Disables operation.
1	Enables operation.

(10) Peripheral enable registers 1 (PER1)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware

that is not used is stopped in order to reduce the power consumption and noise.

When the Stepper Motor Controller/Driver is used, be sure to set bit 5 (MTRCEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 15-8. Format of Peripheral Enable Registers 1 (PER1)

Address: F00F1H After reset: 00H R/W (Note: Bits 0 to 3 and 6 are Read Only)

Symbol	7	6	5	4	3	2	1	0
PER1	ADCEN	0	MTRCEN	SGEN	0	0	0	0

MTRCEN	Control of stepper motor controller/driver clock supply
0	Stops input clock supply. • SFR used by the stepper motor controller/driver cannot be written. • The stepper motor controller/driver is in the reset status.
1	Supplies input clock. • SFR used by stepper motor controller/driver can be read and written.

15.3 Operation

In the following, the operation of the Stepper Motor Controller/Driver module as a driver for external meters is described.

15.3.1 Stepper motor controller/driver operation

This section describes the generation of PWM signals of the driver k for driving external meters. Further, the explanation about the duty factor, operation of 1-bit addition circuit, detecting zero points and digital noise filter is shown.

Remark k = 1 to 4

(1) Driving meters

External meters can be driven both in full bridge configuration and in half bridge configuration:

- Driving meters in full bridge configuration

  Deflection of the needle of a meter in full bridge configuration is determined by the sine and cosine value of its desired angle. Since the PWM signals do not inherit a sign, separate signals for positive and negative sine and cosine values are generated.

  The four signals at pins SMk1 to SMk4 of the driver k are:

  – sine side, positive (sin +)
  – sine side, negative (sin –)
  – cosine side, positive (cos +)
  – cosine side, negative (cos –)

  Two output control circuits select which signal (sign) for sine side and cosine side is output (bits MCMPCk.DIR[1:0]). At the remaining two output pins, the signal is set to low level.

  To drive meter k in full bridge mode, set bit MCMPCk.AOUT to 0.

- Driving meters in half bridge configuration

  In this mode, the same signal is sent to both sine pins (SMk1 and SMk2) and both cosine pins (SMk3 and SMk4), respectively. The setting of output control bits MCMPCk.DIR[1:0] is neglected.

  To drive meter k in half bridge mode, set bit MCMPCk.AOUT to 1.

(2) Generation of PWM signals

Bit data corresponding to the length of the PWM pulses has to be written to the compare registers MCMPk0 (sine side) and MCMPk1 (cosine side).

A timer counter is counting up. The rising edge of the PWM pulse is initiated at the overflow of the counter. The falling edge of the PWM pulse is initiated when the counter value equals the contents of the compare register.

The absolute pulse length in seconds is defined by the timer count clock (f_{MC0}). Various cycle times can be set via the timer mode control register MCNTC0.

Instruction

When writing data to compare registers, proceed as follows:

1. Confirm that MCMPCk.TEN = 0.
2. Write 8-bit PWM data to MCMPk0 and MCMPk1.
3. Set MCMPCk.ADB0 and MCMPCk.ADB1 as desired.
4. Set MCMPCk.TEN = 1 to start the counting operation.

   The data in MCMPk0/MCMPk1 will automatically be copied to the compare slave register when the counter overflows. The new pulse width is valid immediately.

   Bit MCMPCk.TEN is automatically cleared to 0 by hardware.

(3) Duty factor

The minimum pulse width that can be generated is zero (output signal is low) and the maximum pulse width is 255 clock cycles (maximum value of 8-bit compare registers).

The count range of the timer counter defines the duty factor. It can be set by bit MCNTC0.FULL:

- count range 01H to FFH (MCNTC0.FULL = 0)
  Formula for the duty cycle:
  PWM duty = MCMPki / 255 with k = 1 to 4 and i = 0, 1
  One count cycle is comprised of 255 clock cycles. A PWM signal with maximum pulse length is a steady high level signal. The duty factor is 100%.
- count range 00H to FFH (MCNTC0.FULL = 1)
  Formula for the duty cycle:
  PWM duty = MCMPki / 256 with k = 1 to 4 and i = 0, 1
  One count cycle is comprised of 256 clock cycles. A PWM signal with maximum pulse length is comprised of 255 clock cycles at high level and one clock cycle at low level. The duty factor is 255/256 *100% = 99.6%.

(4) Operation of 1-bit addition circuit

The precision of the angle of a needle is implicitly defined by the number of bits of the compare registers MCMPk0 and MCMPk1 (8 bit).

If the 1-bit addition circuit is enabled, every second pulse of the PWM signal is extended by one bit (one clock cycle). In average, a pulse width precision of 1/2 bit (1/2 clock) can be achieved.

The following figures show the timing of PWM output signals with 1-bit addition disabled and enabled.

Remarks 1. The PWM pulse is not generated until the first overflow occurs after the counting operation has been started.

2. The PWM signal is two cycle counts delayed compared to the overflow signal and the match signal. This is not depicted in the figures.

(5) Detecting zero points

For the detection of zero points, proceed as follows:

1. Set ZPDn pin to analog input by setting ZPDnEN = 0, select reference level using ZPDnS2 to ZPDnS0, and set the ZPDkPC bit in the ZPDSi register to 1.
2. Enable ZPD comparator operation by setting ZPDnEN = 1, and wait for comparator stabilization time.
3. ZPDn flag is cleared by setting the TWINk bit from 0 to 1 and detection operation starts.
4. Apply input signal to ZPDn pin.
5. Clear the TWINk bit to 0 when ZPD detection period is finished.

(6) Digital noise filter

The noise removal circuit suppresses short pulses/spikes of the comparator output to gain stable comparison results.

The minimum voltage comparator output pulse width to be validated is configurable by selecting the sampling clock for the digital noise removal, refer to CMPCTL.DBCL[3:0]. Spikes shorter than 2 sampling cycles are suppressed. Pulses longer than 3 sampling cycles are recognized as valid pulses. For pulses between 2 and 3 sampling cycles, the behavior is not defined.

Figure 15-9. Output Timing without 1-bit Addition

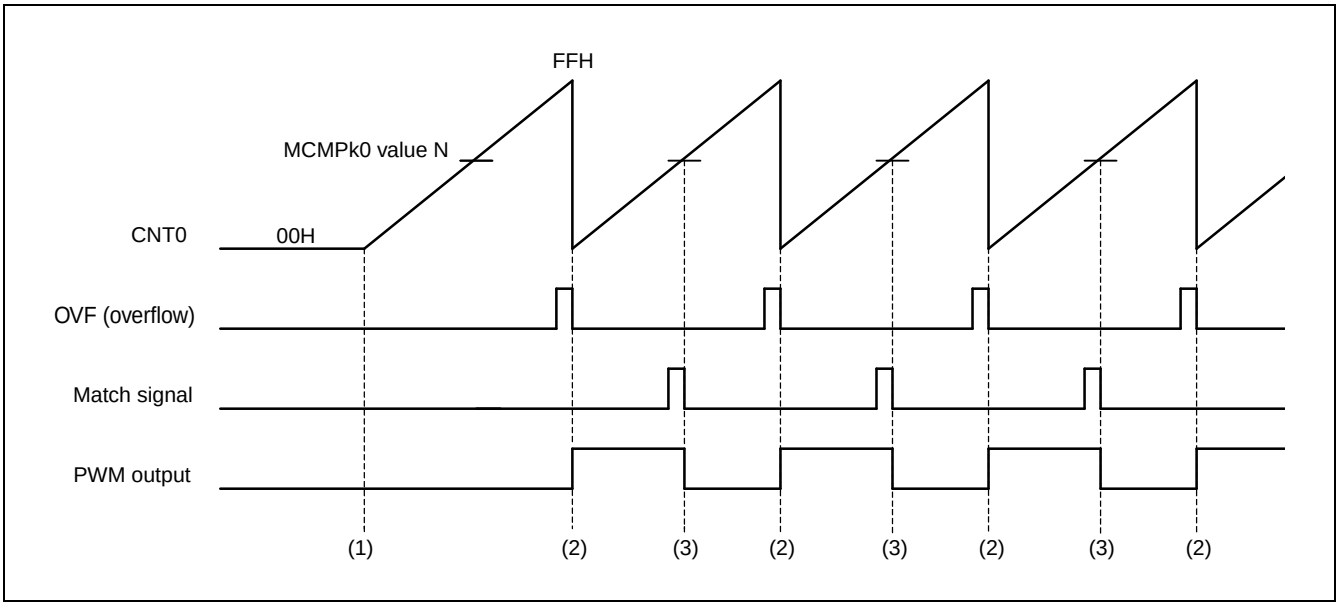
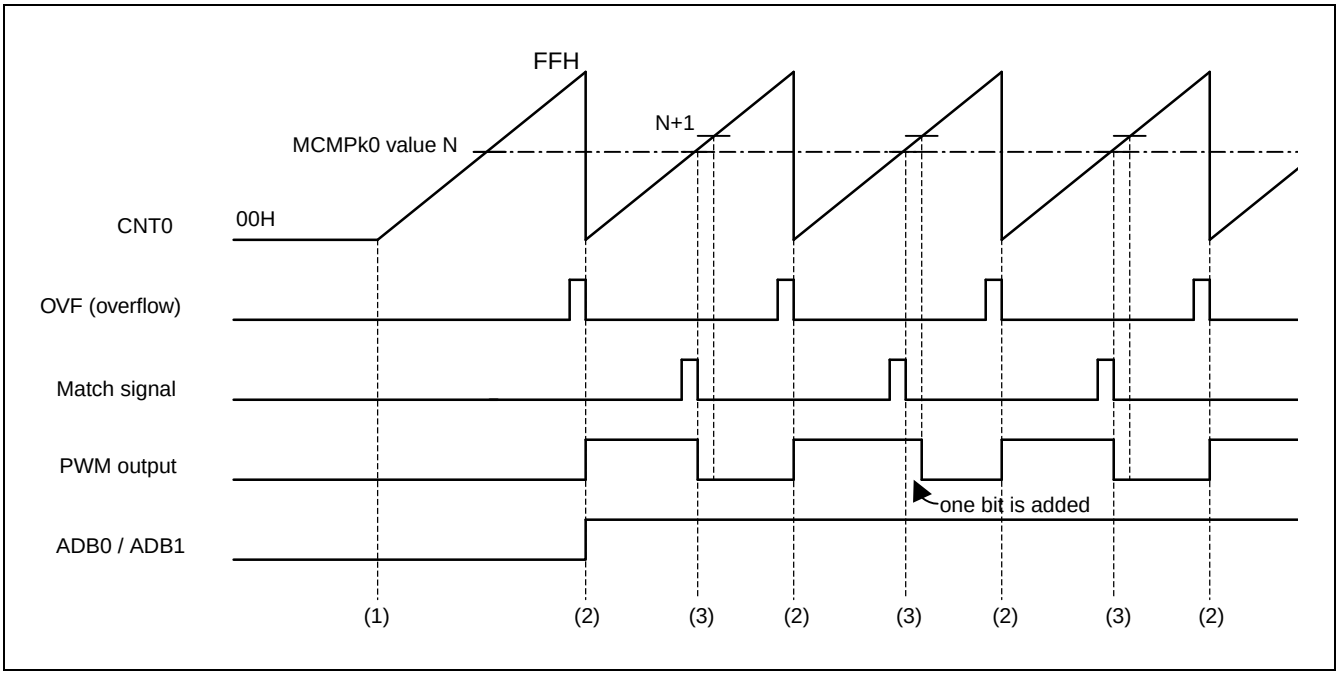


Figure 15-10. Output Timing with 1-bit Addition



Sequence

1. Start of counting (MCNTC0.PCE is set to 1)
2. Generation of overflow signal (start of PWM pulse)
3. Generation of match signal (timer counter CNT0 matches compare register, end of PWM pulse)

15.4 Timing

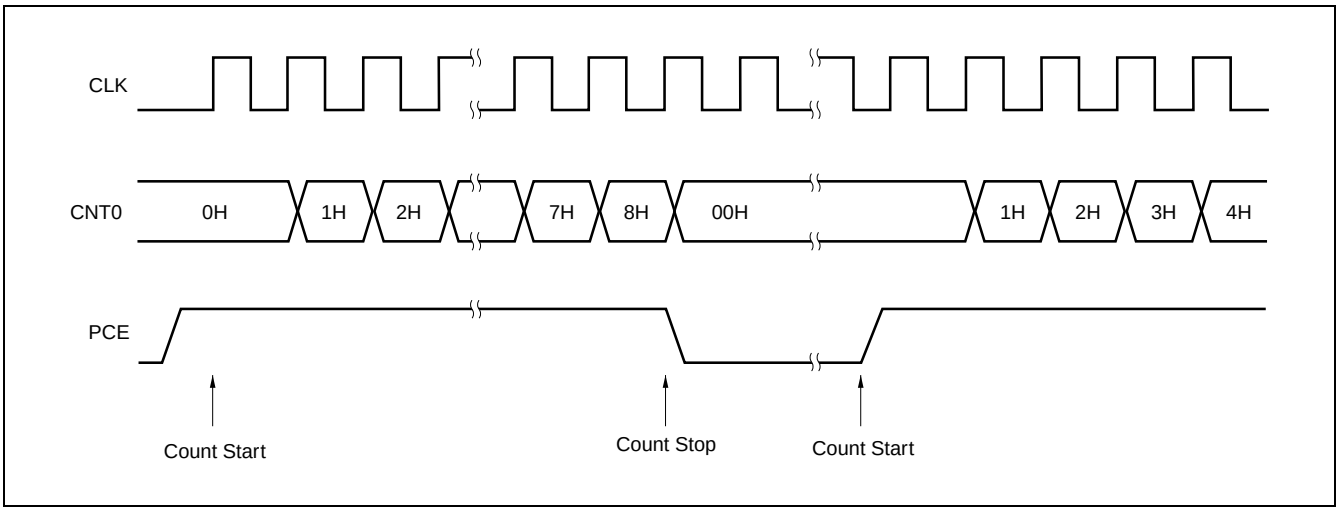
This section starts with the timing of the timer counter and general output timing behaviour. This section describes the timing at which the timer counter increments and when the count value is generally output. An example showing automatic phase shifting is also provided.

15.4.1 Timer counter

The free running up counter is clocked by the timer count clock selected in register MCNTC0.

The counting operation is enabled or disabled by the MCNTC0.PCE bit.

Figure 15-11. Restart Timing after Count Stop (Count Start—Count Stop—Count Start)



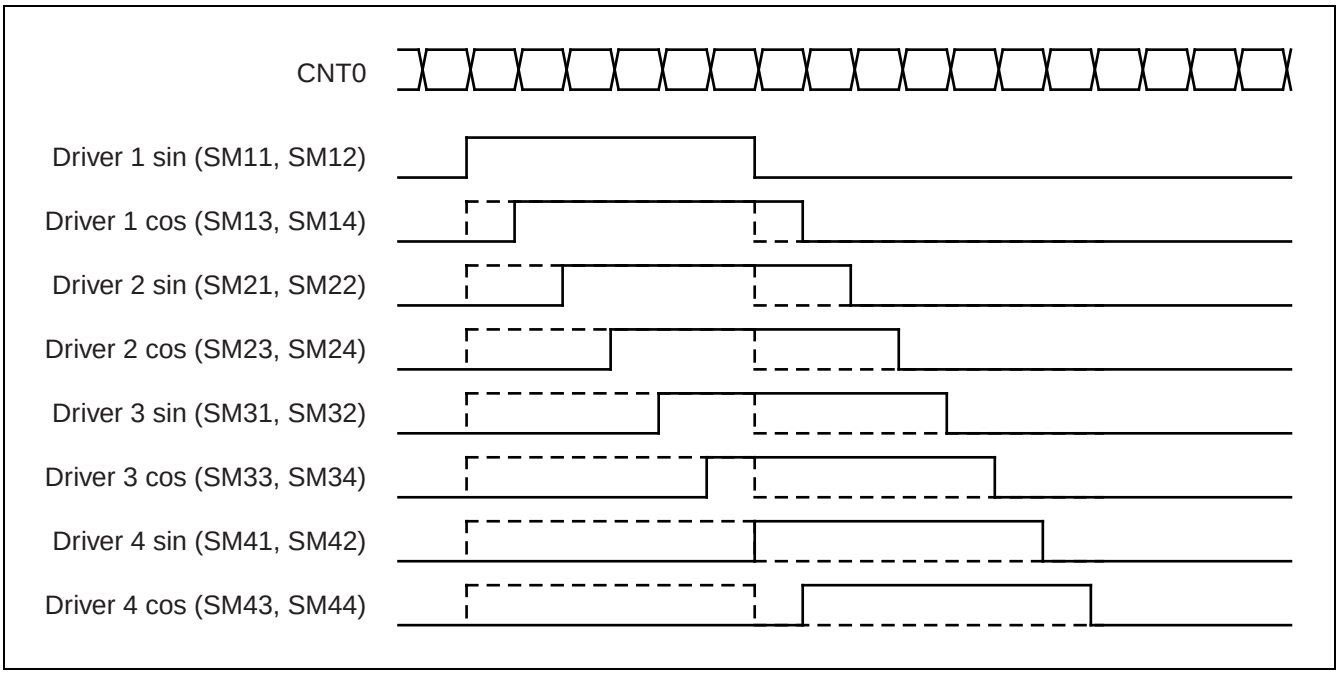
Sequence

- Count Start:
  - Enable counting operation (MCNTC0.PCE = 1)
  - Timer counter starts with value 00H. Depending on bit MCNTC0.FULL, all following counter cycles start with 00H or 01H, respectively.
- Count Stop:
  - Disable counting operation (MCNTC0.PCE = 0)
  - Counting is stopped and timer counter is set to 00H.

15.4.2 Automatic PWM phase shift

If the sine and cosine waveforms of meters 1 to 4 switch simultaneously as indicated by the dotted lines in **Figure 15-11**, the power supply might fluctuate, increasing susceptibility to electromagnetic interference. To prevent this, the signals output to drivers 1 to 4 are shifted one cycle of the timer count clock specified by the MCNTC0 register so that they are output at the timing indicated by the solid lines in **Figure 15-12**.

Figure 15-12. Output Timing of Signals SM11 to SM44



CHAPTER 16 LCD CONTROLLER/DRIVER

<R>

	48-pin	64-pin	80-pin	100-pin	128-pin
	R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
LCD (Seg x Com)	27 x 4	39 x 4	48 x 4	53 x 4	54 x 4

16.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the RL78/D1A are as follows.

(1) The LCD driver voltage generator uses internal resistance division method.
(2) Automatic output of segment and common signals based on automatic display data memory read
(3) Three different display modes:
- Static
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(4) Six different frame frequencies, selectable in each display mode
(5) R5F10CGx/R5F10DGx: Segment signal outputs: 27 (SEG0 to SEG7, SEG9, SEG14 to SEG15, SEG24 to SEG27, SEG29 to SEG35, SEG40 to SEG44)
Common signal outputs: 4 (COM0 to COM3)

R5F10CLx/R5F10DLx: Segment signal outputs: 39 (SEG0 to SEG9, SEG14 to SEG19, SEG21, SEG23 to SEG44),
Common signal outputs: 4 (COM0 to COM3)

R5F10CMx/R5F10DMx: Segment signal outputs: 48 (SEG0 to SEG47),
Common signal outputs: 4 (COM0 to COM3)

R5F10TPx/R5F10DPx: Segment signal outputs: 53 (SEG0 to SEG52),
Common signal outputs: 4 (COM0 to COM3)

<R> R5F10DSx: Segment signal outputs: 54 (SEG0 to SEG53)
Common signal outputs: 4 (COM0 to COM3)

Table 16-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 16-1. Maximum Number of Pixels

(a) R5F10CGx/R5F10DGx

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	27	27 (27 segment signals, 1 common signal)Note 1
	1/3	3	COM0 to COM2		81 (27 segment signals, 3 common signals)Note 2
		4	COM0 to COM3		108 (27 segment signals, 4 common signals)Note 3

Notes 1. 3-digit LCD panel, each digit having an 8-segment 8. configuration.
2. 9-digit LCD panel, each digit having a 3-segment 8. configuration.
3. 13-digit LCD panel, each digit having a 2-segment 8. configuration.

(b) R5F10CLx/R5F10DLx

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	39	39 (39 segment signals, 1 common signal)Note 1
	1/3	3	COM0 to COM2		117 (39 segment signals, 3 common signals)Note 2
		4	COM0 to COM3		156 (39 segment signals, 4 common signals)Note 3

Notes 1. 4-digit LCD panel, each digit having an 8-segment 8. configuration.
2. 13-digit LCD panel, each digit having a 3-segment 8. configuration.
3. 19-digit LCD panel, each digit having a 2-segment 8. configuration.

(c) R5F10CMx/R5F10DMx

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	48	48 (48 segment signals, 1 common signal)**Note 1**
	1/3	3	COM0 to COM2		144 (48 segment signals, 3 common signals)**Note 2**
		4	COM0 to COM3		192 (48 segment signals, 4 common signals)**Note 3**

Notes 1. 6-digit LCD panel, each digit having an 8-segment 8. configuration.
2. 16-digit LCD panel, each digit having a 3-segment 8. configuration.
3. 24-digit LCD panel, each digit having a 2-segment 8. configuration.

(d) R5F10TPx/R5F10DPx

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	53	53 (53 segment signals, 1 common signal)Note 1
	1/3	3	COM0 to COM2		159 (53 segment signals, 3 common signals)Note 2
		4	COM0 to COM3		212 (53 segment signals, 4 common signals)Note 3

Notes 1. 6-digit LCD panel, each digit having an 8-segment 8. configuration.
2. 17-digit LCD panel, each digit having a 3-segment 8. configuration.
3. 26-digit LCD panel, each digit having a 2-segment 8. configuration.

<R> **(e) R5F10DSx**

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	54	54 (54 segment signals, 1 common signal)Note 1
	1/3	3	COM0 to COM2		162 (54 segment signals, 3 common signals)Note 2
		4	COM0 to COM3		212 (54 segment signals, 4 common signals)Note 3

Notes 1. 6-digit LCD panel, each digit having an 8-segment 8. configuration.
2. 17-digit LCD panel, each digit having a 3-segment 8. configuration.
3. 26-digit LCD panel, each digit having a 2-segment 8. configuration.

16.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

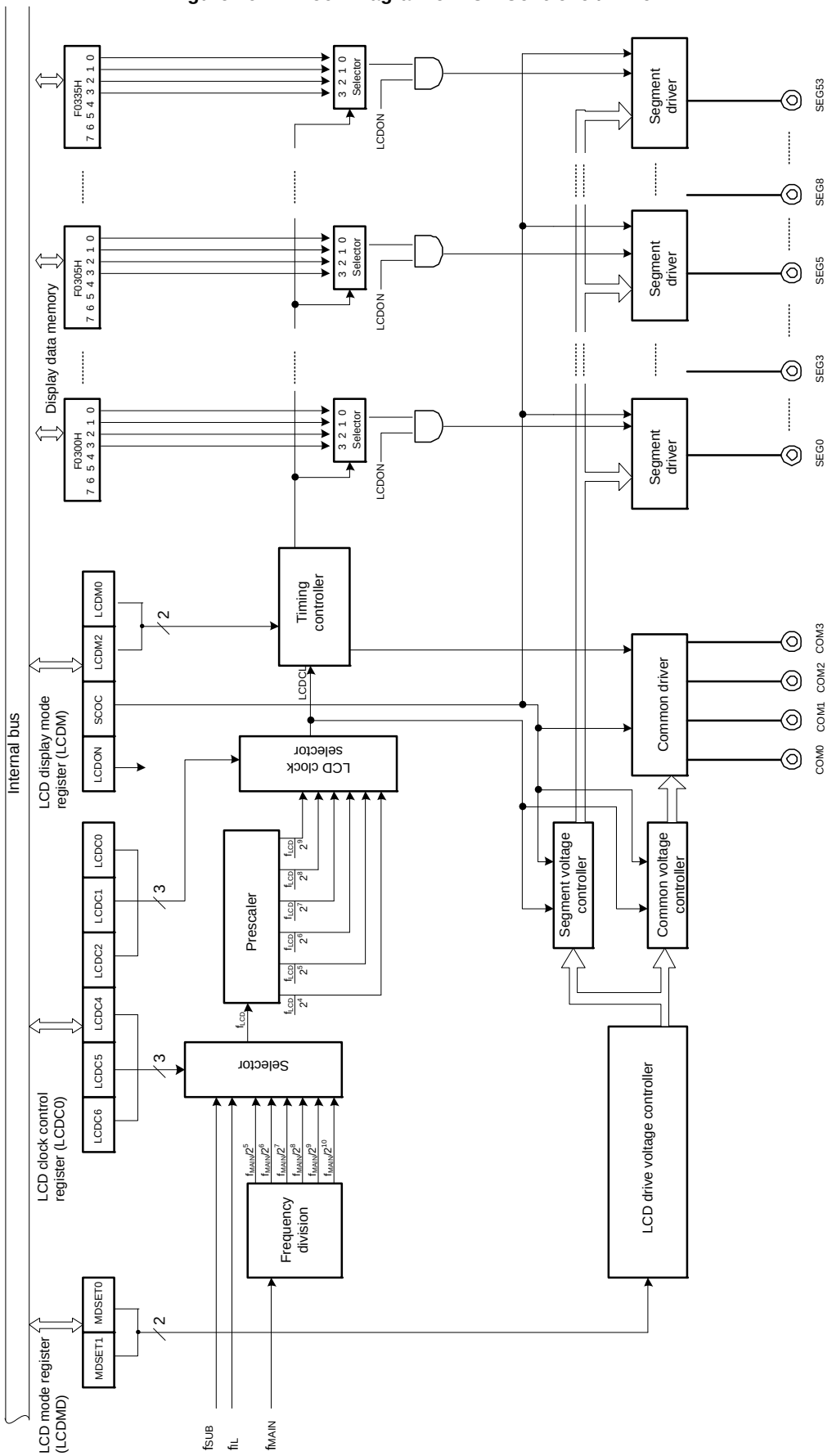
Table 16-2. Configuration of LCD Controller/Driver

<R>

Item	Configuration	
Display outputs	R5F10CGx/R5F10DGx	27 segment signals (SEG0 to SEG7, SEG9, SEG14 to SEG15, SEG24 to SEG27, SEG29 to SEG35, SEG40 to SEG44), 4 common signals (COM0 to COM3)
	R5F10CLx/R5F10DLx	39 segment signals (SEG0 to SEG9, SEG14 to SEG19, SEG21, SEG23 to SEG44), 4 common signals (COM0 to COM3)
	R5F10CMx/R5F10DMx	48 segment signals (SEG0 to SEG47), 4 common signals (COM0 to COM3)
	R5F10TPx/R5F10DPx	53 segment signals (SEG0 to SEG52), 4 common signals (COM0 to COM3)
	R5F10DSx	54 segment signals (SEG0 to SEG53), 4 common signals (COM0 to COM3)
Control registers	LCD mode register (LCDMD) LCD display mode register (LCDM) LCD clock control register (LCDC0) LCD port function registers 0, 1, 3, 5, 7 to 9, 13 (LCDPF0, LCDPF1, LCDPF3, LCDPF5, LCDPF7 to LCDPF9, LCDPF13)	

<R>

Figure 16-1. Block Diagram of LCD Controller/Driver



16.3 Registers Controlling LCD Controller/Driver

The following ten registers are used to control the LCD controller/driver.

- LCD mode register (LCDMD)
- LCD display mode register (LCDM)
- LCD clock control register (LCDC0)
- <R> LCD port function registers 0, 1, 3 to 5, 7 to 9, 13 (LCDPF0, LCDPF1, LCDPF3 to LCDPF5, LCDPF7 to LCDPF9, LCDPF13)

(1) LCD mode register (LCDMD)

LCDMD sets the LCD drive voltage generator.

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

Reset signal generation sets LCDMD to 00H.

Figure 16-2. Format of LCD Mode Register

Address: FFF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDMD	0	0	MDSET1	MDSET0	0	0	0	0

MDSET1	MDSET0	LCD drive voltage generator selection
0	0	No internal resistor connection (power save mode).
0	1	Internal resistance division method, internal resistor connection (no step-down transforming, Used when $V_{LCD} = V_{DD}$)
1	1	Internal resistance division method, internal resistor connection (step-down transforming, Used when $V_{LCD} = 3/5V_{DD}$)
Other than the above		Setting prohibited

Caution Bits 0 to 3, 6 and 7 must be set to 0.

(2) LCD display mode register (LCDM)

LCDM specifies whether to enable display operation. It also specifies whether to enable segment pin/common pin output, gate booster circuit control, and the display mode.

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

Reset signal generation sets LCDM to 00H.

Figure 16-3. Format of LCD Display Mode Register

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	0
LCDM	LCDON	SCOC	0	0	0	LCDM2	0	LCDM0

SCOC	LCDON	LCD display combination control
0	x	Output ground level to segment/common pins
1	0	Display off (output select level to common pins and deselete level to all segment pins)
1	1	Display on (output select level to both common pins and segment pins)

LCDM2	LCDM0	LCD controller/driver display mode selection	
		Resistance division method	
		Number of time slices	Bias mode
0	0	4	1/3
0	1	3	1/3
1	0	Static	
Other than the above		Setting prohibited	

Note When LCD display is not to be performed or not required, power consumption can be reduced by using the following settings.

<1> Set SCOC (bit 6 of the LCD display mode register (LCDM)) to 0.

<2> Set MDSET0 and MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) to 0.
(The current flowing to the internal resistors can be reduced.)

Cautions 1. Bits 1, 3 to 5 must be set to 0.

2. When displaying in a mode with a large number of COMs, such as 4 COM, V_{LC0} may not be able to obtain sufficient contrast under the low-voltage conditions, depending on the panel characteristics. Use the LCD controller/driver after having performed thorough LCD display evaluation and confirmed that there are no problems regarding the display quality.

(3) LCD clock control register (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

LCDC0 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDC0 to 00H.

Figure 16-4. Format of LCD Clock Control Register

Address: FFF42H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	LCDC6	LCDC5	LCDC4	0	LCDC2	LCDC1	LCDC0

LCDC6	LCDC5	LCDC4		LCD source clock (f_{LCD}) selection			
				f_{MAIN} = 4MHz	f_{MAIN} = 8MHz	f_{MAIN} = 20MHz	f_{MAIN} = 24MHz (f_{IH} only)
0	0	0	f_{XT}	32.768kHz			
0	0	1	$f_{MAIN}/2^5$	125kHz	250kHz	625kHz	750kHz
0	1	0	$f_{MAIN}/2^6$	62.5kHz	125kHz	312.5kHz	375kHz
0	1	1	$f_{MAIN}/2^7$	31.25kHz	62.5kHz	156.25kHz	187.5kHz
1	0	0	$f_{MAIN}/2^8$	15.625kHz	31.25kHz	78.125kHz	93.75kHz
1	0	1	$f_{MAIN}/2^9$	7.81kHz	15.625kHz	39.06kHz	46.875kHz
1	1	0	$f_{MAIN}/2^{10}$		7.81kHz	19.53kHz	23.43kHz
1	1	1	f_{IL}	15kHz (typ.)			

LCDC2	LCDC1	LCDC0		LCD(LCDCL) selection			
				f_{LCD} = 32.768kHz	f_{LCD} = 31.25kHz	f_{LCD} = 46.875kHz	f_{LCD} = 15kHz
0	0	0	$f_{LCD}/2^4$	2048Hz	1953.13Hz	2929.69Hz	937.5Hz
0	0	1	$f_{LCD}/2^5$	1024Hz	976.56Hz	1464.84Hz	468.75Hz
0	1	0	$f_{LCD}/2^6$	512Hz	488.28Hz	732.42Hz	234.38Hz
0	1	1	$f_{LCD}/2^7$	256Hz	244.14Hz	366.21Hz	117.19Hz
1	0	0	$f_{LCD}/2^8$	128Hz	122.07Hz	183.1Hz	58.59Hz
1	0	1	$f_{LCD}/2^9$	64Hz	61.04Hz	91.55Hz	29.29Hz
Other than the above			Setting prohibited (same as “000” setting)				

Caution Bits 3 and 7 must be set to 0.

Remarks 1. f_{XT}: XT1 clock oscillation frequency

2. f_{IL}: Low-speed on-chip oscillator clock frequency

LCD frame frequency (Hz) example :

Duty	LCDCL = 64Hz (T = 15.6ms)	LCDCL = 128Hz (T = 7.8ms)	LCDCL = 256Hz (T = 3.9ms)	LCDCL = 122.07Hz (T = 8.192ms)	LCDCL = 244.14Hz (T = 4.096ms)	LCDCL = 234.38Hz (T = 4.27ms)
1/3	21	43	85	40	81	78
1/4	16	32	64	30	61	58
Static	64	128	256	122	244	234

Remark Frame period TF = 3 ×T when 1/3 duty mode
TF = 4 ×T when 1/4 duty mode
TF = T when static mode

(4) LCD port function register 0 (LCDPF0)

This register sets whether to use pins P00 to P07 as port pins (other than segment output pins) or segment output pins.

LCDPF0 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF0 to 00H.

Figure 16-5. Format of LCD Port Function Register 0

Address: F0050H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF0	LCDPF07	LCDPF06	LCDPF05	LCDPF04	LCDPF03	LCDPF02	LCDPF01	LCDPF00

LCDPF0n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

(5) LCD port function register 1 (LCDPF1)

This register sets whether to use pins P10 to P17 as port pins (other than segment output pins) or segment output pins.

LCDPF1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF1 to 00H.

Figure 16-6. Format of LCD Port Function Register 1

Address: F0051H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF1	LCDPF17	LCDPF16	LCDPF15	LCDPF14	LCDPF13	LCDPF12	LCDPF11	LCDPF10

LCDPF1n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

(6) LCD port function register 3 (LCDPF3)

This register sets whether to use pins P30 to P37 as port pins (other than segment output pins) or segment output pins.

LCDPF3 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF3 to 00H.

Figure 16-7. Format of LCD Port Function Register 3

Address: F0053H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF3	LCDPF37	LCDPF36	LCDPF35	LCDPF34	LCDPF33	LCDPF32	LCDPF31	LCDPF30

LCDPF3n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

<R> **(7) LCD port function register 4 (LCDPF4)**

This register sets whether to use pins P42 to P47 of 128-pin products as port pins (other than segment output pins) or segment output pins.

LCDPF3 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF3 to 00H.

Figure 16-8. Format of LCD Port Function Register 3

Address: F0054H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF4	LCDPF47	LCDPF46	LCDPF45	LCDPF44	LCDPF43	LCDPF2	0	0

LCDPF4n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 2 to 7

(8) LCD port function register 5 (LCDPF5)

This register sets whether to use pins P50 to P57 as port pins (other than segment output pins) or segment output pins.

LCDPF5 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF5 to 00H.

Figure 16-9. Format of LCD Port Function Register 5

Address: F0055H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF5	LCDPF57	LCDPF56	LCDPF55	LCDPF54	LCDPF53	LCDPF52	LCDPF51	LCDPF50

LCDPF5n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

(9) LCD port function register 7 (LCDPF7)

This register sets whether to use pins P72 to P75 as port pins (other than segment output pins) or segment output pins.

LCDPF7 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF7 to 00H.

Figure 16-10. Format of LCD Port Function Register 7

Address: F0057H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF7	0	0	LCDPF75	LCDPF74	LCDPF73	LCDPF72	0	0

LCDPF7n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

(10) LCD port function register 8 (LCDPF8)

This register sets whether to use pins P80 to P87 as port pins (other than segment output pins) or segment output pins.

LCDPF8 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF8 to 00H.

Figure 16-11. Format of LCD Port Function Register 8

Address: F0058H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF8	LCDPF87	LCDPF86	LCDPF85	LCDPF84	LCDPF83	LCDPF82	LCDPF81	LCDPF80

LCDPF8n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

(11) LCD port function register 9 (LCDPF9)

This register sets whether to use pins P90 to P97 as port pins (other than segment output pins) or segment output pins.

LCDPF9 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF9 to 00H.

Figure 16-12. Format of LCD Port Function Register 9

Address: F0059H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF9	LCDPF97	LCDPF96	LCDPF95	LCDPF94	LCDPF93	LCDPF92	LCDPF91	LCDPF90

LCDPF9n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

(12) LCD port function register 13 (LCDPF13)

This register sets whether to use pins P136 as port pins (other than segment output pins) or segment output pins.

LCDPF13 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF13 to 00H.

Figure 16-13. Format of LCD Port Function Register 13

Address: F005DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF13	0	LCDPF136	0	0	0	0	0	0

LCDPF13n	Port/segment output specification
0	Used as port or alternate function other than segment output
1	Used as segment output

Remark n = 0 to 7

16.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure:

<1> Set the LCD drive method via MDSET0 and MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)).
<R> <2> Set the pins to be used as segment outputs to the port function registers (LCDPF0, LCDPF1, LCDPF3 to LCDPF5, LCDPF7 to LCDPF9, LCDPF13).
<3> Set an initial value to the RAM for LCD display.
<4> Set the number of time slices via LCDM0 to LCDM2 (bits 0 to 2 of the LCD display mode register (LCDM)).
<5> Set the LCD source clock and LCD clock via LCD clock control register (LCDC0).
<6> Set SCOC (bit 6 of the LCD display mode register (LCDM)) to 1.
<7> Start output corresponding to each data memory by setting LCDON (bit 7 of the LCD display mode register (LCDM)) to 1.

Subsequent to this procedure, set the data to be displayed in the data memory.

Remark Use the following procedure to set to the display-off state and disconnect the internal resistors when using the internal resistance division method.
<1> Clear LCDON (bit 7 of LCDM) (LCDON = 0).
Deselect signals are output from all segment pins and common pins, and a non-display state is entered.
<2> Clear SCOC (bit 6 of the LCD display mode register (LCDM)) (SCOC = 0).
Ground levels are output from all segment pins and common pins.
<3> Assume MDSET0, MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) = (0, 0) and set to no internal resistor connection (power save mode).

Caution When displaying in a mode with a large number of COMs, such as 4 COM, V_{LC0} may not be able to obtain sufficient contrast under the low-voltage conditions, depending on the panel characteristics. Use the LCD controller/driver after having performed thorough LCD display evaluation and confirmed that there are no problems regarding the display quality.

16.5 LCD Display Data Memory

The LCD display data memory is mapped at the following addresses:

<R> F0300H to F0334H for R5F10TPx, R5F10DPx, and R5F10DSx

F0300H to F032FH for R5F10CMx and R5F10DMx

F0300H to F032CH for R5F10CLx, R5F10DLx, R5F10CGx and R5F10DGx

Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 16-13 shows the relationship between the contents of the LCD display data memory and the segment/common outputs.

The areas not to be used for display can be used as normal RAM.

Figure 16-14. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs (1/3)
(a) R5F10CGx/R5F10DGx

	b7	b6	b5	b4	b3	b2	b1	b0	
F032CH	0	0	0	0					SEG44
F032BH	0	0	0	0					SEG43
F032AH	0	0	0	0					SEG42
⋮									
F0305H	0	0	0	0					SEG5
F0304H	0	0	0	0					SEG4
F0303H	0	0	0	0					SEG3
F0302H	0	0	0	0					SEG2
F0301H	0	0	0	0					SEG1
F0300H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

Note R5F10CGx/R5F10DGx has 27 segment signals (SEG0 to SEG7, SEG9, SEG14 to SEG15, SEG24 to SEG27, SEG29 to SEG35, SEG40 to SEG44).
Thus, SEG8 (F0308H), SEG10 to SEG13 (F030AH to F030DH), SEG16 to SEG23 (F0310H to F0317H), SEG28 (F031CH) and SEG36 to SEG39 (F0324H to F0327H) are not existed.

Caution No memory is allocated to the higher 4 bits. Be sure to set these bits to 0.

Figure 16-14. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs (2/3)

(b) R5F10CLx/R5F10DLx

	b7	b6	b5	b4	b3	b2	b1	b0	
F032CH	0	0	0	0					SEG44
F032BH	0	0	0	0					SEG43
F032AH	0	0	0	0					SEG42
F0305H	0	0	0	0					SEG5
F0304H	0	0	0	0					SEG4
F0303H	0	0	0	0					SEG3
F0302H	0	0	0	0					SEG2
F0301H	0	0	0	0					SEG1
F0300H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

<R> **Note** R5F10CLx/R5F10DLx has 39 segment signals (SEG0 to SEG9, SEG14 to SEG19, SEG21, SEG23 to SEG44).
Thus, SEG10 to SEG13 (F030AH to F030DH), SEG20 (F0314H), and SEG22 (F0316H) are not existed.

Caution No memory is allocated to the higher 4 bits. Be sure to set these bits to 0.

(c) R5F10CMx/R5F10DMx

	b7	b6	b5	b4	b3	b2	b1	b0	
F032FH	0	0	0	0					SEG47
F032EH	0	0	0	0					SEG46
F032DH	0	0	0	0					SEG45
F0305H	0	0	0	0					SEG5
F0304H	0	0	0	0					SEG4
F0303H	0	0	0	0					SEG3
F0302H	0	0	0	0					SEG2
F0301H	0	0	0	0					SEG1
F0300H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

<R> **Note** R5F10CMx/R5F10DMx have 48 segment signals (SEG0 to SEG47).

Caution No memory is allocated to the higher 4 bits. Be sure to set these bits to 0.

Figure 16-14. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs (2/3)

(d) R5F10TPx/R5F10DPx

	b7	b6	b5	b4	b3	b2	b1	b0	
F0334H	0	0	0	0					SEG52
F0333H	0	0	0	0					SEG51
F0332H	0	0	0	0					SEG50
F0305H	0	0	0	0					SEG5
F0304H	0	0	0	0					SEG4
F0303H	0	0	0	0					SEG3
F0302H	0	0	0	0					SEG2
F0301H	0	0	0	0					SEG1
F0300H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

<R> **Note** R5F10TPx/R5F10DPx has 53 segment signals (SEG0 to SEG52).

Caution No memory is allocated to the higher 4 bits. Be sure to set these bits to 0.

<R> **(e) R5F10DSx**

	b7	b6	b5	b4	b3	b2	b1	b0	
F0334H	0	0	0	0					SEG53
F0333H	0	0	0	0					SEG52
F0332H	0	0	0	0					SEG51
F0305H	0	0	0	0					SEG5
F0304H	0	0	0	0					SEG4
F0303H	0	0	0	0					SEG3
F0302H	0	0	0	0					SEG2
F0301H	0	0	0	0					SEG1
F0300H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

Note R5F10DSx has 54 segment signals (SEG0 to SEG53).

Caution No memory is allocated to the higher 4 bits. Be sure to set these bits to 0.

16.6 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). The pixels turn off when the potential difference becomes lower than V_{LCD}.

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in **Table 16-3**. In the static display mode, the same signal is output to COM0 to COM3.

In the three-time-slice mode, leave the COM3 pin open.

Table 16-3. COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3
Static display mode				
Three-time-slice mode				Open
Four-time-slice mode				

(2) Segment signals

(a) R5F10CGx/R5F10DGx

The segment signals correspond to 27 bytes of the LCD display data memory (F0300H to F0307H, F0309H, F030EH, F030FH, F0318H to F031BH, F031DH to F0323H, F0328H to F032CH) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG7, SEG9, SEG14 to SEG15, SEG24 to SEG27, SEG29 to SEG35, SEG40 to SEG44).

(b) R5F10CLx/R5F10DLx

The segment signals correspond to 39 bytes of the LCD display data memory (F0300H to F0309H, F030EH to F0313H, F0315H, F0317H to F032CH) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG9, SEG14 to SEG19, SEG21, SEG23 to SEG44).

(c) R5F10CMx/R5F10DMx

The segment signals correspond to 48 bytes of the LCD display data memory (F0300H to F032FH) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG47).

(d) R5F10TPx/R5F10DPx

The segment signals correspond to 53 bytes of the LCD display data memory (F0300H to F0334H) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG52).

<R> **(e) R5F10DSx**

The segment signals correspond to 54 bytes of the LCD display data memory (F0300H to F0335H) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG53).

Check, with the information given above, what combination of front-surface electrodes (corresponding to the segment signals) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data memory, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

LCD display data memory bits 1 to 3 are not used for LCD display in the static display, and bit 3 is not used for three-time slot mode, respectively. So these bits can be used for purposes other than display.

<R> The higher 4 bits of “F0300H to F0335H” are fixed to 0.

(3) Output waveforms of common signals and segment signals during LCD display signal output period

The voltages shown in **Table 16-4** are output to the common signals and segment signals during the LCD display signal output period.

When both common and segment signals are at the select voltage, a display on-voltage of $\pm V_{LCD}$ is obtained. The other combinations of the signals correspond to the display off-voltage.

Table 16-4. LCD Drive Voltage

(a) Static display mode (during LCD display signal output period)

Common Signal \ Segment Signal	Select Signal Level	Deselect Signal Level
	V_{SS}/V_{LC0}	V_{LC0}/V_{SS}
V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

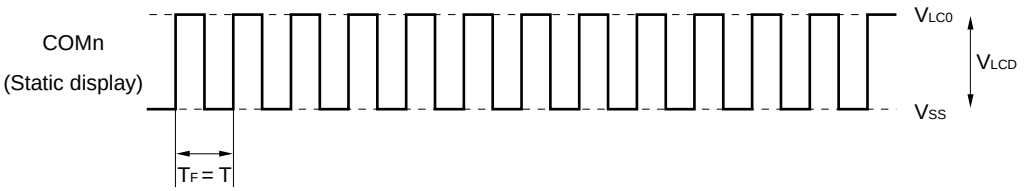
(b) 1/3 bias method (during LCD display signal output period)

Common Signal \ Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{LC0}	V_{LC1}/V_{LC2}
Select signal level	V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$
Deselect signal level	V_{LC2}/V_{LC1}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

Figure 16-15 shows the common signal waveforms, and **Figure 16-16** shows the voltages and phases of the common and segment signals.

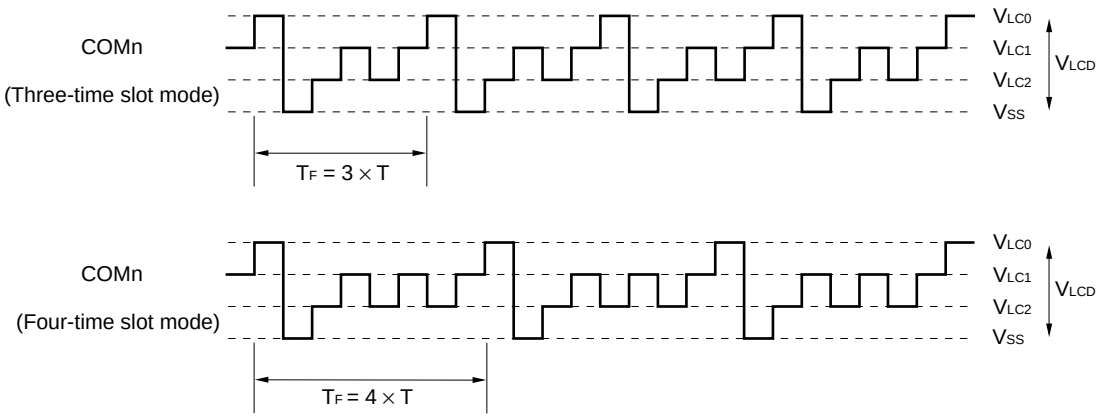
Figure 16-15. Common Signal Waveforms

(a) Static display mode



T: One LCD clock period T_F: Frame frequency

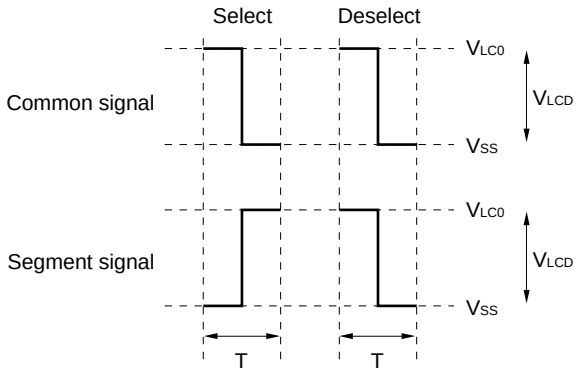
(b) 1/3 bias method



T: One LCD clock period T_F: Frame frequency

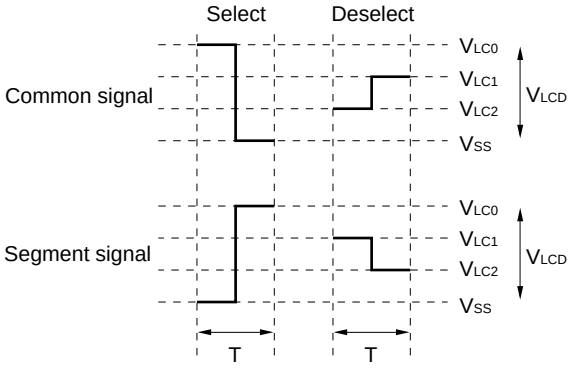
Figure 16-16. Voltages and Phases of Common and Segment Signals

(a) Static display mode



T: One LCD clock period

(b) 1/3 bias method



T: One LCD clock period

16.7 Display Modes

16.7.1 Static display example

Figure 16-18 shows how the three-digit LCD panel having the display pattern shown in **Figure 16-17** is connected to the segment signals (SEG0 to SEG23) and the common signal (COM0) of the RL78/D1A chip. This example displays data "12.3" in the LCD panel. The contents of the display data memory (F0300H to F0317H) correspond to this display.

The following description focuses on numeral "2." (2.) displayed in the second digit. To display "2." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG8 to SEG15 pins according to **Table 16-5** at the timing of the common signal COM0; see **Figure 16-17** for the relationship between the segment signals and LCD segments.

Table 16-5. Select and Deselect Voltages (COM0)

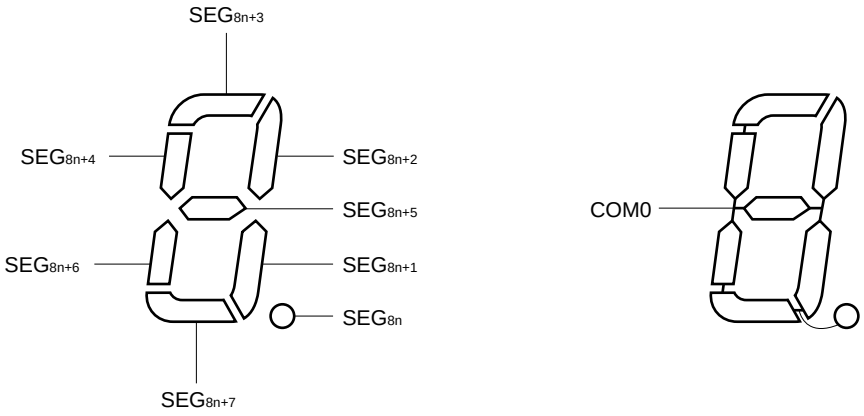
Segment / Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to **Table 16-5**, it is determined that the bit-0 pattern of the display data memory locations (F0308H to F030FH) must be 10110111.

Figure 16-19 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 16-17. Static LCD Display Pattern and Electrode Connections



Remark n = 0 to 2

Figure 16-18. Example of Connecting Static LCD Panel

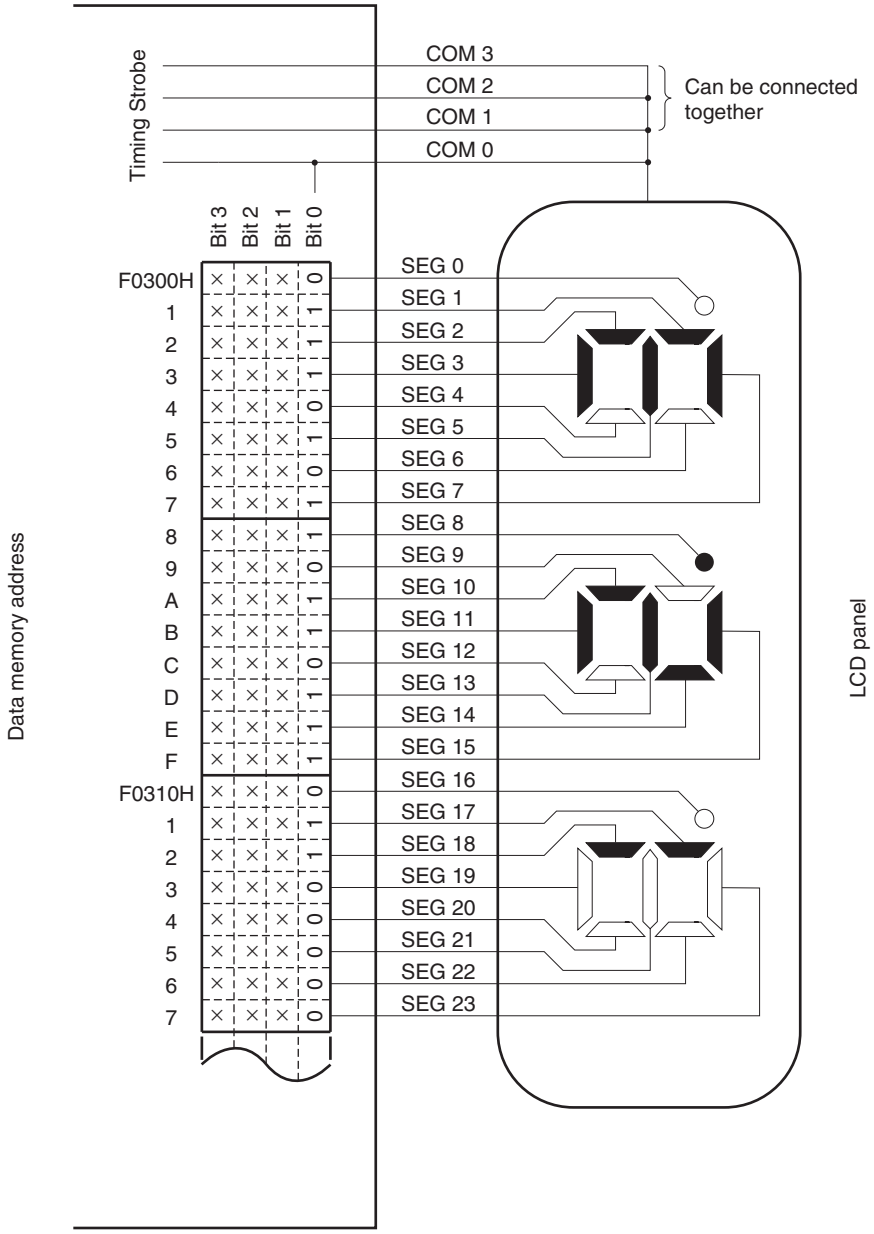
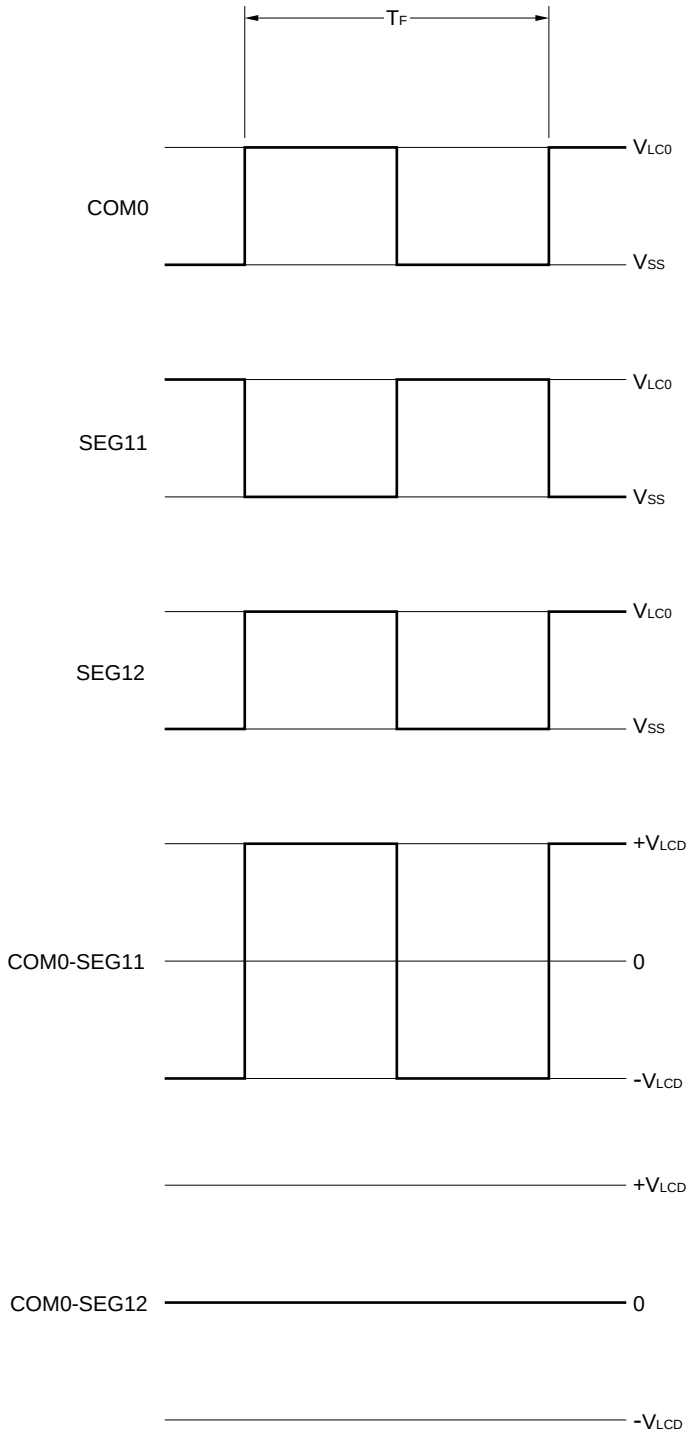


Figure 16-19. Static LCD Drive Waveform Examples



16.7.2 Three-time-slice display example

Figure 16-21 shows how the 8-digit LCD panel having the display pattern shown in **Figure 16-20** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM2) of the RL78/D1A chip. This example displays data "123456.78" in the LCD panel. The contents of the display data memory (addresses F0300H to F0317H) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the third digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG6 to SEG8 pins according to **Table 16-6** at the timing of the common signals COM0 to COM2; see **Figure 16-20** for the relationship between the segment signals and LCD segments.

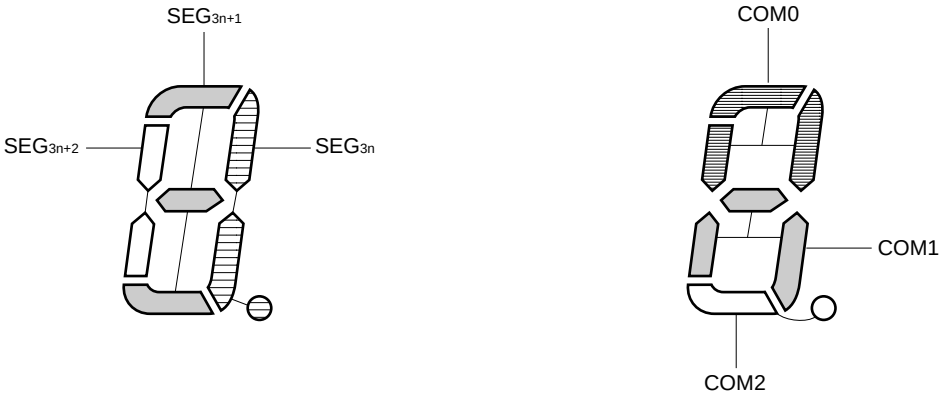
Table 16-6. Select and Deselect Voltages (COM0 to COM2)

Segment / Common	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to **Table 16-6**, it is determined that the display data memory location (F0306H) that corresponds to SEG6 must contain x110.

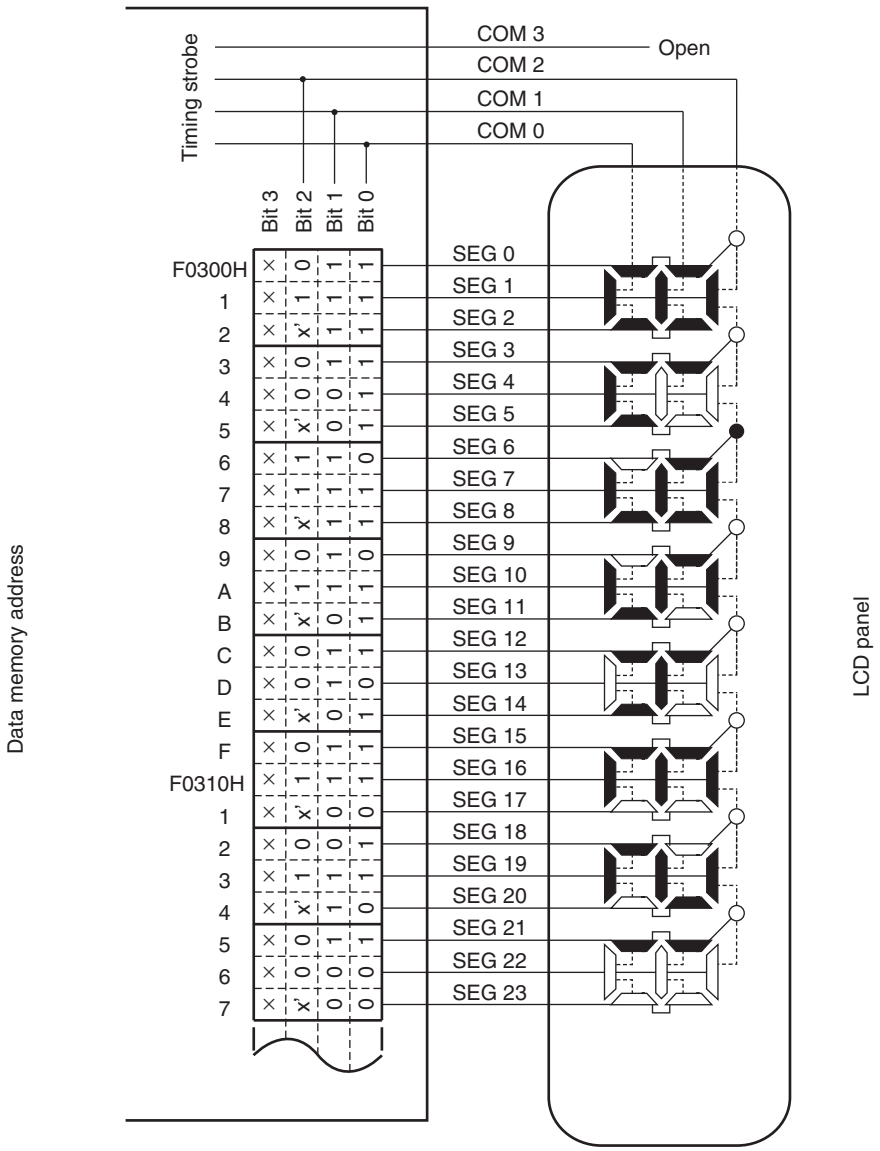
Figure 16-22 shows an example of LCD drive waveforms between the SEG6 signal and each common signal in the 1/3 bias method. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 16-20. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 7

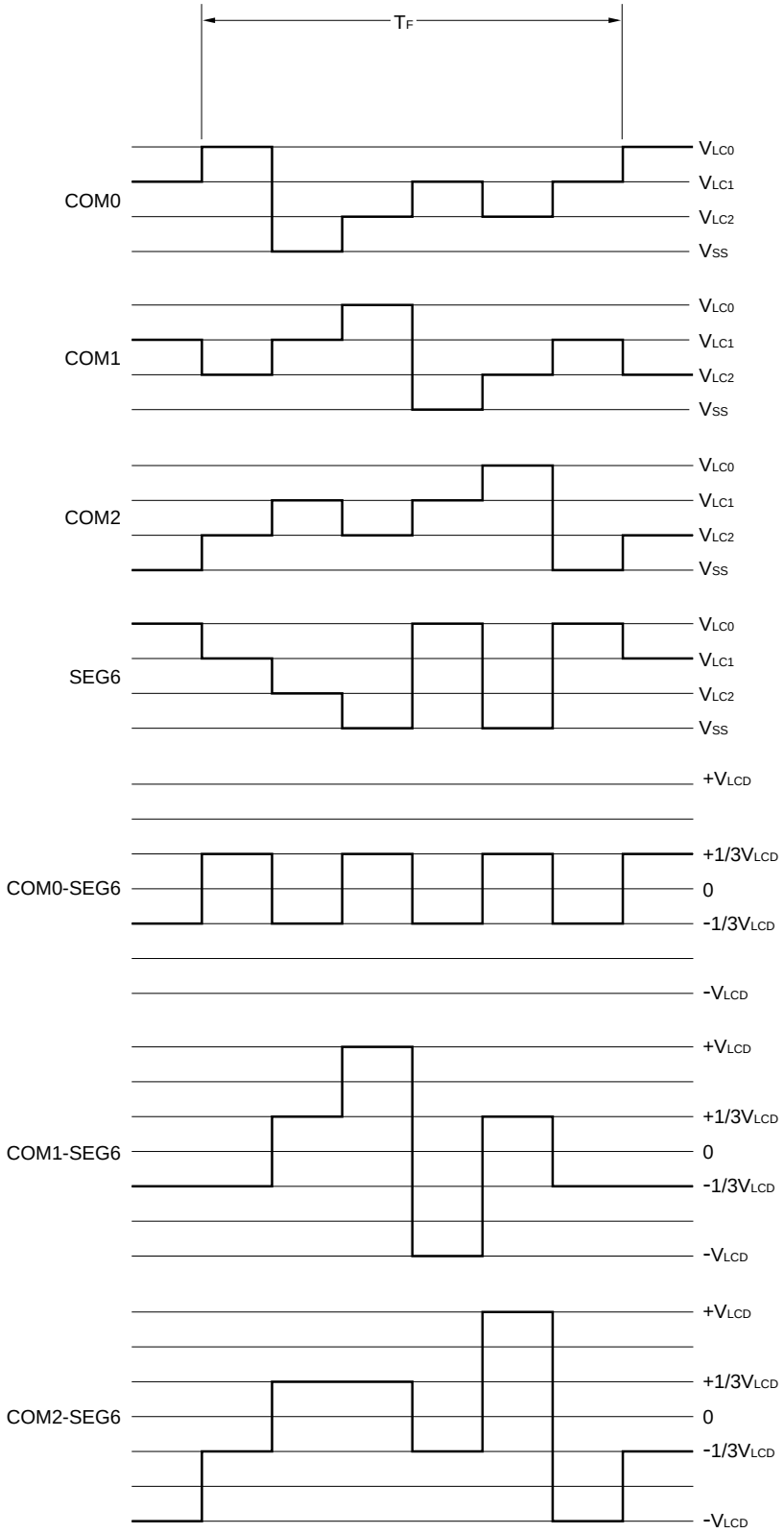
Figure 16-21. Example of Connecting Three-Time-Slice LCD Panel



×': Can be used to store any data because there is no corresponding segment in the LCD panel.

×: Can always be used to store any data because the three-time-slice mode is being used.

Figure 16-22. Three-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



16.7.3 Four-time-slice display example

Figure 16-24 shows how the 12-digit LCD panel having the display pattern shown in **Figure 16-23** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM3) of the RL78/D1A chip. This example displays data "123456.789012" in the LCD panel. The contents of the display data memory (addresses F0300H to F0317H) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the seventh digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 and SEG13 pins according to **Table 16-7** at the timing of the common signals COM0 to COM3; see **Figure 16-23** for the relationship between the segment signals and LCD segments.

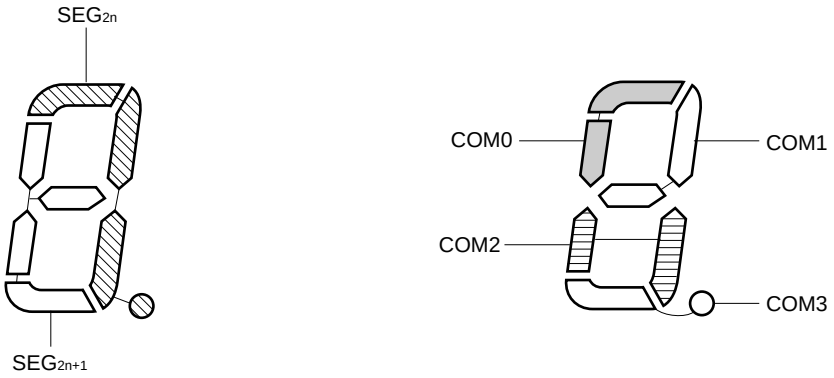
Table 16-7. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to **Table 16-7**, it is determined that the display data memory location (F030CH) that corresponds to SEG12 must contain 1101.

Figure 16-25 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 16-23. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 11

Figure 16-24. Example of Connecting Four-Time-Slice LCD Panel

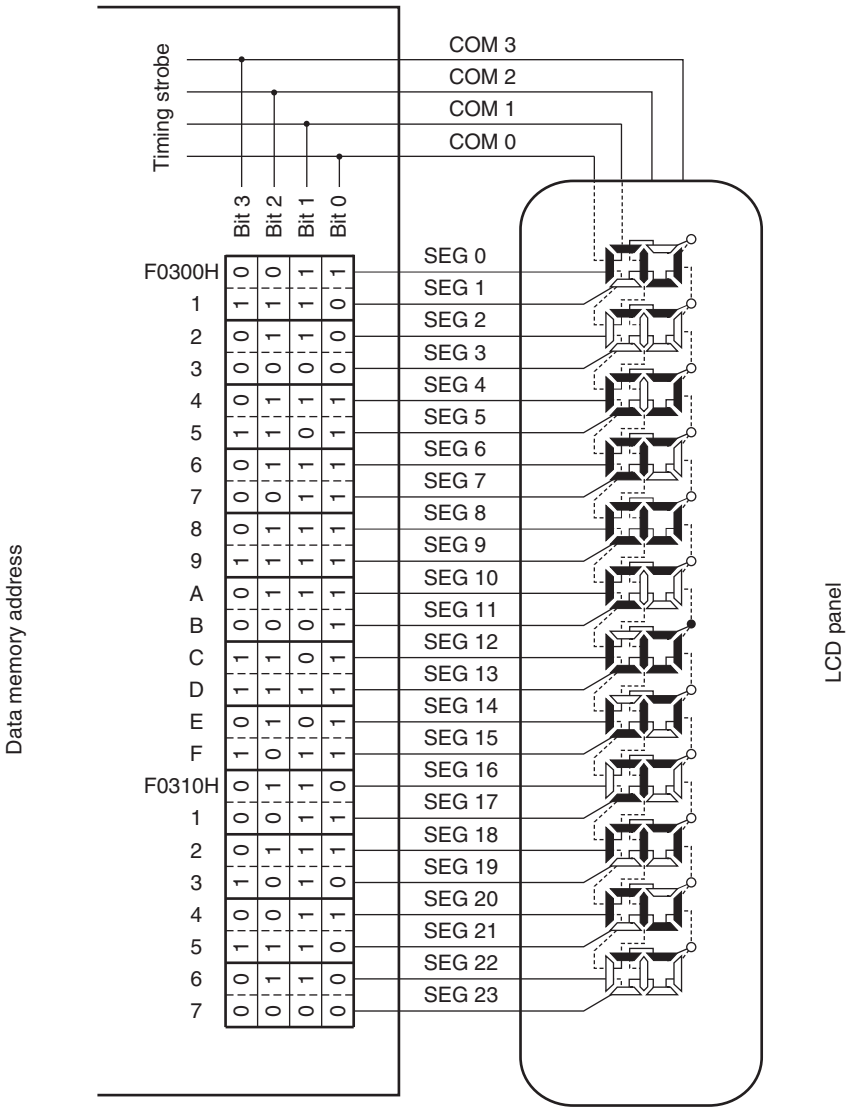
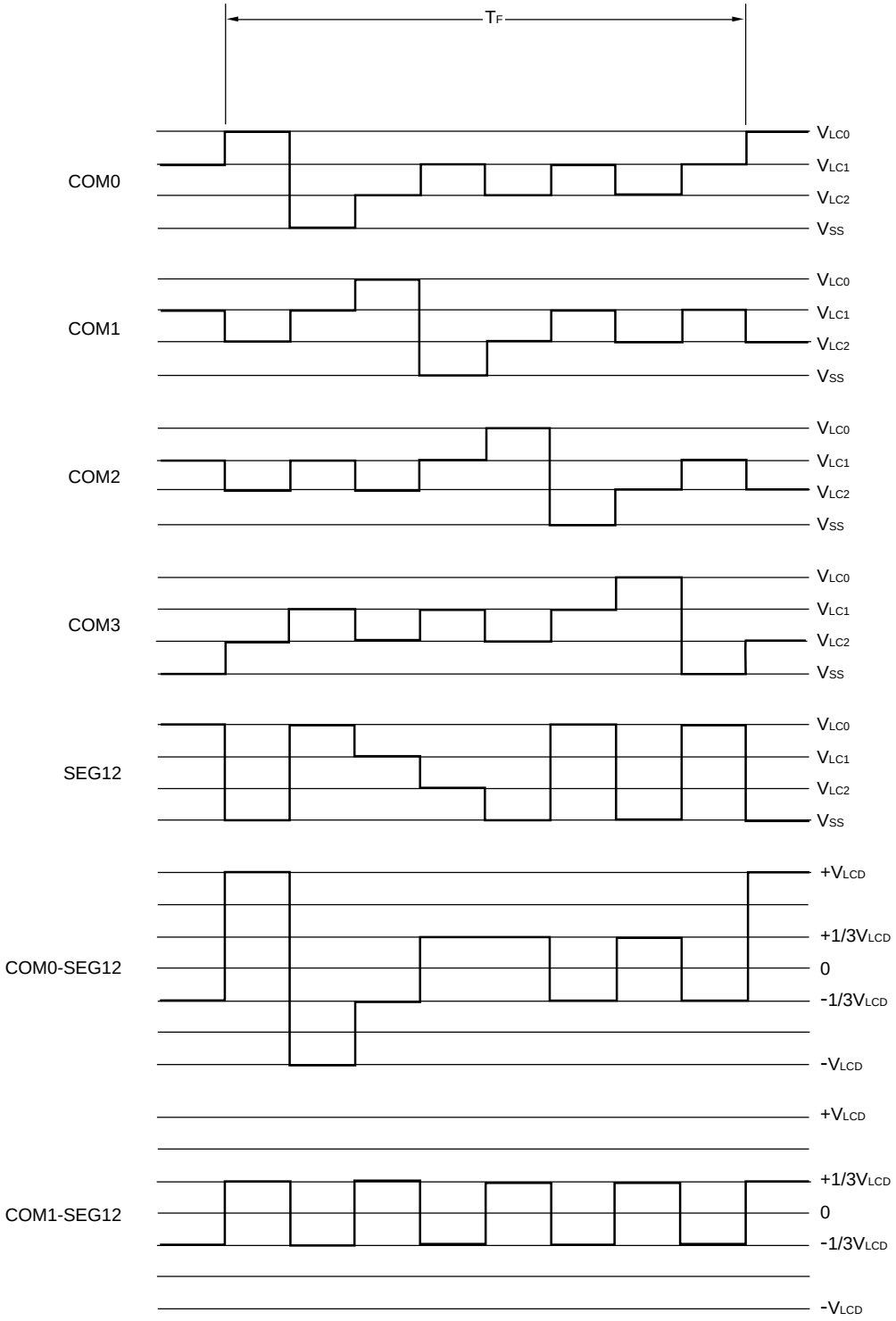


Figure 16-25. Four-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



Remark The waveforms for COM2 to SEG12 and COM3 to SEG12 are omitted.

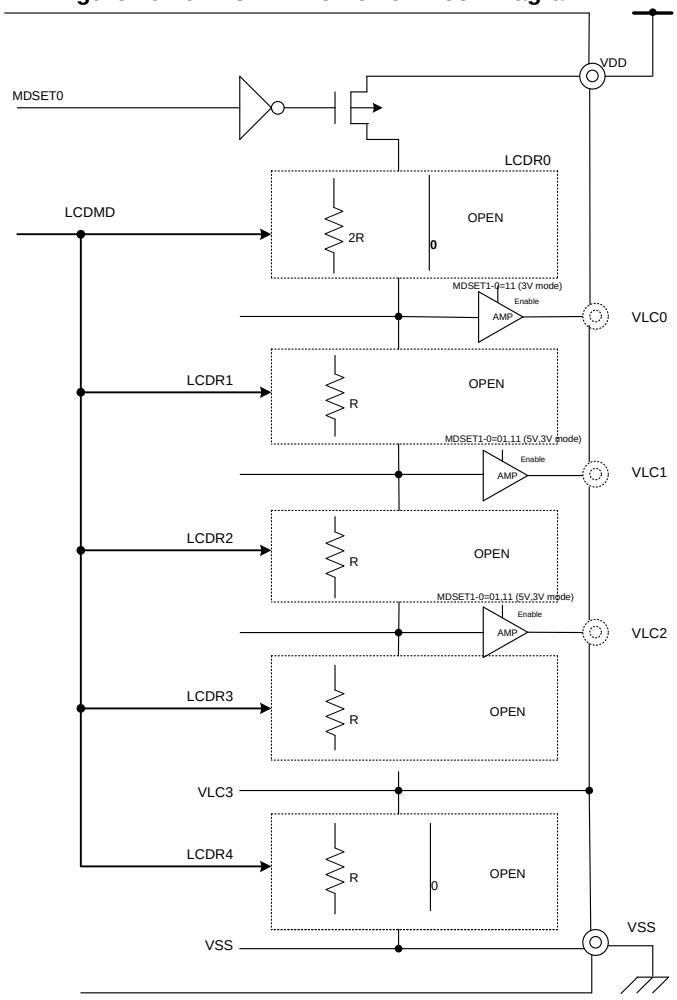
16.8 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, and V_{LC2}

The AMP for VLC0 is only enabled at 3V mode.

Remark: 3V mode (step-down transforming, MDSET1-0=11)

5V mode (no step-down transforming, MDSET1-0=01)

Figure 16-26. LCD Drive Power Block Diagram



Note There is no VLCx pins in this product.

mode		LCDR0	LCDR1	LCDR2	LCDR3	LCDR4
Static or or 1/3 bias mode	no step-down transforming	R	R	R	R	0**
	step-down transforming	2R	R	R	R*	0**

With the RL78/D1A, a LCD drive power supply is generated using internal resistance division method.

The RL78/D1A incorporates voltage divider resistors for generating LCD drive power supplies. Using internal voltage divider resistors, a LCD drive power supply that meet each bias method listed in **Table 16-8** can be generated, without using external voltage divider resistors.

Table 16-8. LCD Drive Voltages (with On-Chip Voltage Divider Resistors)

Bias Method / LCD Drive Voltage Pin	No Bias (Static)	1/3 Bias Method
V_{LC0}	V_{LCD}	V_{LCD}
V_{LC1}	$\frac{2}{3}V_{LCD}$	$\frac{2}{3}V_{LCD}$
V_{LC2}	$\frac{1}{3}V_{LCD}$	$\frac{1}{3}V_{LCD}$
V_{LC3}	V_{SS}	V_{SS}

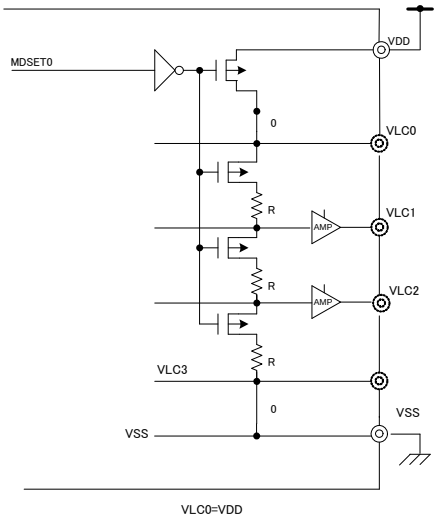
Figure 16-26 shows examples of generating LCD drive voltages internally according to **Table 16-8**.

Table 16-9. Truth Table of LCD Mode

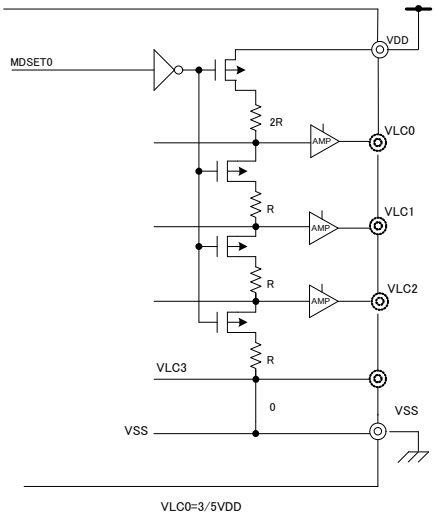
LCDM_2	LCDM_0	MDSET1	MDSET0	Bias	step-down transforming	LCDR0	LCDR1	LCDR2	LCDR3	LCDR4
0	0	0	1	1/3 (Four-time slot)	No	N/A	√	√	√	N/A
0	0	1	1		Yes	√	√	√	√	N/A
0	1	0	1	1/3 (Three-time slot)	No	N/A	√	√	√	N/A
0	1	1	1		Yes	√	√	√	√	N/A
1	0	0	1	Static	No	N/A	√	√	√	N/A
1	0	1	1		Yes	√	√	√	√	N/A

Figure 16-27. Examples of LCD Drive Power Connections

(a) 1/3 bias method and static display mode (MDSET1, MDSET0 = 0, 1) (example of V_{DD} = 5 V, V_{LC0} = 5 V)



(b) 1/3 bias method and static display mode (MDSET1, MDSET0 = 1, 1) (example of V_{DD} = 5 V, V_{LC0} = 3 V)



<R>

CHAPTER 17 LCD BUS INTERFACE (128-pin products only)

The LCD Bus Interface connects the internal RL78 bus system to an external LCD Controller/Driver. It provides an asynchronous 8-bit parallel data bus and two control lines.

The LCD Bus Interface supports bidirectional communication. You can send data to and query data from the LCD controller.

17.1 Functions of LCD Bus Interface

The functions of the LCD Bus Interface in the RL78/D1A are as follows.

- Support of two different control signals modes:
  - mod80 with separate read and write strobe
  - mod68 with read/write signal and data strobe “E” with selectable level
- Data transfer sequence starts when internal data bus access LBDATA register
- 8/16 bit write and read operations
- Programmable transfer speed (max.10 MHz) through
  - selectable clock input
  - programmable transfer time
  - programmable wait states
- DMA trigger generation selectable upon two events (The interrupt can be used as DMA trigger only.)
  - internal data transfer allowed
  - external bus access completed
- Flags that indicate the status of the data register and the progress of data transfer to or from the LCD controller.
- DMA for read and write operations

Caution When LCDB is used under EVDDx ≦ VDD, registers of LCD C/D related must be initial value (LCDON=0, SCOC=0, MDSET1-0=00, LCDPFx=0), otherwise the normal operation can’t be guaranteed.

Remark If the concerned pins are configured as LCD Bus Interface pins, change between input and output is performed automatically by LCD Bus Interface to do read and write operations.

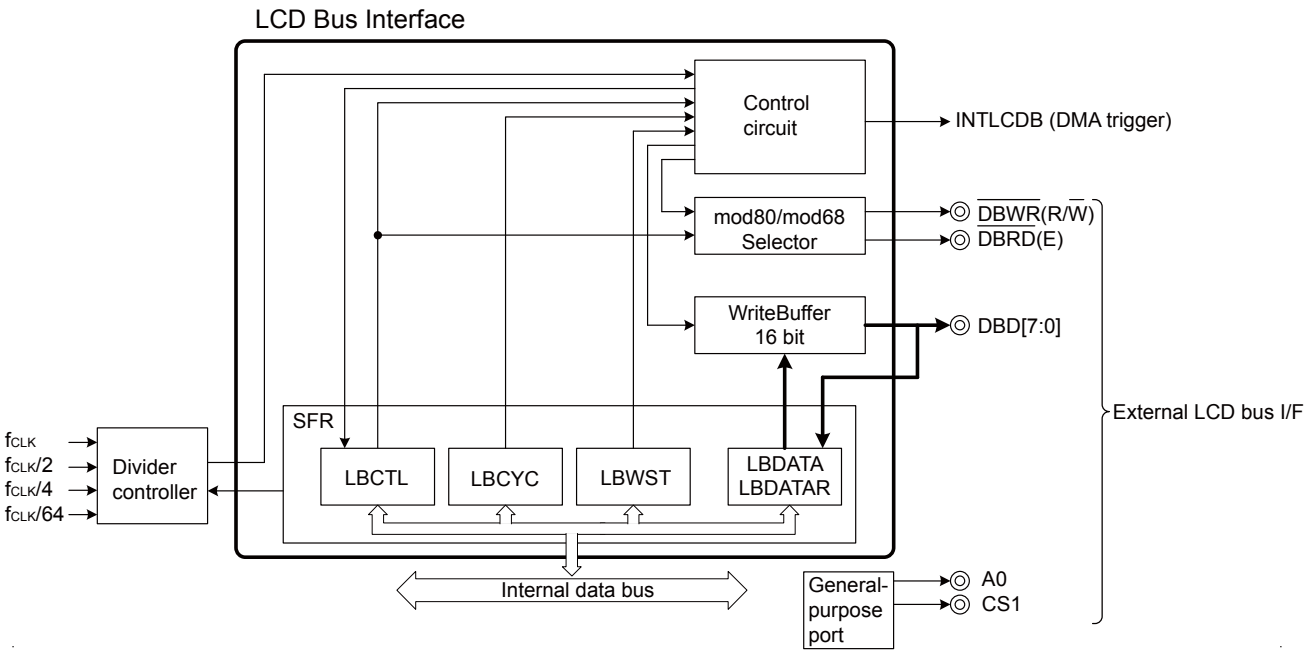
17.2 Configuration of LCD Bus Interface

The LCD Bus Interface consists of the following hardware.

Table 17-1. Configuration of LCD Bus Interface

Item	Configuration
Data I/O pins	8 pins (DBD7 to DBD0)
Control pins	$\overline{\text{DBWR}}$, $\overline{\text{DBRD}}$ (mod80 mode (IMD = 0)) R/W, E (mod68 mode (IMD = 1))
Data registers	LCD Bus Interface data register (LBDATA, LBDATAL) LCD Bus Interface read data register (LBDATAR, LBDATARL)
Control registers	LCD Bus Interface mode register (LBCTL) LCD Bus Interface cycle time register (LBCYC) LCD Bus Interface wait status register (LBWST) Port Mode registers 4,11 (PM4,PM11) Port registers 4,11 (P4,P11) Peripheral Enable Register 1 (PER1)

Figure 17-1. Block Diagram of LCD Bus Interface



(1) LCD Bus Interface data register (LBDATA, LBDATAL)

LBDATA contains the data that is transferred via the LCD Bus Interface.

This register can be read/written by 2 different ways as the followings:

- LBDATA: 16-bit access
- LBDATAL: 8-bit access

Reset signal generation sets LBDATA to 0000H.

Figure 17-2. Format of LCD Bus Interface data register (LBDATA, LBDATAL)

Address: FFF44H After reset: 0000H R/W

									LBDATAL							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LBDATA																

Access types

Depending on the access to this register (8-bit or 16-bit), a defined number of transfers via the external bus interface are performed:

- 8-bit accesses:
  The 8-bit accesses is transferred via the bus interface.
- 16-bit accesses:
  The 16-bit accesses is split into 8 bits (first low-8bit, then high-8bit) that are transferred consecutively via the bus interface.

When the data is split into bits and transferred consecutively, the bit order is as follows:

15	8	7	0
2nd 8-bit		1st 8-bit	

Write to this register

A write operation to this register sets the busy flag LBCTL.BYF immediately.

If there is no LCD bus transfer in progress (LBCTL.TPF = 0), the data is copied to the write buffer and LBCTL.BYF is cleared.

If there is a transfer going on (LBCTL.TPF = 1), the data is not copied to the write buffer until the transfer has completed. As soon as the transfer is complete, the data is copied to the write buffer and LBCTL.BYF is cleared.

A transfer via the LCD Bus Interface starts as soon as the LBDATA register is copied to the write buffer. This is indicated by INTLCDB (DMA trigger) that becomes active, provided that LBCTL.TCIS = 0.

Read from this register

A read operation from this register initiates a read transfer via the LCD Bus Interface. The data that is read from the register is always the data that was received during the previous transfer from the LCD Bus Interface.

Remarks 1. Every access must address the base address of the LBDATA register.
Access to the address of LBDATA's high 8 bits is prohibited.

2. LBCTL.BYF must be zero when accessing this register.

(2) LCD Bus Interface read data register (LBDATAR, LBDATARL)

LBDATAR is read-only. It contains the data of the last previous read transfer via the LCD Interface. Reading this register does not start a new read transfer on the LCD Bus Interface.

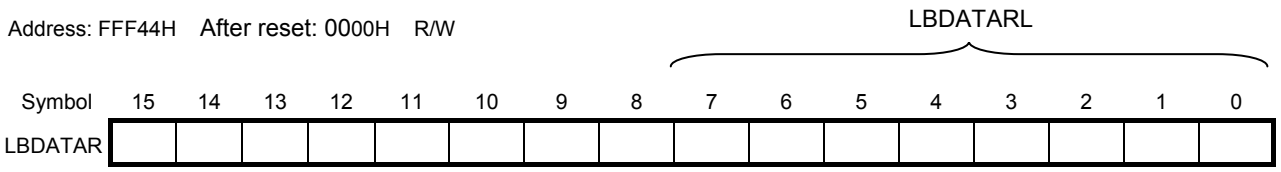
This register can be read in 2 different units under following names:

- LBDATAR: 16-bit access
- LBDATARL: 8-bit access

Reset signal generation sets LBDATAR to 0000H.

Figure 17-3. Format of LCD Bus Interface read data register (LBDATAR, LBDATARL)

Address: FFF44H After reset: 0000H R/W



This register can be read to obtain data that was transferred during a previous read operation to the LBDATA register without initiating a further LCD bus transfer.

Reading the LBDATAR register does not change the status of the LBCTL.BYF and LBCTL.TPF flags.

Remark Read access must address the base address of the LBDATAR register.
Access to the address of LBDATAR's upper 8 bits is prohibited.

17.3 Registers Controlling LCD Bus Interface

The following ten registers are used to control the LCD Bus Interface.

- LCD Bus Interface mode register (LBCTL)
- LCD Bus Interface cycle time register (LBCYC)
- LCD Bus Interface wait status register (LBWST)
- Port Mode registers 4, 11 (PM4, PM11)
- Port registers 4, 11 (P4, P11)
- Peripheral Enable Register 1 (PER1)

(1) Peripheral enable register 1 (PER1)

PER1 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 LCD Bus Interface is used, be sure to set bit 3 (LBEN) of this register to 1.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-4. Format of Peripheral Enable Register 1 (PER1)

Address: F00F1H After reset: 00H R/W: Bits 0 to 2 and 6 (Read Only)

Symbol	<7>	6	<5>	<4>	<3>	2	1	0
PER1	ADCEN	0	MTRCEN	SGEN	LBEN	0	0	0

LBEN	Control of LCD bus controller clock supply
0	Stops input clock supply. • SFR used by the LCD bus controller cannot be written. • The LCD bus controller is in the reset status.
1	Supplies input clock. • SFR used by LCD bus controller can be read and written.

(2) LCD Bus Interface mode register (LBCTL)

LBCTL controls the operation of LCD Bus Interface.

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

Reset signal generation sets LBCTL to 00H.

Figure 17-5. Format of LCD Bus Interface mode register (LBCTL)

Address: F0018H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
LBCTL	EL	IMD	LBC1	LBC0	TCIS	0	TPF	BYF

EL	Control the level of signal “E” in mod68 mode
0	E is active high; data is read/written on the falling edge.
1	E is active low, data is read/written on the rising edge.

IMD	Mode of external bus interface access selection
0	mod80 mode - control signals are $\overline{WR}$ and $\overline{RD}$
1	mod68 mode - control signals are E and R/$\overline{W}$

LBC1	LBC0	Internal clock (SPCLK) selection
0	0	f_{CLK}
0	1	$f_{CLK}/2$
1	0	$f_{CLK}/4$
1	1	$f_{CLK}/64$

TCIS	INTLCDB (DMA trigger) generation control bit
0	During write access to the bus interface, an INTLCDB is generated as soon as data is transferred from LBDATA to the write buffer. During read access from the bus interface, an INTLCDB is generated as soon as data is available in the LBDATA and LBDATAR registers.
1	During write access to the bus interface, an INTLCDB is generated as soon as data is transferred from LBDATA to the write buffer. During read access from the bus interface, an INTLCDB is generated as soon as data is available in the LBDATA and LBDATAR registers.

TPF	Flag of transfer in progress on external bus interface
0	The external bus interface is idle
1	Data is being transferred on the external bus interface

BYF	Data register busy flag
0	Data can be read or written from/to LBDATA Data can be read from LBDATAR
1	Register LBDATA (LBDATAR) is busy

Cautions 1. Bits 2 must be set to 0.

2. Though the LBCTL.TPF flag is intended to determine the current status of the LCD bus data transfer, reading of this flag may indicate a wrong status by accident.
Therefore, instead of polling the LBCTL.TPF flag it is recommended to use a DMA transfer to load new LCD data into the LCD bus interface data register (LBDATAx).

(3) LCB Bus Interface cycle control register (LBCYC)

LBCYC register determines the cycle time of the LCD Bus Interface.

The cycle time is the duration of one bus access for transferring one 8-bit data.

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

Reset signal generation sets LBCYC to 00H.

Figure 17-6. Format of LCB Bus Interface cycle control register (LBCYC)

Address: F0019H After reset: 02H R/W

Symbol	7	6	5	4	3	2	1	0
LBCYC	0	0	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0

CYC5-CYC0	Cycle time
000000B	Setting prohibited
000001B	
000010B	Cycle time is 2 × T
000011B	Cycle time is 3 × T
:	:
111110B	Cycle time is 62 × T
111111B	Cycle time is 63 × T

Remarks 1. T is the clock period of the selected clock (set by LBC1 and LBC0)

2. Always keep LBCYC ≥ 2.

(4) LCB Bus Interface wait control register (LBWST)

LBWST determines the number of wait states of the LCD Bus Interface. The number of wait states defines the duration of the $\overline{DBWR}$ and $\overline{DBRD}$ signals. This duration must remain below the cycle time.

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

Reset signal generation sets LBWST to 00H.

Figure 17-7. Format of LCB Bus Interface wait control register (LBWST)

Address: F001AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LBWST	0	0	0	WST4	WST3	WST2	WST1	WST0

WST4-WST0	Wait cycles
00000B	No wait cycle inserted
00001B	1 wait cycle
00010B	2 wait cycle
00011B	3 wait cycle
:	:
11110B	30 wait cycle
11111B	31 wait cycle

Remarks 1. 1 wait cycle is the clock period of the selected clock (set by LBC1 and LBC0).

2. Always keep WST ≤ CYC − 2.

(5) Port mode registers 4, 11 (PM4, PM11)

These registers set input/output of ports 4, 11 in 1-bit units.

When using the pins as LCD Bus Interface I/O, set the port register and port mode register as shown in Figure J-8.

PM4, PM11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure J-8. Format of Port Mode Registers 4, 11 (PM4, PM11)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFF2B	FFH	R/W

PMmn	Pmn pin I/O mode selection(m = 4, 11 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

17.4 Operation of Timing

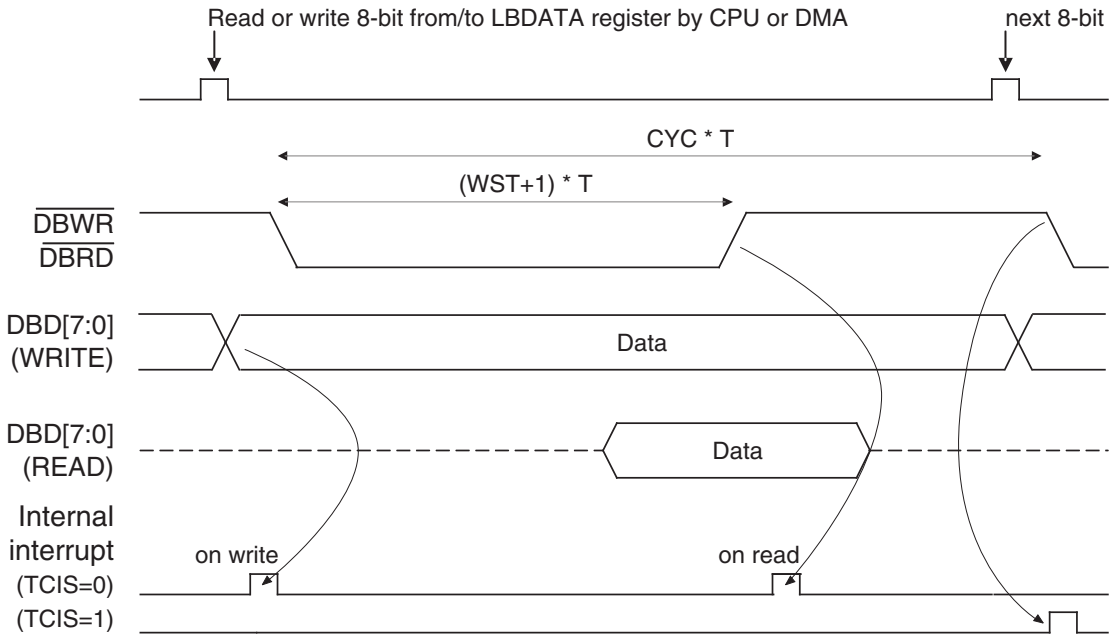
This section starts with the general timing and then presents examples of consecutive write and read operations.

17.4.1 Timing dependencies

The following figure shows the general timing when the mod80 mode is used.

It illustrates the effect of the LBCYC and LBWST register settings. It explains also the impact of LBCTL.TCIS on the INTLCDB generation.

Figure 17-9. LCD Bus Interface timing (mod80 mode)



In mod80 mode, $\overline{DBWR}$ provides the write strobe $\overline{WR}$ and $\overline{DBRD}$ the read strobe $\overline{RD}$.

Notes 1. T is the clock period of the internal clock (SPCLK) selected with the LBC1 and LBC0 bits.

2. CYC is the chosen number of clock cycles (LBCYC).
Always keep LBCYC > 2.

3. WST is the chosen number of wait states (LBWST).
Always keep LBWST < (LBCYC – 2).

The only difference in mod68 mode is, that $\overline{DBWR}$ provides the read/write R/$\overline{W}$ strobe and $\overline{DBRD}$ the E strobe. The active edge of the E strobe is defined by LBCTL.EL.

17.4.2 LCD Bus I/F states during and after accesses

Changing between input and output mode of the LCD bus pins DB[7:0] is done automatically after they are configured as LCD Bus Interface pins via the port configuration registers.

After the pins are configured as DB[7:0] they are operating in input mode.

During and after a bus read access DB[7:0] are operating in input mode and retain this mode also after the read access is completed.

During and after a bus write access DB[7:0] are operating in output mode and retain this mode also after the write access is completed.

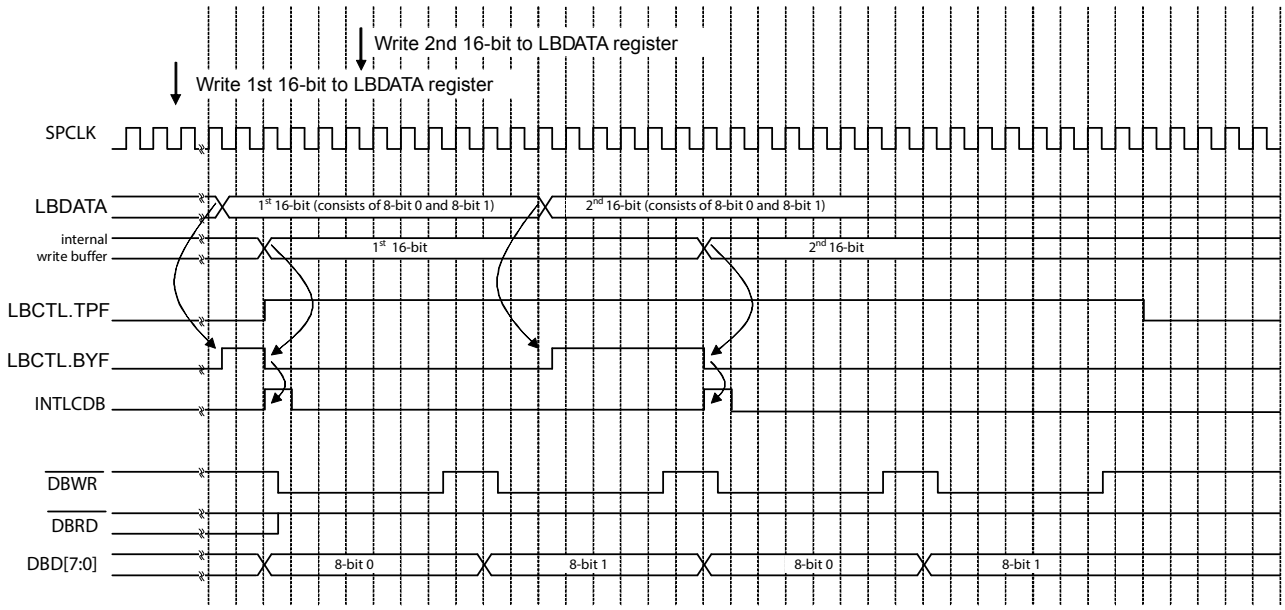
17.4.3 Writing to the LCD bus

This section shows typical sequences of writing 16 bits and 8 bits to the LCD bus.

(1) 16-bit writing

16-bit writing transmits two 8-bit data to the external LCD Controller/Driver.

Figure 17-10. Timing (mod80: LBTCTL.IMD = 0): write consecutive 16 bits, LBWST = 5, LBCYC = 8, LBCTL.TCIS = 0



Note The timing diagrams are for functional explanation purposes only without any relevance to the real hardware implementation.

(a) Sequence

<1> The first 16 bits of LCD data is written to the LBDATA register. The internal bus transfer takes some clocks until the interface register is written.
Then the busy flag LBCTL.BYF is set until the data is copied to the write buffer.

<2> The LBDATA register contents is copied to the write buffer. This clears LBCTL.BYF and causes the INTLCDB output to become active for one clock cycle. Transfer on the LCD bus interface starts with 8-bit data 0. The flag LBCTL.TPF is set to indicate that a transfer is in progress.

<3> Caused by the INTLCDB, the DMA writes a second 16 bits to LBDATA.
The CPU can write this 16 bits as well after it has checked the busy flag LBCTL.BYF. The internal bus transfer again takes some clock cycles until the LBDATA register is written and LBCTL.BYF is set.

<4> Because the transfer (two 8-bit data) on the LCD bus interface is still going on and the LBDATA register contents can not be copied to the write buffer immediately, LBCTL.BYF is set.

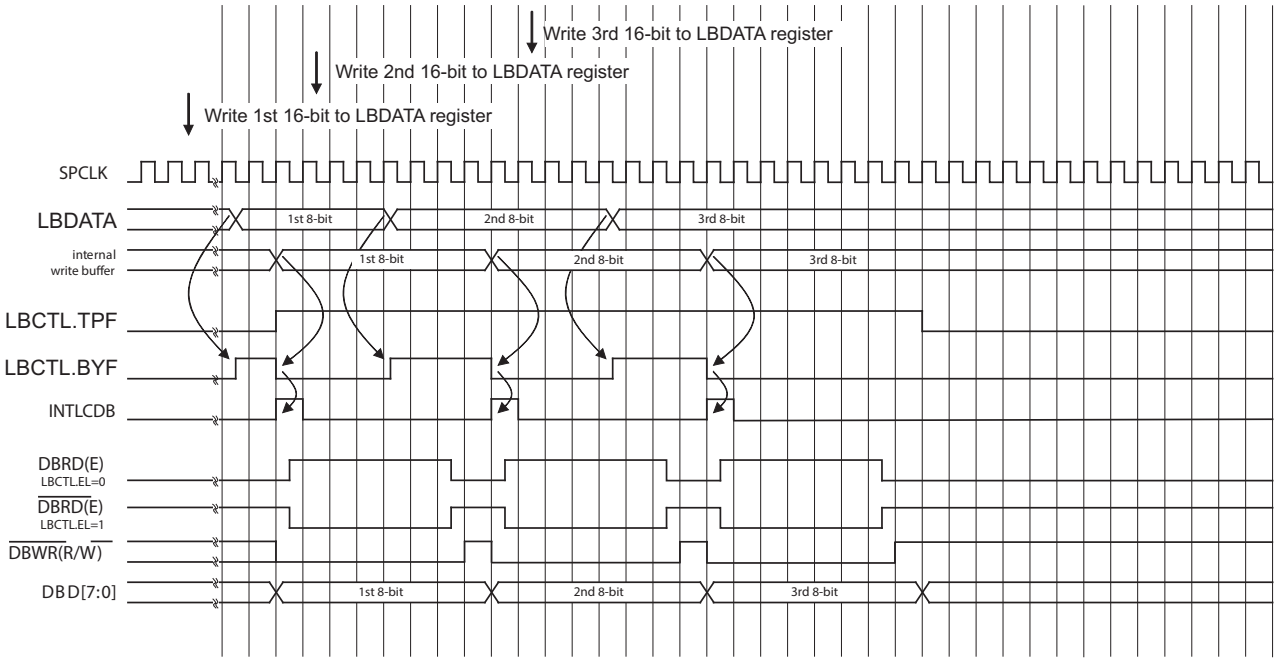
<5> After the transfer over the LCD bus interface has been completed, the write buffer is filled with the contents of LBDATA. The busy flag LBCTL.BYF is cleared, and the INTLCDB becomes active for one clock cycle.

Filling the write buffer starts a new transfer to the external LCD controller.

(2) 8-bit writing

Writing consecutive 8 bits transmits these 8 bits to the external LCD controller/driver.

Figure 17-11. Timing (mod68 mode: LBTCTL.IMD = 1): write consecutive 8 bits, LBWST = 5, LBCYC = 8, , LBCTL.TCIS = 0



Note The timing diagrams are for functional explanation purposes only without any relevance to the real hardware implementation.

(a) Sequence

<1> The first 8-bit of LCD data is written to the LBDATA register. The internal bus transfer takes some clocks until the register of the interface is written.
Then the busy flag LBCTL.BYF is set until the data is copied to the write buffer.

<2> The LBDATA register contents is copied to the write buffer. This clears LBCTL.BYF and causes the INTLCDB output to become active for one clock cycle. Transfer on the LCD bus interface is started. The flag LBCTL.TPF is set to indicate that a transfer is in progress.

<3> Caused by the INTLCDB, the DMA writes a second 8-bit to LBDATA. The CPU can write this 8-bit as well after it has checked the busy flag LBCTL.BYF. The internal bus transfer again takes some clock cycles until the LBDATA register is written and LBCTL.BYF is set.

<4> Since the transfer (one 8-bit data) on the LCD bus interface is still going on and the LBDATA register contents can not be copied to the write buffer immediately, the busy flag LBCTL.BYF remains set.

<5> After the transfer on the LCD bus interface has been completed, the write buffer is filled with the contents of LBDATA. The busy flag LBCTL.BYF is cleared and the INTLCDB becomes active for one clock cycle.

Filling the write buffer starts a new transfer to the external LCD controller.

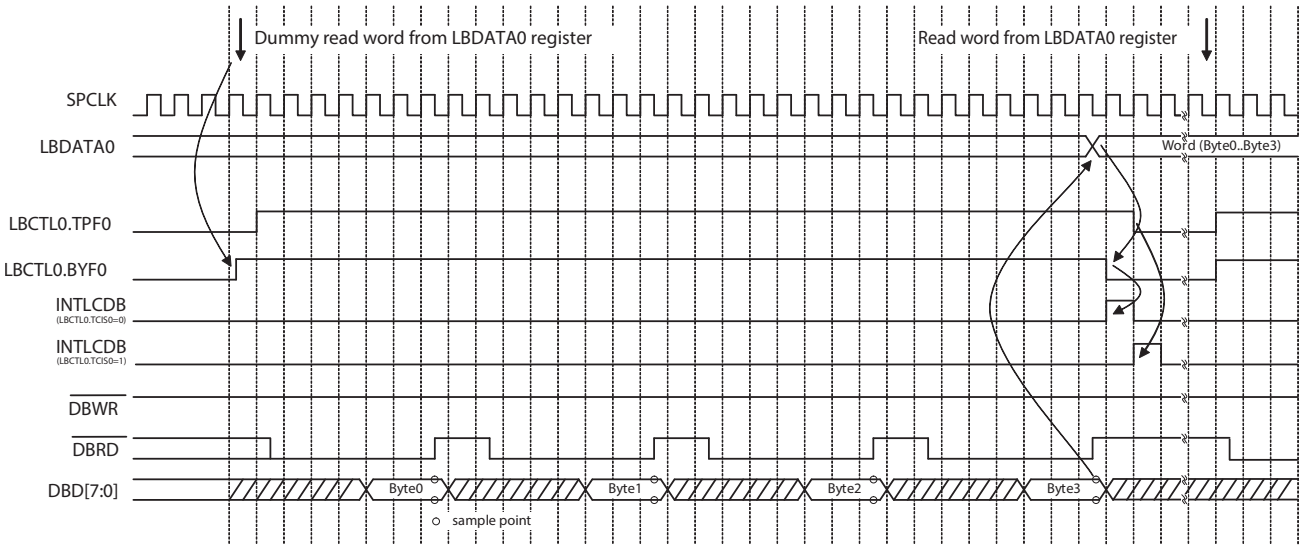
17.4.4 Reading from the LCD bus

You can read from the LCD bus in 8-bit or 16-bit format. The following shows typical sequences of reading 8 bits.

(1) 16-bit reading

The following figure shows 16 bits read operation in mod80 mode.

Figure 17-12. Timing (mod80: LBTCTL.IMD = 0): read word, LBWST = 5, LBCYC = 8, LBCTL.TCIS = 0 and 1



Note The timing diagrams are for functional explanation purposes only without any relevance to the real hardware implementation.

(a) Sequence

<1> A dummy read to the LBDATA register starts the transfer of four bytes from the external LCD controller. The busy flag LBCTL.BYF is set immediately. The “transfer in progress” flag LBCTL.TPF is set on the rising edge of the clock.

The data that is read from LBDATA belongs to a previous transfer and may be ignored.

<2> When the last of the four bytes is sampled and the complete word is available in the LBDATA register, the busy flag LBCTL.BYF is cleared.

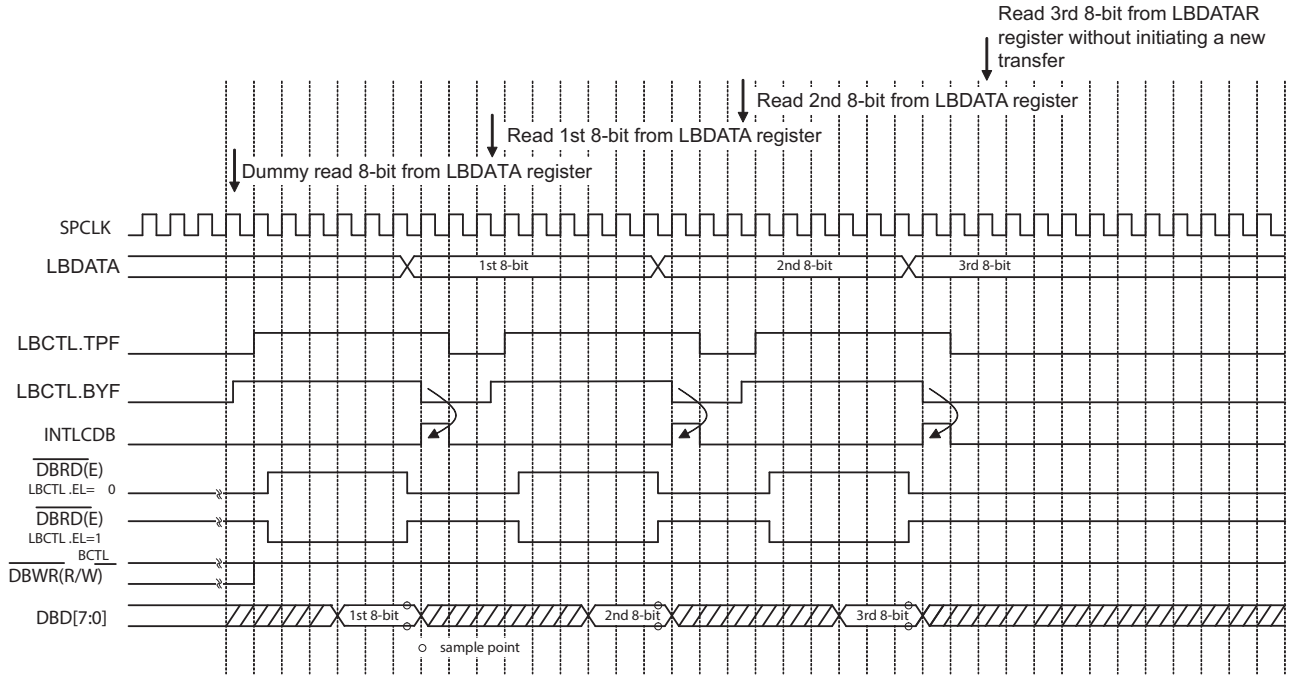
The LBCTL.TPF flag remains set until the cycle time of the last byte has elapsed.

<3> A following read to the LBDATA register provides the LCD controller data and initiates a new transfer.

(2) 8-bit reading

The following figure shows 8 bits read operation in mod68 mode.

Figure 17-13. Timing (mod68: LBTCTL.IMD = 1): read consecutive 8 bits, LBWST = 4, LBCYC = 7, LBCTL.TCIS = 0



Note The timing diagrams are for functional explanation purposes only without any relevance to the real hardware implementation.

(a) Sequence

<1> A dummy read to the LBDATA register starts the transfer of one 8-bit data from the external LCD controller. The busy flag LBCTL.BYF is set immediately. The “transfer in progress” flag LBCTL.TPF is set on the rising edge of the clock.

The data that is read from LBDATA belongs to a previous transfer and may be ignored.

<2> When the data on the LCD Bus Interface is sampled, LBCTL.BYF is cleared and the data is available in LBDATA. The interrupt output INTLCD becomes active for one clock cycle.

<3> A new read to LBDATA is performed while the previous transfer has not been finished (cycle time not elapsed). The busy flag LBCTL.BYF is set immediately, but the new transfer is started after the previous one is complete. The “transfer in progress flag” LBCTL.TPF remains set.

The data that is read from LBDATA is the first 8-bit LCD data.

<4> Again, the data that has been sampled is available in LBDATA and the busy flag LBCTL.BYF is cleared.

<5> Steps 2 to 4 are repeated until the last 8 bits to be read has been sampled.

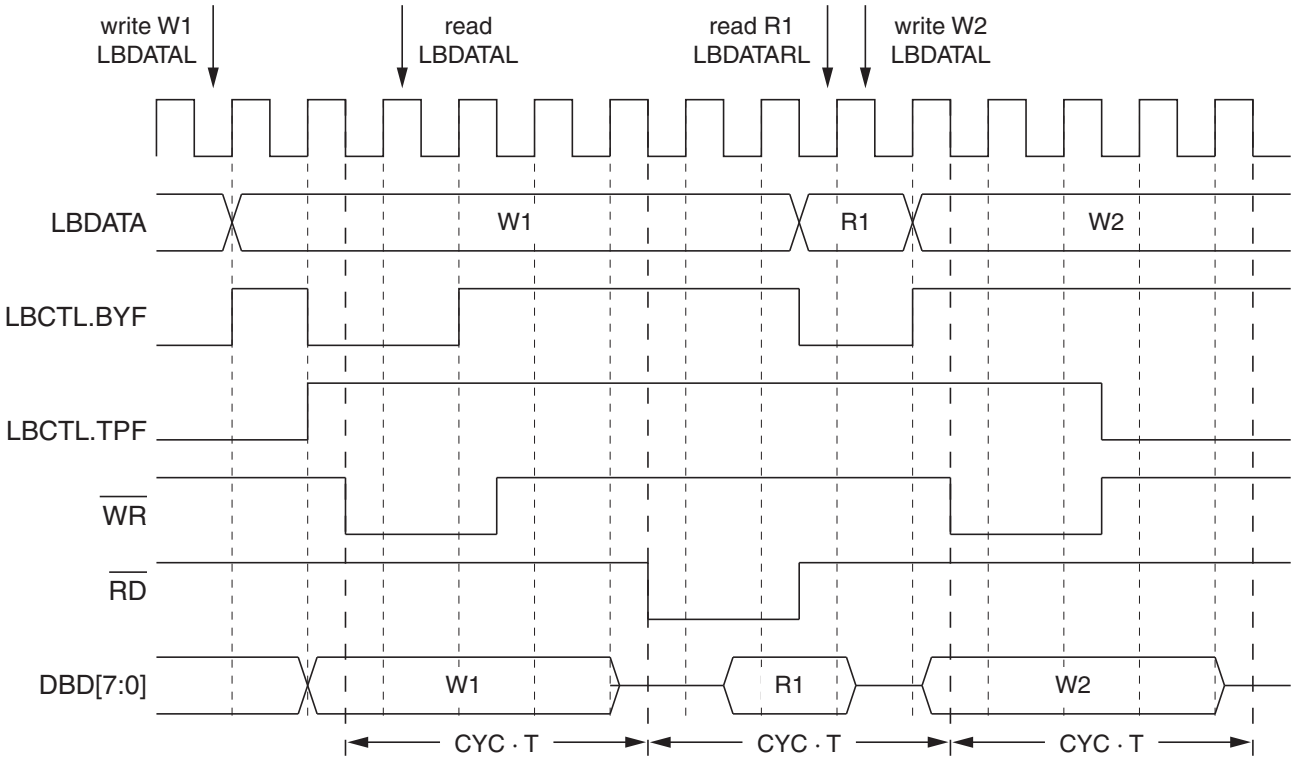
<6> The last 8 bits is not read from the LBDATA register but from LBDATAR in order to avoid a further read transfer on the LCD bus.

17.4.5 Write-Read-Write sequence on the LCD bus

Figure 17-14 shows an example when a write access to the LCD bus is immediately followed by a read access and vice versa. The example is given in mod80 mode (LBCTL.IMD = 0) with 8-bit transfers.

In mode68 mode (LBCTL.IMD = 1) the timing is equivalent, when the RD strobe is considered as the low active E signal (LBCTL.EL = 1).

Figure 17-14. Timing (mod80: LBTCTL.IMD = 0): 8-bit write-read-write, LBWST = 4, LBCYC = 7, LBCTL.TCIS = 0



17.5 Cautions for LCD Bus Interface

17.5.1 Polling of LBCTL.TPF flag may indicate wrong status

Though the LBCTL.TPF flag is intended to determine the current status of the LCD bus data transfer, reading of this flag may indicate a wrong status by accident.

Therefore, instead of polling the LBCTL.TPF flag it is recommended to use a DMA transfer to load new LCD data into the LCD bus interface data register (LBDATAx).

17.5.2 Writing to the LBDATA/ LBDATAL register

When writing to the LBDATAx register while a transfer on the LCD data bus is ongoing a corrupt data transfer may be the result. The critical situation can occur under certain clock constellations.

To avoid the critical situation one of the following measures must be applied.

- **Avoidance of simultaneous write to LBDATAx register and LCD data bus transfer**

  To ensure that LBDATAx register is not written while a transfer is ongoing, the LBDATAx register should be operated upon the occurrence of the LCD Bus Interface interrupt (INTLCDB) with LBCTL.TCIS set to 1.

17.6 Example of LCD Bus Interface Transmission

17.6.1 Connection example of external LCD driver

Example 1.

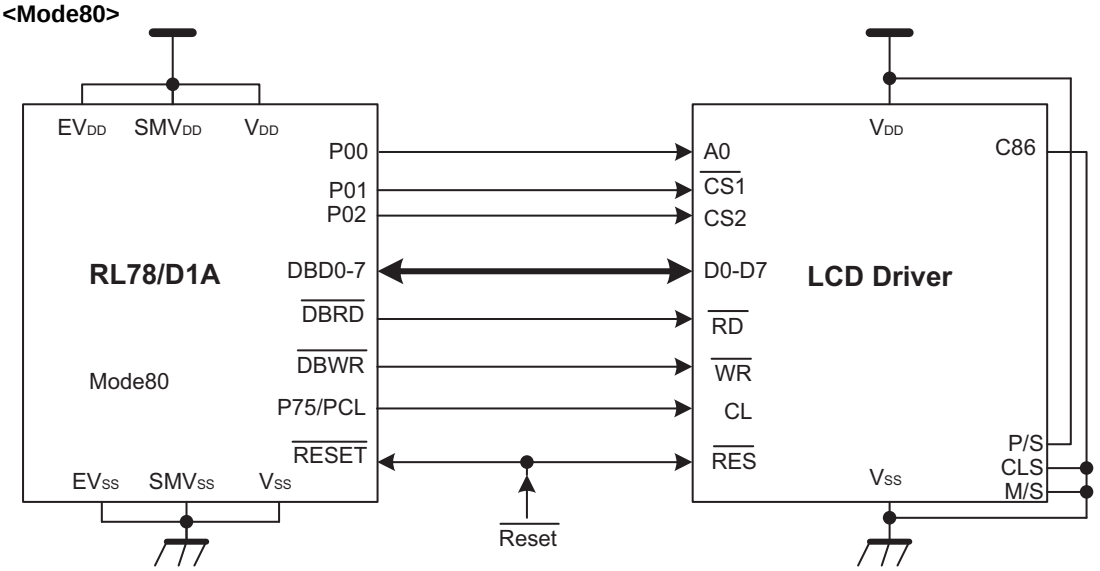
RL78/D1A can be used as a master chip and supply clock from the PCL pin to slave chip (LCD driver) for display clock.

System composition :

- System clock 32 MHz, LCDB access cycle 8 MHz ($f_{CLK}/4$)
- Mod68/80
- CL comes from PCL ($f_{CLK}/2^{11}$ = 15.6 kHz)
- Display ”E”

Figure 17-15. Connection example 1

<Mode80>



<Mode68>

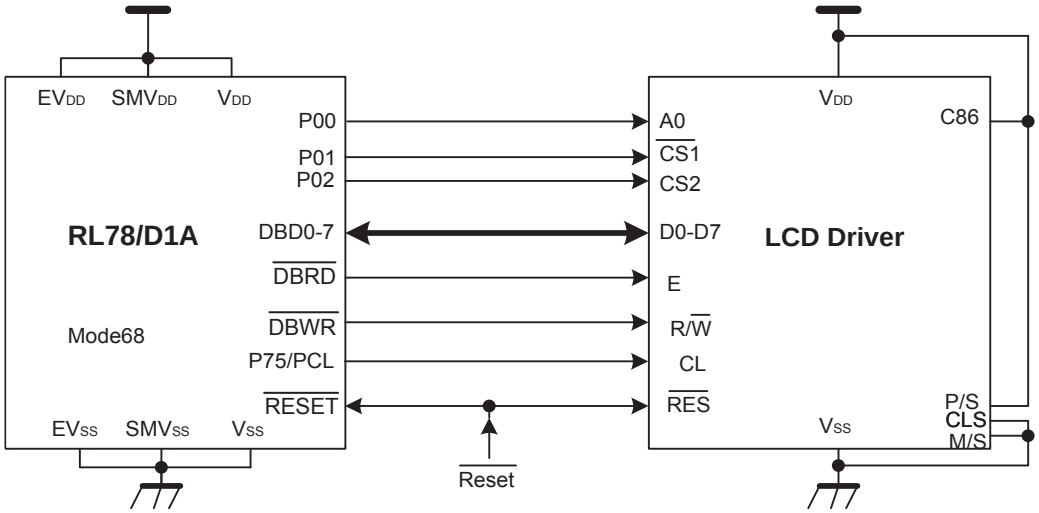


Table 17-2. Connection example 1

No.	LCD driver pin	LCD driver function	Port name
1	A0	To determine D0 to D7 are data or command	P00 Note
2	$\overline{CS1}$	Chip select	P01 Note
3	CS2	Chip select	P02 Note
4	D0 to D7	8-bit bi-directional data bus	DBD0 to DBD7
5	$\overline{RD}$(E)	mod80: read strobe mod68: Enable strobe	_DBRD
6	$\overline{WR}$(R/$\overline{W}$)	mod80: write strobe mod68: Read/Write control	_DBWR
7	CL	Display clock	P75/PCL
8	$\overline{RES}$	Reset	_RESET

Note Using P00 to P02 as A0, $\overline{CS1}$, and CS2 is only an example, other port can also be used.

Example 2.

RL78/D1A can be used as a master chip and supply clock from PCL pin to slave chip (LCD driver) for display clock.

System composition :

- f_{CLK} = 6 MHz
- PCF21119x Mod68
- f_{OSC} comes from PCL (f_{CLK}/16 = 375 kHz)

Figure 17-16. Connection example 2

<Mode68>

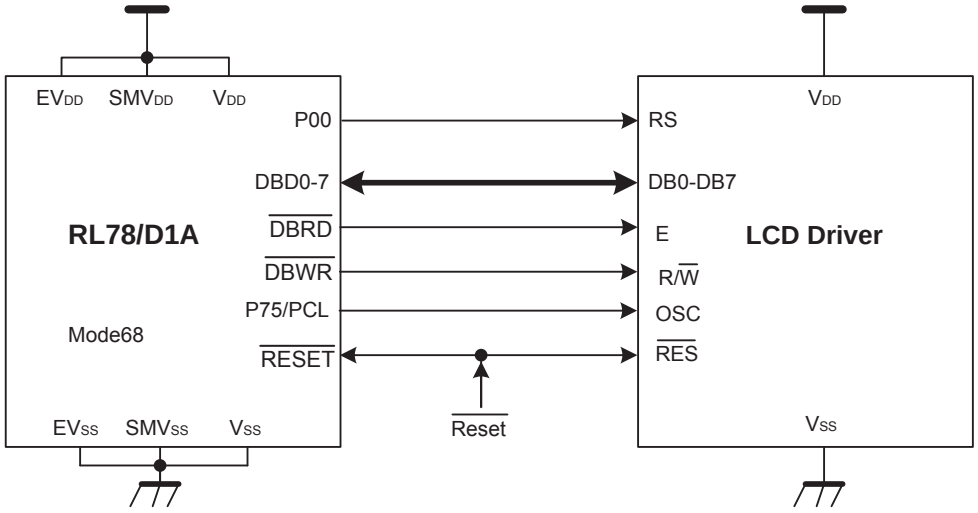


Table 17-3. Connection example 2

No.	LCD driver pin	LCD driver function	Port name
1	RS	Register select	P00 Note
2	DB0 to DB7	8-bit bi-directional data bus	DBD0 to DBD7
3	E	Enable strobe	_DBRD
4	R/$\overline{W}$	Read/Write control	_DBWR
5	OSC	Oscillator or external clock input	P75/PCL
6	$\overline{RES}$	Reset	_RESET

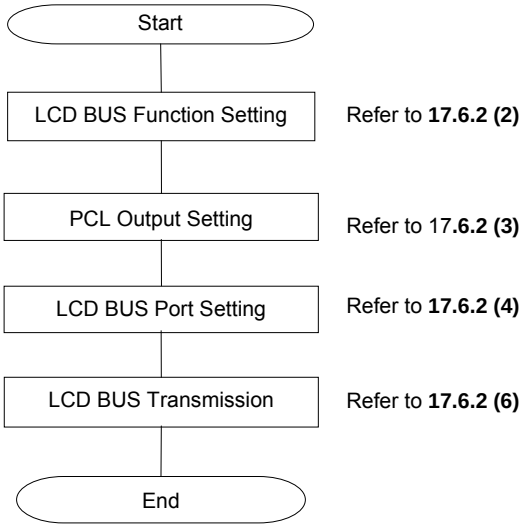
Note Using P00 as RS is only an example, other port can also be used.

17.6.2 Operation procedure of LCD BUS transmission

(1) Flow chart (Reference)

This flow chart is the operation procedure of LCD BUS transmission. Every step is described in details at the following sections. (Right side is the section number.)

Figure 17-17. Whole Flowchart of LCD BUS transmission

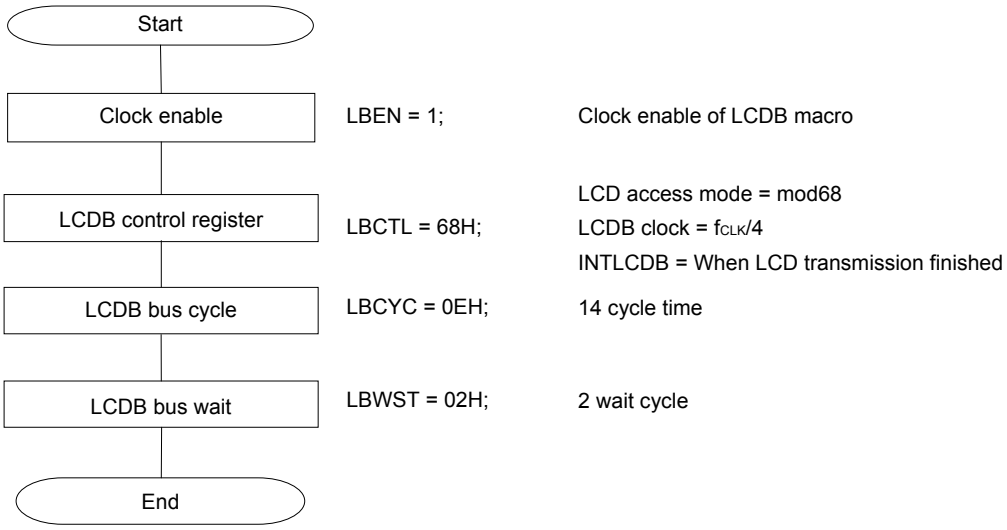


(2) LCD BUS Function Setting

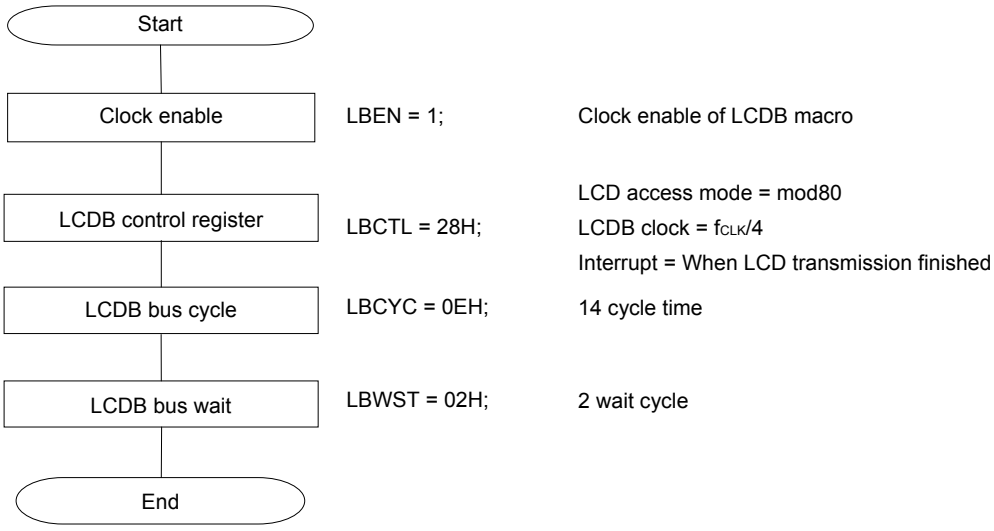
- Enable LCDB macro clock.
- Set access mode to be mode68 or mode80, internal clock (example: f_{CLK}/4),
  INTLCDB = when LCD transmission finished.
- Set LCDB data transmission cycle (example: “14”).
- Set LCDB data transmission wait cycle (example: “2”).

Figure 17-18. Flowchart of LCD BUS Function Setting

<Mode68>



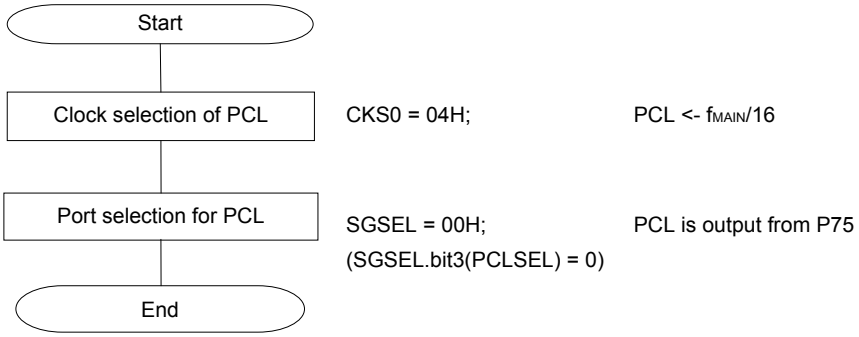
<Mode80>



(3) PCL Clock Setting

- Set the clock of PCL (example: f_{MAIN}/16)
- Set P75 as PCL output

Figure 17-19. Flowchart of PCL Clock Setting



When system clock is higher, suitable PCL frequency divided is needed to satisfy the specification of PCF2119x (f_{OSC} = 120 to 450 kHz), or the specification of S1D15E00 (f_{OSC} = 40 kHz (TYP)). See driver data sheet for details.

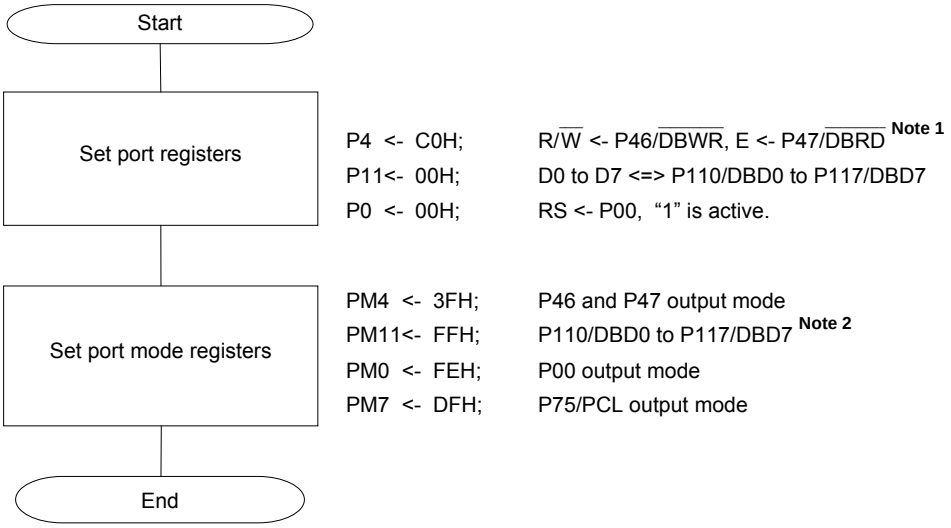
(4) LCD BUS Port Setting

By the following setting, LCDB can be used as bidirectional bus I/F with external LCD diver normally.

- Set PM registers $\overline{\text{DBWR}}$/$\overline{\text{DBRD}}$ to be output mode.
- Set Port registers $\overline{\text{DBWR}}$/$\overline{\text{DBRD}}$ to be “1”.
- Set PM registers of DBD0 to DBD7 to be input mode.
- Set Port registers of DBD0 to DBD7 to be “0”.

Figure 17-20. Flowchart of LCD BUS Port Setting

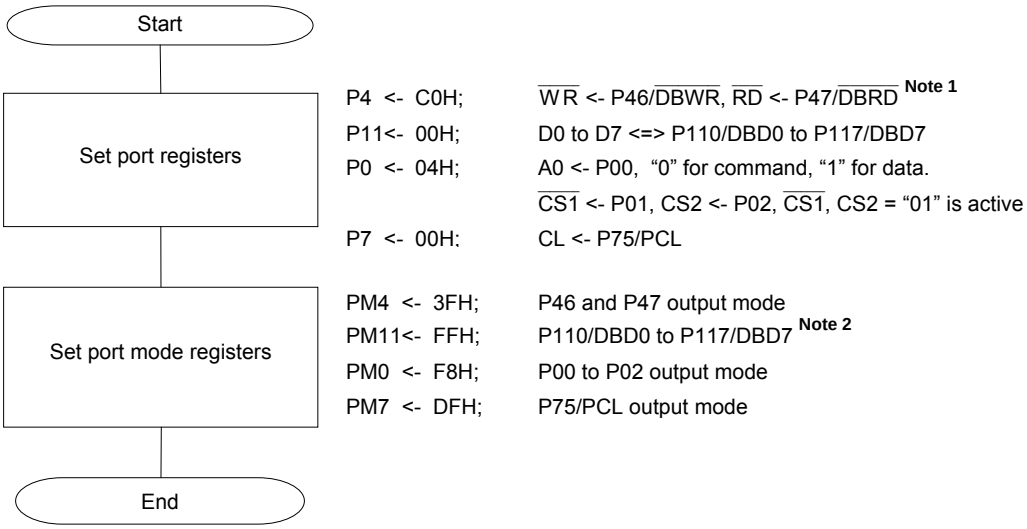
<Mode68 (use PCF2119x) >



Notes 1. P46 and P47 port registers must be set “1” , and LBCTL.bit7(LBEL) = 0

2. Bi-direction, P110 to P117 must be set input mode to achieve bi-direction bus, bus input/output is only determined by $\overline{\text{DBWR}}$ and $\overline{\text{DBRD}}$, not by PM11 register.

<Mode80 (use S1D15E00) >



Notes 1. P46 and P47 port registers must be set “1” because $\overline{\text{DBRD}}$ and $\overline{\text{DBWR}}$ are “L” active.

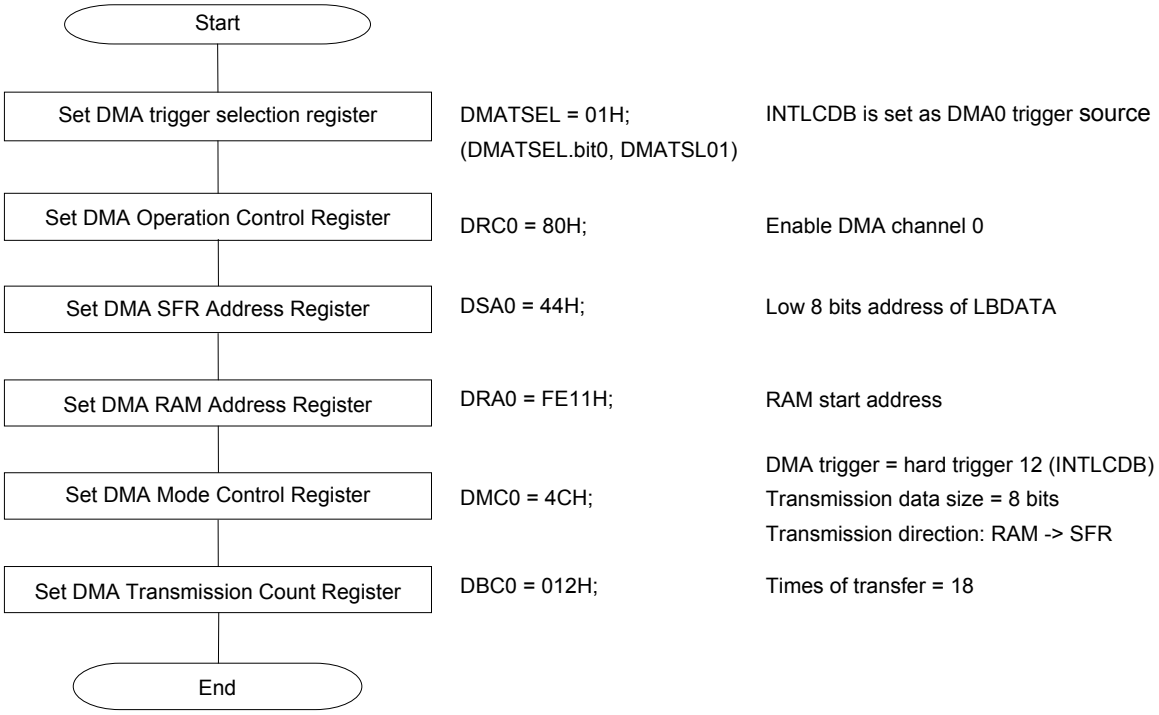
2. Bi-direction, P110 to P117 must be set input mode to achieve bi-direction bus, bus input/output is only determined by $\overline{DBWR}$ and $\overline{DBRD}$, not by PM11 register.

(5) DMA transmission setting

- Set DMATSEL = 01H, INTLCDB is set as DMA0 trigger instead of INTCSI00.
- Write LBDATA address (FFF44H) to DMA SFR address register.
- Write FFE11H to DMA RAM address register. **Note**
- Set DMA trigger source to be “1100” (INTLCDB).
- Set transmission data size (example: 8 bits).
- Set transmission direction (example: RAM -> SFR).
- Set times of transfer (example: 18).

Note First transmission, the content of FFE10H need to be transferred by normal “Write to LBDATA” means.

Figure 17-21. Flowchart of DMA transmission setting



LCD driver -> LCDB read operation setting example:

Set DSA0 = 44H (SFR LBDATA)

Set DRA = FE30H (RAM read data start address)

Set DBC0 = 005H (times of transfer)

Set DMC0 = 0CH (SFR -> RAM)

(6) LCD BUS transmission

- **Use S1D15E00 (EPSON)**

  The data/command transmitted via DMA should be beforehand stored in a RAM address. Setting content, for example, is shown below. Please refer to the data sheet of S1D15E00 for command details.

RAM address	Value	Command	Description
FFE10H	A0H	ADC Select. Normal, SEG0→SEG131: 0(H) →Column Address→83(H)	LCD driver initial (S1D15E0)
FFE11H	C0H	Common Output Mode Select. Normal scanning direction of COM, COM0 → COM95	
FFE12H	A6H	Display Normal/Reverse. RAM data = HIGH Potential at LCD On (normal)	
FFE13H	A4H	Display All Points ON/OFF Normal display mode.	
FFE14H	61H	Duty Ratio Set (2 byte)	
FFE15H	00H	Set duty ratio of 1/8, starting point (block) is 0 (COM0 to 3)	
FFE16H	81H	Electronic Volume (2 byte)	
FFE17H	05H	The electronic volume register is set 05H (Small)	
FFE18H	40H	Temperature Gradient Set Temperature gradient is –0.06%/°C.	
FFE19H	8AH	Display Starting Line Set (2 byte)	Display setting
FFE1AH	00H	Display Starting Line is set to 0.	
FFE1BH	B0H	Set the Page Address The page address is set to 0.	
FFE1CH	10H	Set the Column Address The high-order 4 bits of the display data RAM is 0000B, low-order 4 bits is default (0000B)	
FFE1DH	7FH	Write the Display Data	The display data is “E”
FFE1EH	49H		
FFE1FH	49H		
FFE20H	49H		
FFE21H	41H		
FFE22H	AFH	Turn ON display.	Display ON

- **Use PCF2119x (NXP Semiconductors)**

  Here, only describes the initial routine of LCD driver, the DMA part is omitted, please refer to the former example of S1D15E00. Please refer to the data sheet of PCF2119x for command details.

Value	Command	Description
34H	Function_set Note 8 bits data length, 2 line × 16 characters, 1:18 multiplex drive mode.	LCD driver initial (PCF2119x)
34H	Function_set Note	
34H	Function_set Note	
34H	Function_set Note	
08H	Display_ctl Display, cursor and character blink are off.	
01H	Clear_display Fixed value.	
07H	Entry_mode_set Address increments by 1, display shifts	

Note Same instruction is specified to ensure enough BF checked time.

The flow of LCD BUS transmission without DMA is following.

It takes about 330 μs (165 driver oscillator cycles) to finish Clear_display command, but other commands need about 6 μs (3 driver oscillator cycles) when f_{OSC} = 450 kHz.

Busy flag check operation is carried in PCF2119x. The Busy Flag (BF) indicates the busy state (bit BF = 1) until initialization ends. The busy state lasts 2 ms. The busy flag is output to pin DB7 when RS = 0 and R/$\overline{W}$ = 1.

Pin DB7 of LCD BUS can be used as the busy flag, by reading bit7 of LBDATA/LBDATAR, we can judge whether the driver internal operations are completed or not.

CHAPTER 18 SOUND GENERATOR

The Sound Generator generates an audio-frequency tone signal and a high-frequency pulse-width modulated (PWM) signal. The duty cycle of the PWM signal defines the volume.

By default, the two signal components are routed to separate pins. But both signals can also be combined to generate a composite signal that can be used to drive a loudspeaker circuit.

18.1 Overview

The Sound Generator consists of a programmable square wave tone generator and a programmable pulse-width modulator.

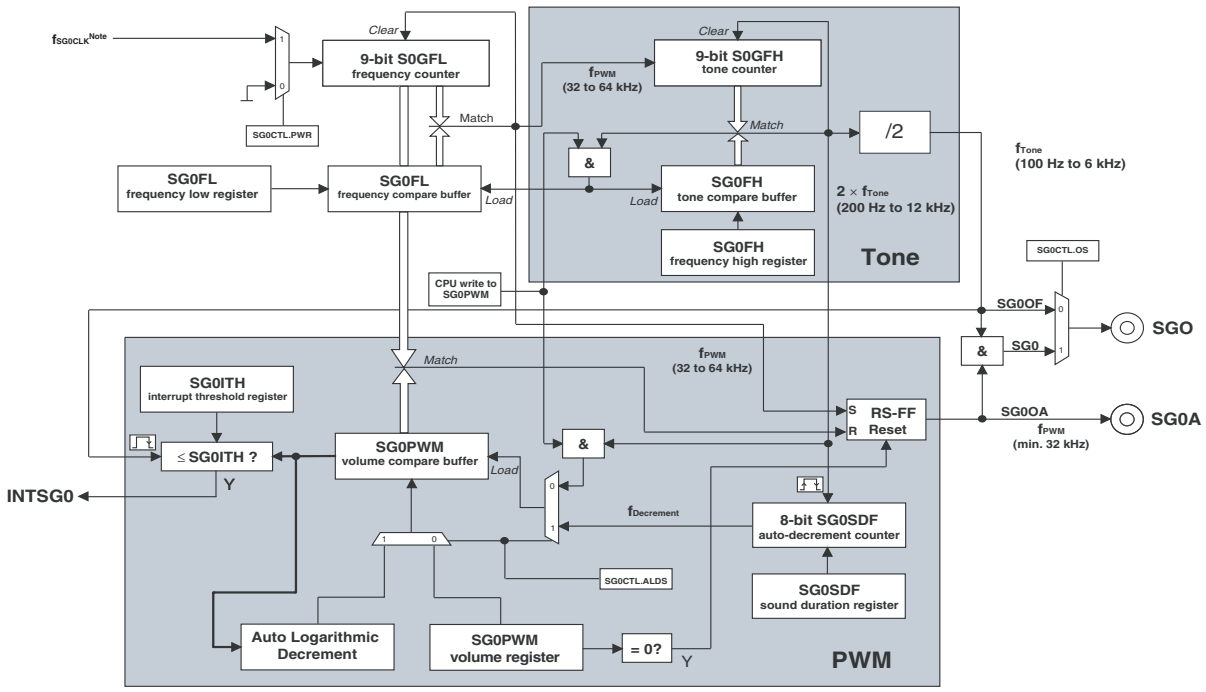
Features

- Programmable sound frequency
- Activation of the automatic logarithmic decrement function for linear volume decreasing without CPU instruction
- Programmable volume level (9 bit resolution)
- Wide range of PWM signal frequency
- Sound can be stopped or retriggered (even if ALD is switched on)
- Composite or separated frequency/volume output for external circuitry variation
- Variable PWM level for generation of the SGTIO interrupt (noise dependent end interrupt)
- Hardware-optimized update of frequency and volume to avoid audible artifacts

18.1.1 Description

The following figure provides a functional block diagram of the Sound Generator.

Figure 18-1. Sound Generator Block Diagram



Note The Sound Generator's input clock frequency f_{SG0CLK} is f_{CLK} or $f_{CLK}/2$. Refer the table below.

f_{CLK}	f_{SG0CLK}	
32 MHz	$f_{CLK}/2$	16 MHz
24 MHz	f_{CLK}	24 MHz
16 MHz	f_{CLK}	16 MHz
8 MHz	f_{CLK}	8 MHz
4 MHz	f_{CLK}	4 MHz

Tone generator

The tone generator consists of two up-counters with compare registers. The values written to the frequency registers are automatically copied to compare buffers. The counters are reset to zero when their values match the contents of the associated compare buffers.

The 9-bit counter S0GFL generates a clock with a frequency between 32 kHz and 64 kHz. This clock constitutes the PWM frequency.

It is also the input of the second 9-bit counter S0GFH. The resulting tone signal behind the by-two-divider has a frequency between 250 Hz and 6 kHz and a 50 % duty cycle.

PWM generator

The PWM generator modulates the duty cycle according to the desired volume. It is controlled by the volume register SG0PWM. The value written to this register is automatically copied to the associated volume compare buffer.

The PWM generator continually compares the value of the counter S0GFL with the contents of its volume compare buffer.

The RS flipflop of the PWM generator is set by the pulses generated by the counter S0GFL. It is reset when the SG0FL counter value matches the contents of the volume buffer. Thus, the PWM output signal can have a duty cycle between 0 % (null volume) and 100 % (maximum volume).

The PWM frequency is above 32 kHz and hence outside the audible range.

Outputs

The Sound Generator is connected to the pins SGO and SGOA. By default, pin SGO provides the tone signal SG0OF and pin SGOA the PWM signal SG0OA that holds the volume (“amplitude”) information.

If bit SG0CTL.OS is set, pin SGO provides the composite signal SG0O that can directly control a speaker circuit.

These signal is output to one of the pin below.

<R>

	48-pin products	64-pin products	80-pin products	100-pin products	128-pin products
SGO/SGOF	P93/P73	P93/P73	P93/P73	P93/P73/P135	P93/P73/P135
SGOA	P92/P72	P92/P72	P92/P72	P92/P72/P134	P92/P72/P134

18.1.2 Principle of operation

The software-controlled registers SG0FL, SG0FH, and SG0PWM are equipped with hardware buffers. The Sound Generator operates on these buffers.

This approach eliminates audible artifacts, because the buffers are only updated in synchronization with the generated tone waveform.

Remark This section provides an overview. For details please refer to **18.3 Sound Generator Operation**.

(1) Generation of the tone frequency

The tone frequency is determined by two counters and their associated compare register values. Two counters are necessary to keep the tone pulse and the PWM signal synchronized.

The first counter (S0GFL) provides the input to the tone generator and also to the PWM generator. It is used to keep the PWM frequency outside the audio range (above 30 kHz) and within the signal bandwidth of the external sound system (usually below 64 kHz). Its match value defines also the 100 % volume level.

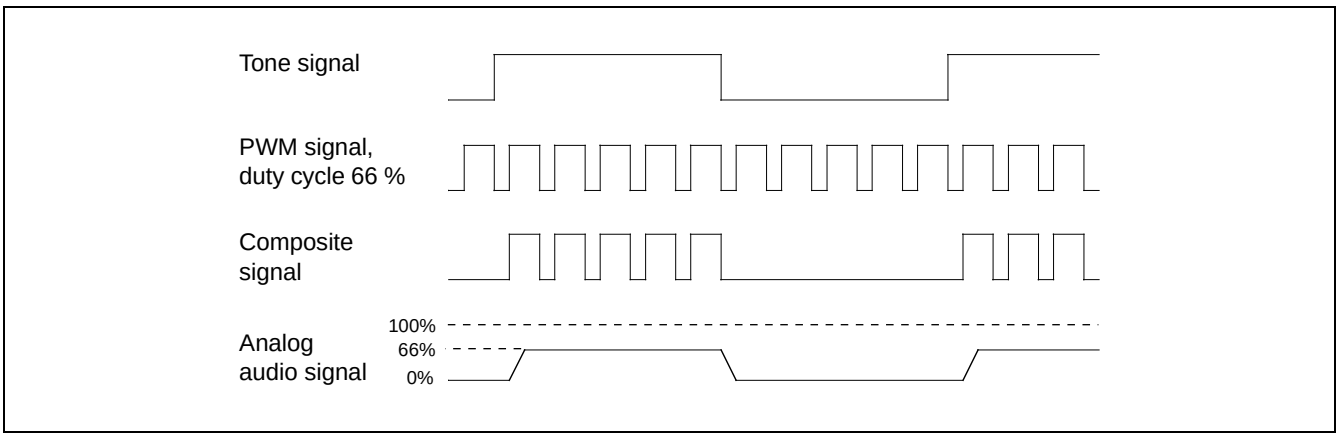
The second counter (S0GFH) generates the tone frequency (250 Hz to 6 kHz).

Remark If the target values of the counters S0GFL/S0GFH are changed to generate a different tone frequency, the volume register SG0PWM has to be adjusted to keep the same volume.

(2) Generation of the volume information

The volume information (the “amplitude” of the audible signal) is provided as a high-frequency PWM signal. In composite mode, the PWM signal is ANDed with the tone signal, as illustrated in the following figure.

Figure 18-2. Generation of the Composite Output Signal



After low-pass filtering, the analog signal amplitude corresponds to the duty cycle of the PWM signal. Low-pass filtering (averaging) is an inherent characteristic of a loudspeaker system.

The duty cycle can vary between 0 % and 100 %. Its generation is controlled by the counter register SG0FL and the volume register SG0PWM.

When the volume register SG0PWM is cleared, the sound stops immediately.

18.2 Sound Generator Registers

The Sound Generator is controlled by means of the following registers:

Table 18-1. Sound Generator Registers Overview

Register name	Shortcut	Address
Control register	SG0CTL	F0287H
Frequency register SG0FL	SG0FL	F0280H
Frequency register SG0FH	SG0FH	F0282H
Amplitude register	SG0PWM	F0284H
Duration factor register	SG0SDF	F0286H
Interrupt threshold register	SG0ITH	F0288H
Sound generator and PCL pin select register	SGSEL	FFF3FH

(1) Peripheral enable register (PER1) and peripheral clock select register (PCKSEL)

Peripheral enable register (PER1)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER1	ADCEN	0	MTRCEN	SGEN	0	0	0	0

Peripheral clock select register (PCKSEL)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PCKSEL	0	CAN MCKE1	CAN MCK1	CAN MCKE0	CAN MCK0	0	0	SGCLK SEL

SG clock source selection

SGEN	SGCLKSEL	Selection of operation clock	Bus clock supply
0	X	Clock supply stopped	Clock supply stopped (SFR write is impossible)
1	0	F_{CLK} is supplied	F_{CLK} is supplied (SFR R/W is possible)
1	1	F_{CLK}/2 is supplied	F_{CLK} is supplied (SFR R/W is possible)

(2) Control register (SG0CTL)

The 8-bit SG0CTL register controls the operation of the Sound Generator.

This register can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 18-3. Format of Control Register (SG0CTL)

Address: F0287H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SG0CTL	0	0	0	SG0PWR	0	0	SG0OS	SG0ALDS

SG0PWR	Power save mode selection
0	Clock input switched off (the Sound Generator is disabled and does not operate).
1	Clock input switched on (the Sound Generator is enabled and ready to use).

SG0OS	SG0 output mode selection
0	Selects frequency output at SGO/SGOF.
1	Selects frequency and amplitude mixed output at SGO/SGOF.

SG0ALDS	Automatic logarithmic decrement of amplitude
0	Automatic logarithmic decrement deactivated.
1	Automatic logarithmic decrement activated.

Caution Change the contents of this register only when the sound is stopped (register SG0PWM cleared).

(3) Frequency register SG0FL (SG0FL)

The 16-bit SG0FL register is used to specify the target value for the PWM frequency. It holds the target value for the 9-bit counter SG0FL.

This register can be read/written in 16-bit units. It cannot be written if bit SG0CTL.PWR = 0.

This register is cleared by any reset.

Address: F0280H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0FL	0	0	0	0	0	0	0	Counter SG0FL target value								

For the calculation of the resulting PWM frequency refer to **18.3.2 (2) PWM calculations**.

The value written to SG0FL defines also the reference value for the maximum sound amplitude (100% PWM duty cycle). A 100 % duty cycle (continually high) will be generated if the SG0PWM value is higher than the SG0FL value. For details see **18.3.2 (2) PWM calculations**.

Remarks 1. The bits SG0FL[15:9] are not used.

2. The maximum value to be written is 510 (01FEH). This yields a PWM frequency of 19.7 kHz in case of the sound generator input clock SG0CLK 10 MHz. The minimum value to be written depends on the capability of the external circuit. A value of 408 (198H) would yield a PWM frequency of 39.1 kHz in case of the sound generator input clock SG0CLK 16 MHz.

3. The value read from this register does not necessarily reflect the current PWM frequency, because this frequency is determined by the frequency compare buffer value. The buffer might not be updated yet. For details see **18.3.1 (1) Updating the frequency buffer values**.

(4) Frequency register SG0FH (SG0FH)

The 16-bit SG0FH register is used to specify the final tone frequency. It holds the target value for the 9-bit counter S0GFH.

This register can be read/written in 16-bit units. It cannot be written if bit SG0CTL.PWR = 0.

This register is cleared by any reset.

Address: F0282H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0FH	0	0	0	0	0	0	0	Counter SG0FH target value								

For the calculation of the resulting tone frequency refer to **18.3.1 (2) Tone frequency calculation**.

Remarks 1. The bits SG0FH[15:9] are not used.

2. Legal values depend on the contents of register SG0FL which defines the frequency of the input pulse. For example: If the counter S0GFL generates a frequency of 32.4 kHz, a value of 63 would generate a tone frequency of 253 Hz.

3. The value read from this register does not necessarily reflect the current tone frequency, because this frequency is determined by the frequency compare buffer value. The buffer might not be updated yet. For details see **18.3.1 (1) Updating the frequency buffer values**.

(5) Amplitude register (SG0PWM)

The 16-bit register SG0PWM is used to specify the sound volume. It holds the target value for the sound amplitude that is given by the duty cycle of the PWM signal.

This register can be read/written in 16-bit units. It cannot be written if bit SG0CTL.PWR = 0.

This register is cleared by any reset.

Address: F0284H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0PWM	0	0	0	0	0	0	0	Sound volume target value								

The value written to this register must be considered in conjunction with the contents of register SG0FL. The register SG0FL specifies the maximum value of the counter S0GFL.

For the calculation of the resulting duty cycle refer to **18.3.2 (2) PWM calculations**.

The setting takes effect after the SG0PWM buffer has been updated (see **18.3.2 (1) Updating the volume buffer value**).

Remarks 1. The bits SG0PWM[15:9] are not used.

2. The value read from this register does not necessarily reflect the current volume, because the value of counter SG0FL is compared with the contents of the volume buffer. The buffer might not be updated yet.

3. The sound stops immediately when this register is cleared.

(6) Duration factor register (SG0SDF)

Address: F0286H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SG0SDF								

The amplitude is decremented every x output frequency pulses at falling edge of sound frequency. In other words, the amplitude will last for x output frequency pulses.

SG0SDF value	x
00000000	1
00000001	2
…	…
11111111	256

(7) Interrupt threshold register (SG0ITH)

Address: F0288H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0ITH	0	0	0	0	0	0	0									

The interrupt SGTIO is generated when the amplitude buffer reaches the value stored in SG0ITH register at next falling edge of sound frequency.

(8) Sound generator and PCL pin select register (SGSEL)

This register is used for alternate switch of sound generator and PCL output pins.

SGOA output can be stopped when it is not used if SGSEL_2 is set to “1”.

Figure 18-4. Format of SGSEL Register

Address: FFF3F After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SGSEL	0	0	0	0	PCLSEL	SGSEL2	SGSEL1	SGSEL0

SGSEL2	SGSEL1	SGSEL0	Pin select of sound generator outputs	
			SGO/SGOF	SGOA
0	0	0	P73	P72
0	0	1	P93	P92
0	1	0	P135	P134
0	1	1	Setting prohibit	
1	0	0	P73	No port is selected (output disabled)
1	0	1	P93	
1	1	0	P135	
1	1	1	Setting prohibit	

18.3 Sound Generator Operation

This section explains the details of the Sound Generator.

18.3.1 Generating the tone

The tone signal is generated by the compare match signal of the SG0FH counter value with the value of the SG0FH buffer, followed by a by-two-divider. At each compare match, the counter is reset to zero.

Remember that the SG0FH counter is clocked by the output of the SG0FL counter.

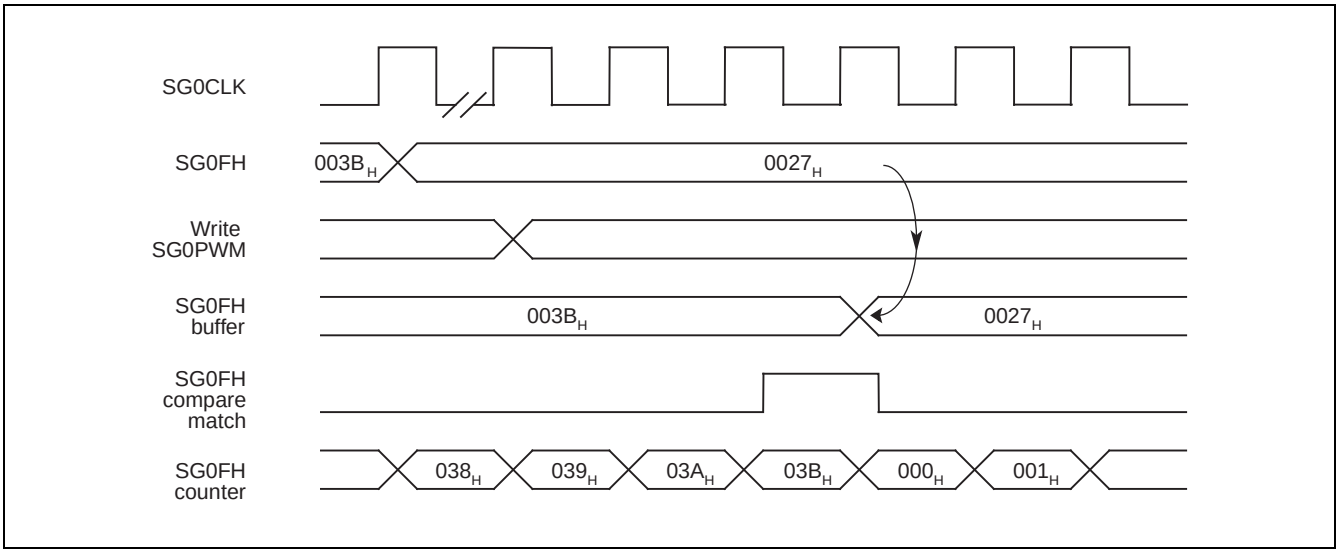
(1) Updating the frequency buffer values

The values of the frequency buffers can be changed by writing to the associated frequency registers SG0FL and SG0FH.

Changing the value of the SG0FL (equivalent to SG0F[15:0]) register would also yield a change of the PWM frequency, i.e. the sound volume. Therefore it is obligatory to write the correct PWM value to SG0PWM before a new SG0FL value is copied to the frequency buffers.

The following figure shows an example (not to scale).

Figure 18-5. Update Timing of the Frequency Buffers



If SG0FL is set to 01AEH and a 193 Hz tone is generated, as in the above example, the time span between writing to the SG0PWM register and updating the buffer can be up to 5.17 ms.

(2) Tone frequency calculation

The tone frequency can be calculated as:

$f_{tone} = f_{SG0CLK} / (([SG0FL\ buffer] + 1) \times ([SG0FH\ buffer] + 1) \times 2)$

where:

f_{SG0CLK} : Frequency of Sound Generator's input clock

$f_{SG0CLK} = f_{CLK} / 2$

[SG0FL buffer] : Contents of the SG0FL buffer

[SG0FH buffer] : Contents of the SG0FH buffer

Example

If:

– f_{CLK} = 20 MHz

– $f_{SG0CLK} = f_{CLK} / 2$ = 10 MHz

– [SG0FL buffer] = 255 (00FFH) (this yields a PWM frequency of 39.01 kHz)

– [SG0FH buffer] = 32 (0020H)

then:

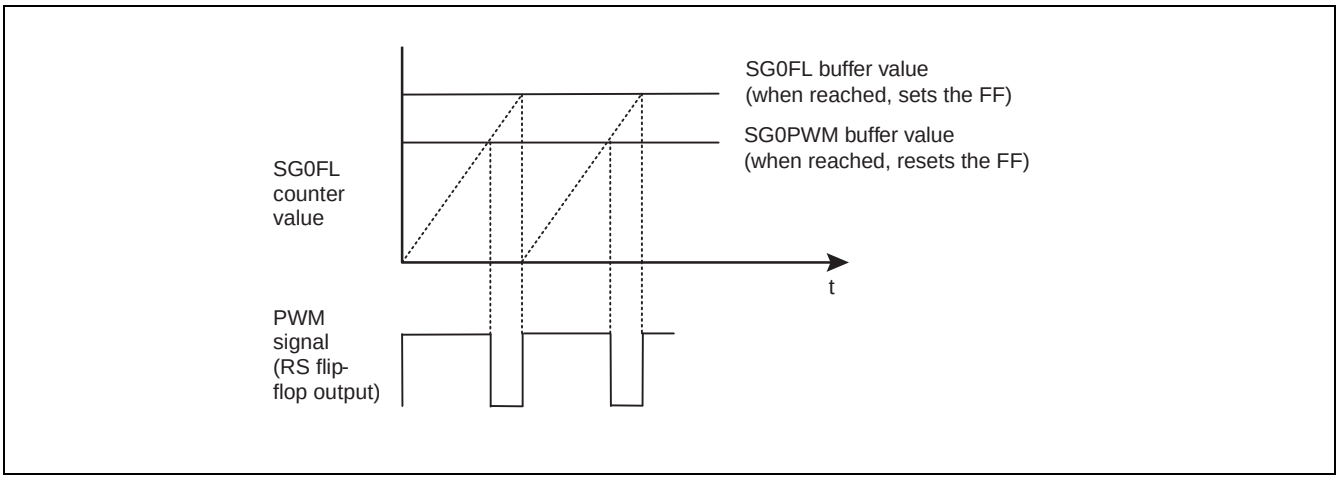
– f_{tone} = 592 Hz

Remark Note that the buffer contents can differ from the contents of the associated register until the next compare match.

18.3.2 Generating the volume information

The sound volume information is generated by comparing the SG0FL counter value with the contents of the SG0PWM volume buffer. An RS flipflop is set when the counter matches the SG0FL buffer and reset when the counter reaches the value of the volume buffer SG0PWM.

Figure 18-6. PWM Signal Generation



The duty cycle of the PWM signal is determined by the difference between the contents of the SG0FL counter buffer and the contents of the SG0PWM volume buffer. The larger the difference, the smaller the duty cycle.

The PWM signal is continually high when the value of the volume buffer is higher than the value of the frequency compare buffer.

Remark To achieve 100 % duty cycle for all PWM frequencies, SG0FL must not be set to a value above 1FEH.

The PWM signal is continually low when the value of the volume buffer is zero—the sound has stopped.

(1) Updating the volume buffer value

The value of the volume compare buffer can be changed by writing to the volume register SG0PWM.

- If the register is cleared by writing 0000H, the register value is copied to the volume compare buffer with the next rising edge of SG0CLK.
- As a result, the sound stops at the latest after one period of SG0CLK.
- If a non-zero value is written to the register, the buffer is updated with the next falling or rising edge of the tone frequency (match between SG0FH counter value and SG0FH buffer value).

(2) PWM calculations

PWM frequency

The PWM frequency is generated by the counter SG0FL. It can be calculated as:

$f_{PWM} = f_{SG0CLK} / (([SG0FL buffer] + 1)$

where:

f_{SG0CLK} : Frequency of Sound Generator's input clock

$f_{SG0CLK} = f_{CLK} / 2$

[SG0FL buffer] : Contents of the SG0FL buffer

Duty cycle

The duty cycle of the PWM signal is calculated as follows:

- If [SG0PWM buffer] > [SG0FL buffer]:
  Duty cycle = 100 %
- If 0 ≤ [SG0PWM buffer] ≤ [SG0FL buffer]:
  Duty cycle = [SG0PWM buffer] / ([SG0FL buffer] + 1)

where:

[SG0PWM buffer] : Contents of SG0PWM buffer

[SG0FL buffer] : Contents of SG0FL buffer

Example

If [SG0FL] is set to 240 (00F0H), the following table applies:

Table 18-2. Duty Cycle Calculation Example

[SG0PWM]	Calculation	Duty cycle [%]
01FFH		100
...		100
00F1H	241 / 241	100
00F0H	240 / 241	99.6
00EFH	239 / 241	99.2
...	...	...
0001H	1 / 241	0.41
0000H	0 / 241	0

The table shows, how the contents of register SG0FL affects the achievable volume resolution.

18.4 Sound Generator Application Hints

This section provides supplementary programming information.

18.4.1 Initialization

To enable the Sound Generator, set SG0CTL.PWR to 1. This connects the SG0 to the clock SG0CLK.

Check bit SG0CTL.OS.

When SG0CTL.OS is 0, the signal at pin SGO is a symmetrical square waveform with the frequency f_{tone}. When SG0CTL.OS is 1, the signal at pin SGO is composed of the tone signal and PWM pulses.

The frequency data registers SG0FL and SG0FH provide the buffer values for the counters. The combined value represents the frequency of the tone.

18.4.2 Start and stop sound

The sound is started by writing a non-zero value to the volume register SG0PWM.

Before starting the sound, all other register settings must be made.

The sound is stopped by writing 0000H to the volume register SG0PWM. The sound is stopped regardless of the current value of amplitude output or frequency output. Thus, the sound can be stopped quickly, even if a very low sound frequency is chosen.

18.4.3 Change sound volume

The sound volume is changed by writing a new value to register SG0PWM.

The new volume takes effect with the next edge of the tone pulse (rising or falling).

18.4.4 Generate special sounds

To generate special sounds (like blinker clicks etc.), frequency and volume can be changed simultaneously.

To change the frequency of a sound that has already started:

1. Write to the frequency registers SG0FL and SG0FH separately in 16-bit mode.
2. Write to the volume register SG0PWM.

CHAPTER 19 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR

19.1 Functions of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator has the following functions.

- 16 bits × 16 bits = 32 bits (Unsigned)
- 16 bits × 16 bits = 32 bits (Signed)
- 16 bits × 16 bits + 32 bits = 32 bits (Unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (Signed)
- 32 bits ÷ 32 bits = 32 bits, 32-bits remainder (Unsigned)

19.2 Configuration of Multiplier and Divider/Multiply-Accumulator

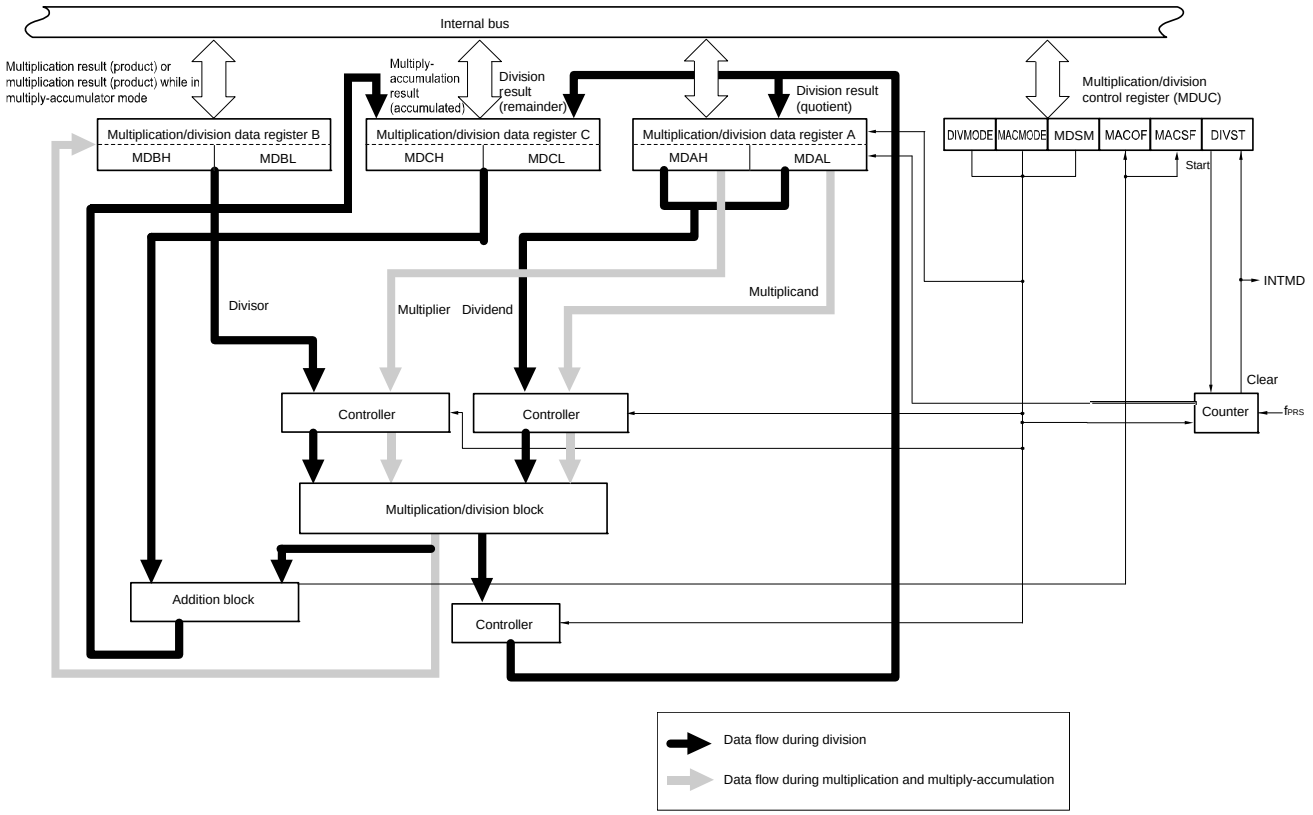
The multiplier and divider/multiply-accumulator consists of the following hardware.

Table 19-1. Configuration of Multiplier and Divider/Multiply-Accumulator

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 19-1 shows a block diagram of the multiplier and divider/multiply-accumulator.

Figure 19-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator



(1) Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode or multiply-accumulator mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

The MDAH and MDAL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 19-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol FFFF3H FFFF2H

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

Symbol FFFF1H FFFF0H

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

Cautions 1. Do not rewrite the MDAH and MDAL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL registers values read during division operation processing (when the MDUC register value is 81H or C1H) will not be guaranteed.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDAH and MDAL registers during operation execution.

Table 19-2. Functions of MDAH and MDAL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	MDAH: Multiplier (unsigned) MDAL: Multiplicand (unsigned)	–
Multiplication mode (signed) Multiply-accumulator mode (signed)	MDAH: Multiplier (signed) MDAL: Multiplicand (signed)	–
Division mode (unsigned)	MDAH: Dividend (unsigned) Higher 16 bits MDAL: Dividend (unsigned) Lower 16 bits	MDAH: Division result (unsigned) Higher 16 bits MDAL: Division result (unsigned) Lower 16 bits

(2) Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and multiply-accumulator mode, and set the divisor data in the division mode.

The MDBH and MDBL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 19-3. 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	MDBH 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) or multiply-accumulation operation processing. The operation result will be an undefined value.

2. Do not set the MDBH and MDBL registers to 0000H in the division mode. If they are set, the operation result will be an undefined value.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDBH and MDBL registers during operation execution.

Table 19-3. Functions of MDBH and MDBL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	–	MDBH: Multiplication result (product) (unsigned) Higher 16 bits MDBL: Multiplication result (product) (unsigned) Lower 16 bits
Multiplication mode (signed) Multiply-accumulator mode (signed)	–	MDBH: Multiplication result (product) (signed) Higher 16 bits MDBL: Multiplication result (product) (signed) Lower 16 bits
Division mode (unsigned)	MDBH: Divisor (unsigned) Higher 16 bits MDBL: Divisor (unsigned) Lower 16 bits	–

(3) Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers are used to store the accumulated result while in the multiply-accumulator mode or the remainder of the operation result while in the division mode. These registers are not used while in the multiplication mode.

The MDCH and MDCL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 19-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R/W

Symbol F00E3H F00E2H

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

Symbol F00E1H F00E0H

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

Cautions 1. The MDCH and MDCL registers values read during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) will not be guaranteed.

2. During multiply-accumulator processing, do not use software to rewrite the values of the MDCH and MDCL registers. If this is done, the operation result will be undefined.

3. The data is in the two's complement format in the multiply-accumulator mode (signed).

Table 19-4. Functions of MDCH and MDCL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned or signed)	–	–
Multiply-accumulator mode (unsigned)	MDCH: Initial accumulated value (unsigned) higher 16 bits MDCL: Initial accumulated value (unsigned) lower 16 bits	MDCH: accumulated value (unsigned) higher 16 bits MDCL: accumulated value (unsigned) lower 16 bits
Multiply-accumulator mode (signed)	MDCH: Initial accumulated value (signed) higher 16 bits MDCL: Initial accumulated value (signed) lower 16 bits	MDCH: accumulated value (signed) higher 16 bits MDCL: accumulated value (signed) lower 16 bits
Division mode (unsigned)	–	MDCH: Remainder (unsigned) higher 16 bits MDCL: Remainder (unsigned) lower 16 bits

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during multiply-accumulation

  <Multiplier A> <Multiplier B> < accumulated value > < accumulated result >

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) + MDC (bits 31 to 0) = [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

  (The multiplication result is stored in the MDBH (bits 15 to 0) and MDBL (bits 15 to 0).)

- Register configuration during division

  <Dividend> <Divisor>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

  <Quotient> <Remainder>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

19.3 Register Controlling Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator is controlled by using the multiplication/division control register (MDUC).

(1) Multiplication/division control register (MDUC)

The MDUC register is an 8-bit register that controls the operation of the multiplier and divider/multiply-accumulator.

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

Reset signal generation clears this register to 00H.

Figure 19-5. 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	MACMODE	0	0	MDSM	MACOF	MACSF	DIVST

DIVMODE	MACMODE	MDSM	Operation mode selection
0	0	0	Multiplication mode (unsigned) (default)
0	0	1	Multiplication mode (signed)
0	1	0	Multiply-accumulator mode (unsigned)
0	1	1	Multiply-accumulator mode (signed)
1	0	0	Division mode (unsigned), generation or not generation of a division completion interrupt (INTMD)
1	1	0	Division mode (unsigned), not generation of a division completion interrupt (INTMD)
Other than the above			Setting prohibited

MACOF	Overflow flag of multiply-accumulation result (accumulated value)
0	No overflow
1	With over flow
<Set condition> • For the multiply-accumulator mode (unsigned) The bit is set when the accumulated value goes outside the range from 00000000h to FFFFFFFFh. • For the multiply-accumulator mode (signed) The bit is set when the result of adding a positive product to a positive accumulated value exceeds 7FFFFFFFh and is negative, or when the result of adding a negative product to a negative accumulated value exceeds 80000000h and is positive.	

MACSF	Sign flag of multiply-accumulation result (accumulated value)
0	The accumulated value is positive.
1	The accumulated value is negative.
Multiply-accumulator mode (unsigned):	The bit is always 0.
Multiply-accumulator mode (signed):	The bit indicates the sign bit of the accumulated value.

DIVST[Note]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Note The DIVST bit can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) the DIVST bit. The DIVST bit is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to multiplication/division data register A (MDAH, MDAL), respectively.

Cautions 1. Do not rewrite the DIVMODE, MDSM bits during operation processing (while the DIVST bit is 1). If it is rewritten, the operation result will be an undefined value.

2. The DIVST bit cannot be cleared (0) by using software during division operation processing (while the DIVST bit is 1).

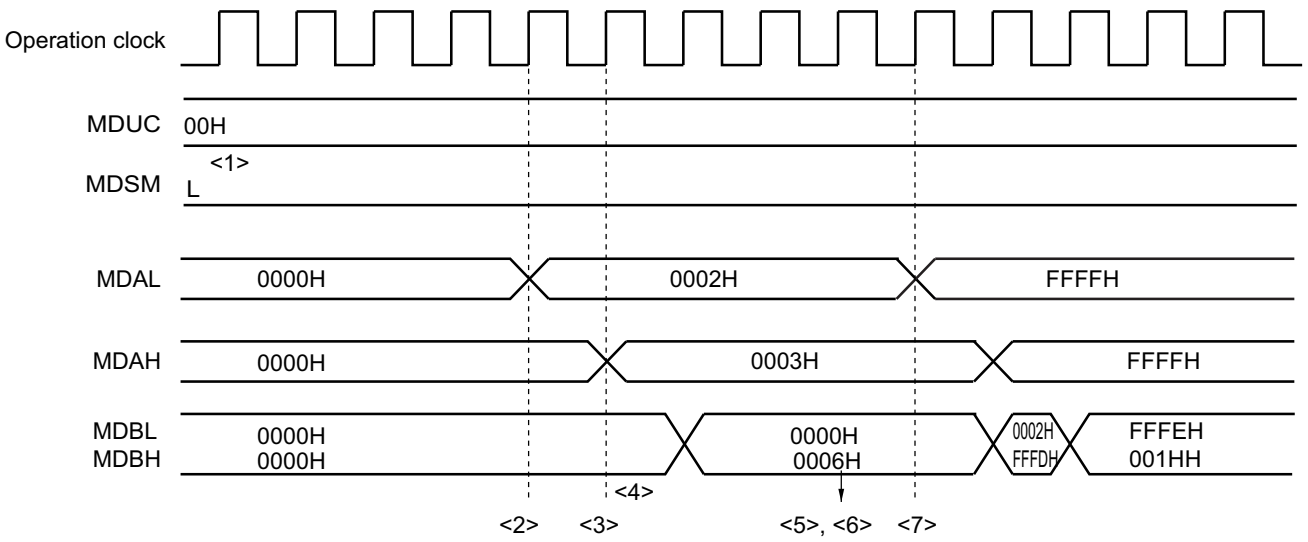
19.4 Operations of Multiplier and Divider/Multiply-Accumulator

19.4.1 Multiplication (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 00H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> The next time multiplication, division or multiply-accumulation is performed, start with the initial settings of each step.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 18-6.

Figure 19-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)



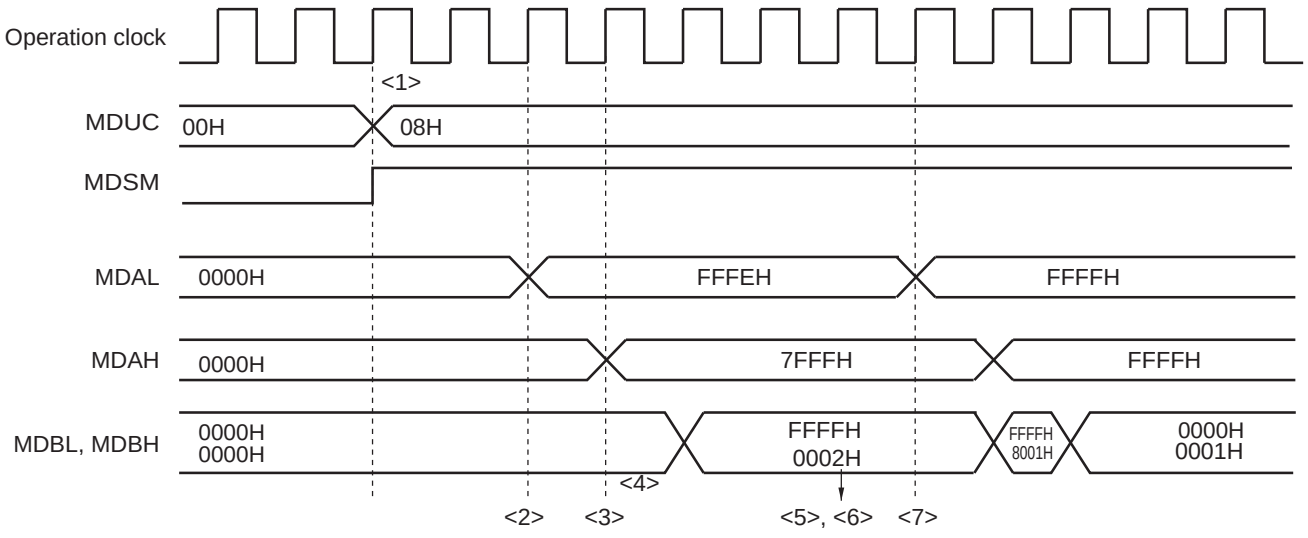
19.4.2 Multiplication (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 08H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> The next time multiplication, division, or multiply-accumulation is performed, start with the initial settings of each step.

Caution The data is in the two's complement format in multiplication mode (signed).

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 19-7.

Figure 19-7. Timing Diagram of Multiplication (Signed) Operation (–2 × 32767 = –65534)

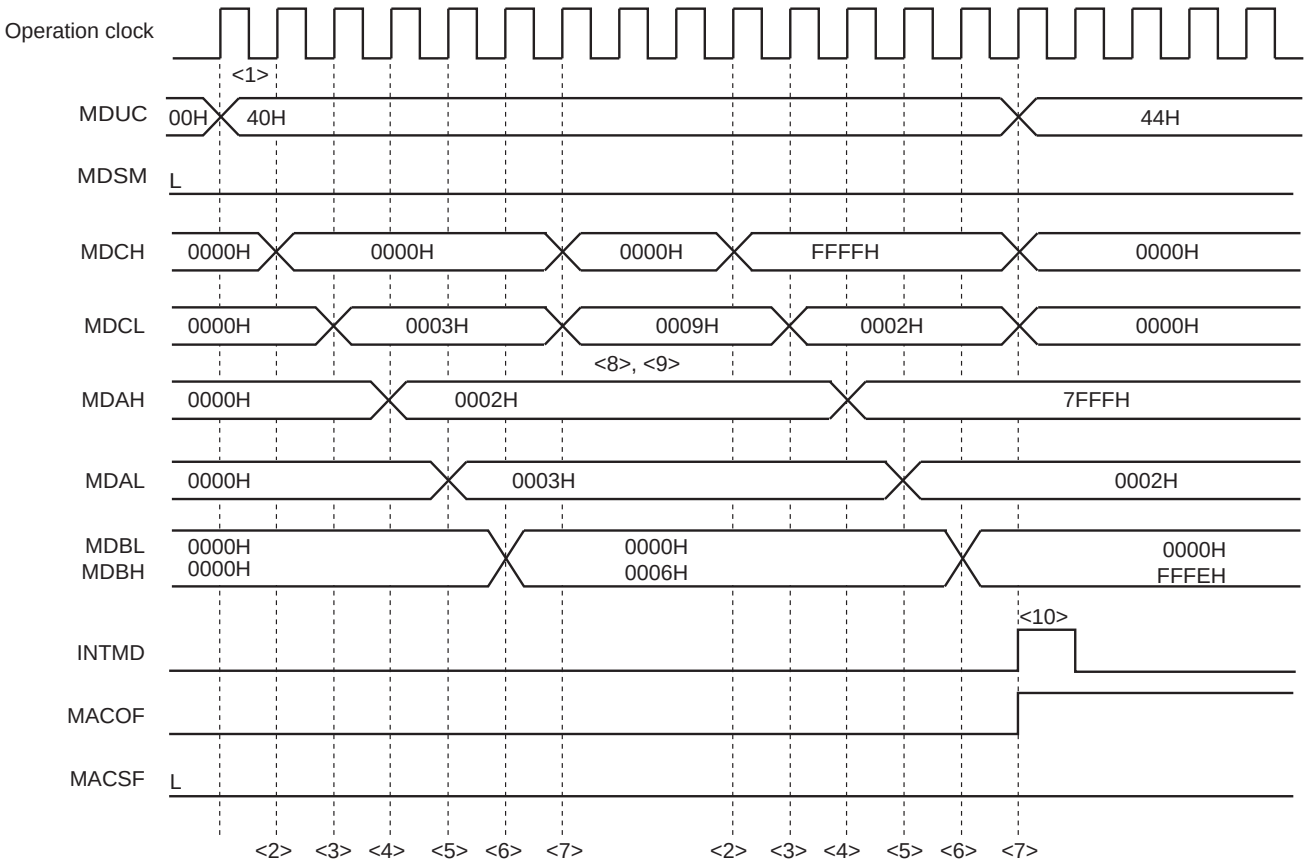


19.4.3 Multiply-accumulation (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 40H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (L) (MDCL).
  - <3> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (H) (MDCH).
  - <4> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <5> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2>, <3>, and <4>. Multiplication operation is automatically started when the multiplier is set to the MDAH register, respectively.)
- During operation processing
  - <6> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <7> After <6>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<5>).)
- Operation end
  - <8> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <9> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <8> and <9>.)
  - (<10> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation
  - <11> The next time multiplication, division or multiply-accumulation is performed, start with the initial settings of each step.

Remark Steps <1> to <10> correspond to <1> to <10> in Figure 19-8.

Figure 19-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation
(2 × 3 + 3 = 9 → 32767 × 2 + 4294901762 = 0 (over flow generated))



19.4.4 Multiply-accumulation (signed) operation

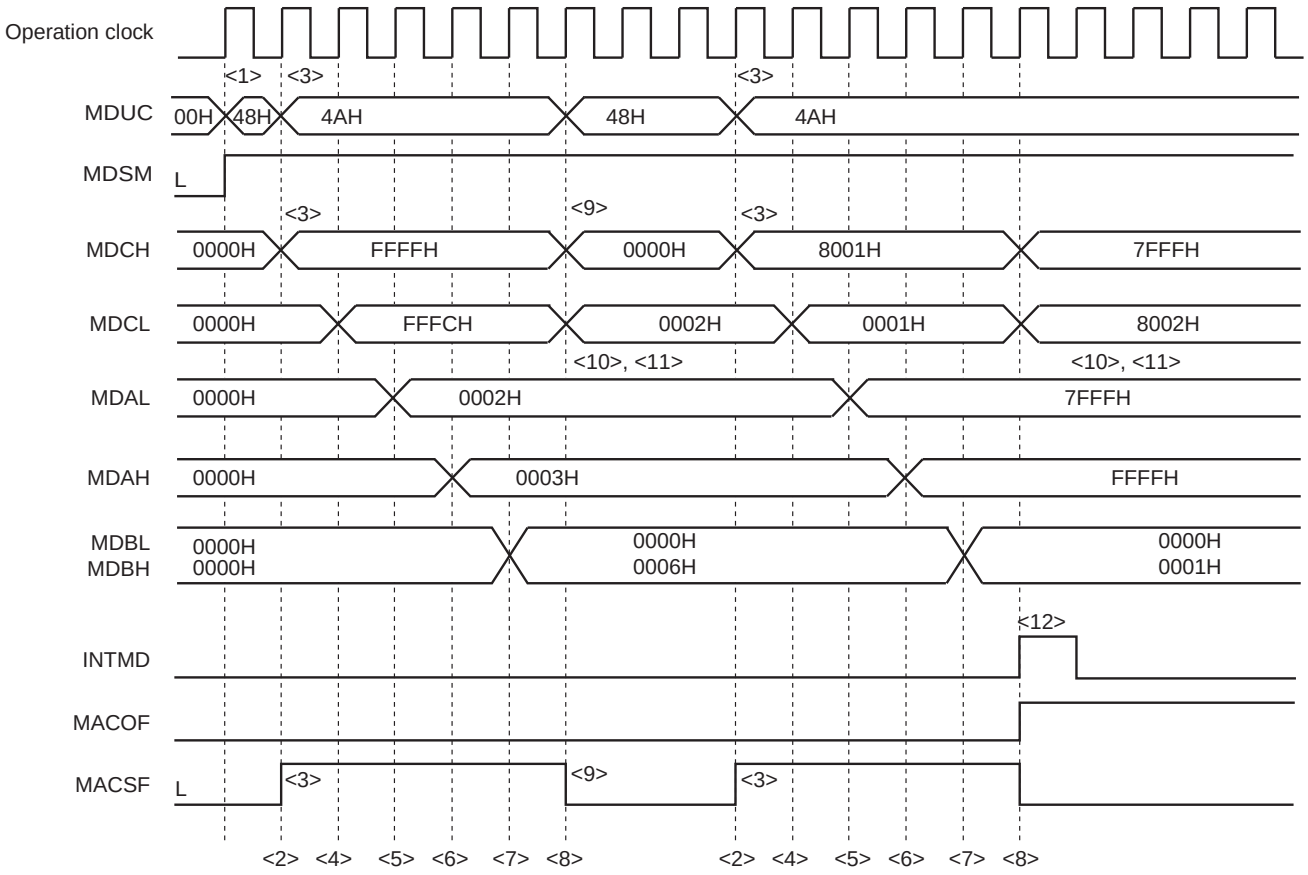
- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 48H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (H) (MDCH).
    (<3> If the accumulated value in the MDCH register is negative, the MACSF bit is set to 1.)
  - <4> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (L) (MDCL).
  - <5> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <6> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2>, <4>, and <5>. Multiplication operation is automatically started when the multiplier is set to the MDAH register, respectively.)
- During operation processing
  - <7> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <8> After <7>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<6>).)
- Operation end
  - <9> If the accumulated value stored in the MDCL and MDCH registers is positive, the MACSF bit is cleared to 0.
  - <10> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <11> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <10> and <11>.)

  (<12> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation
  - <13> The next time multiplication, division or multiply-accumulation is performed, start with the initial settings of each step.

Caution The data is in the two's complement format in multiply-accumulation (signed) operation.

Remark Steps <1> to <11> correspond to <1> to <11> in Figure 19-9.

Figure 19-9. Timing Diagram of Multiply-Accumulation (signed) Operation
(2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (overflow occurs.))



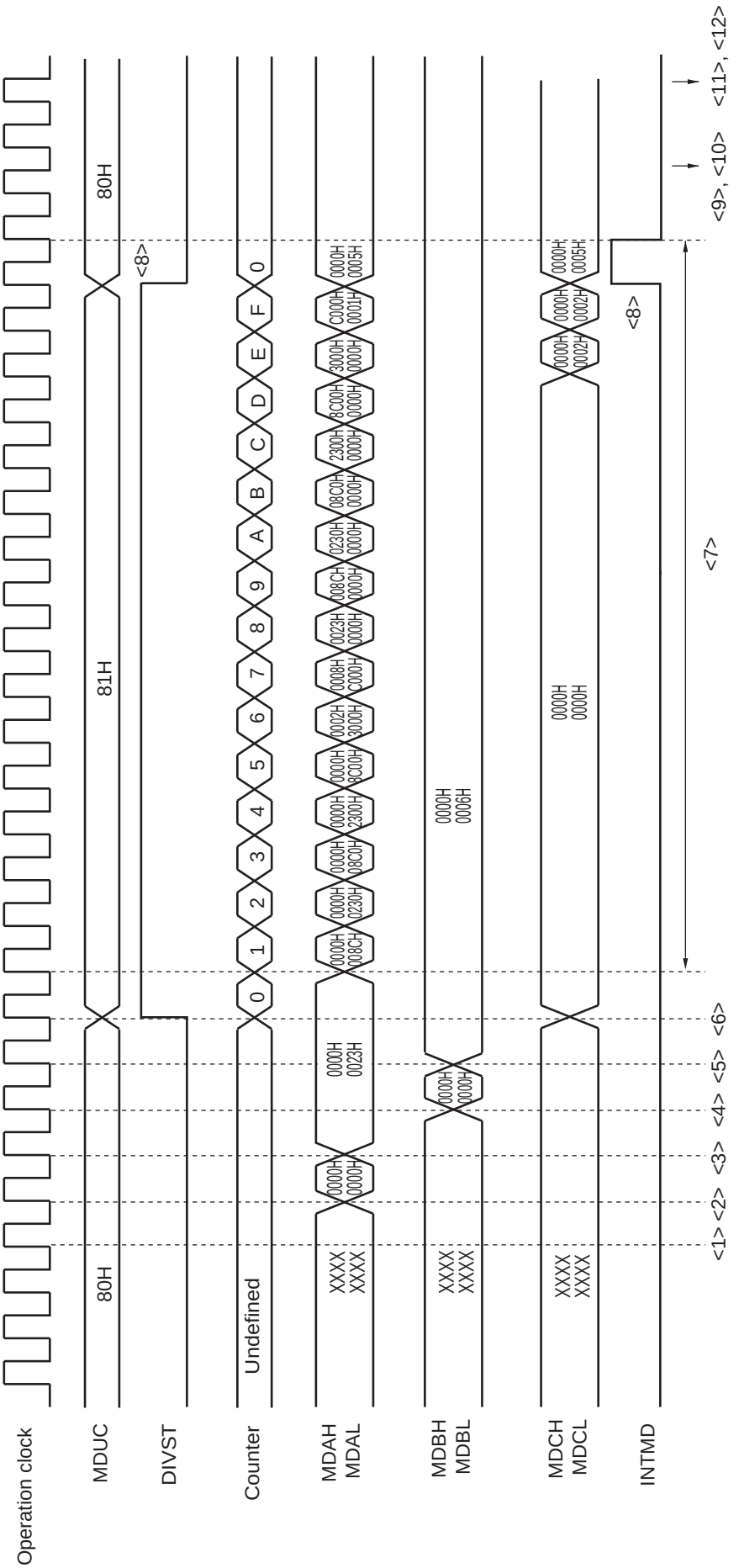
19.4.5 Division operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 80H.
  - <2> Set the dividend (higher 16 bits) to multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of the MDUC register to 1.
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether the DIVST bit has been cleared

    (The read values of the MDBL, MDBH, MDCL, and MDCH registers during operation processing are not guaranteed.)
- Operation end
  - <8> The DIVST bit is cleared and the operation ends. At this time, an interrupt request signal (INTMD) is generated if the operation was performed with MACMODE = 0.
  - <9> Read the quotient (lower 16 bits) from the MDAL register.
  - <10> Read the quotient (higher 16 bits) from the MDAH register.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from multiplication/division data register C (H) (MDCH).
    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  - <13> The next time multiplication division or multiply-accumulation is performed, start with the initial settings of each step.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 19-10.

Figure 19-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 20 DMA CONTROLLER

<R>

	48-pin	64-pin	80-pin	100-pin	128-pin
	R5F10CGx/ R5F10DGx	R5F10CLx/ R5F10DLx	R5F10CMx/ R5F10DMx	R5F10TPx/ R5F10DPx	R5F10DSx
DMA	2 channels			4 channels	

The RL78/D1A 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.

20.1 Functions of DMA Controller

<R>

- Number of DMA channels: 2 channels (48, 64 or 80-pin products)
  4 channels (100 or 128-pin products)
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 1024 times
- Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- Transfer mode: Single-transfer mode
- Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface
    (CSI00, CSI01, CSI10, UART0, and UART1)
  - Timer (channel 0, 1, 3, 5, 7, 10 to 15, 17, 20, 21, 23, 25, 27)
  - I^2C
- 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

20.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 20-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0 to 3 (DSA0 to DSA3) • DMA RAM address registers 0 to 3 (DRA0 to DRA3)
Count register	• DMA byte count registers 0 to 3 (DBC0 to DBC3)
Control registers	• DMA mode control registers 0 to 3 (DMC0 to DMC3) <R> • DMA trigger selection register (DMATSEL) Note • DMA operation control register 0 to 3 (DRC0 to DRC3) • DMA all-channel forced wait register (DWAITALL)

<R> **Note** 128-pin products only.

Remark Channels 2 and 3 are incorporated in 100 and 128-pin products.

(1) DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DSAn register can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Figure 20-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1), F0170H (DSA2), F0171H (DSA3) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

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 can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, the DRAn register stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DRAn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 20-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) , F0172H, F0173H (DRA2), F0174H, F0175H (DRA3) — After reset: 0000H — R/W

DRA0H: FFFB3H
DRA1H: FFFB5H
DRA2H: F0173H
DRA3H: F0175H

DRA0L: FFFB2H
DRA1L: FFFB4H
DRA2L: F0172H
DRA3L: F0174H

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

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.

The DBCn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 20-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W
F0176H, F0177H (DBC2), F0178H, F0179H (DBC3)

DBC0H: FFFB7H
DBC1H: FFFB9H
DBC2H: F0177H
DBC3H: F0179H

DBC0L: FFFB6H
DBC1L: FFFB8H
DBC2L: F0176H
DBC3L: F0178H

	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)

20.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)
- DMA all-channel forced wait register (DWAITALL)

Remark n: DMA channel number (n = 0 to 3)

(1) DMA mode control register n (DMCn)

The DMCn register is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of the DMCn register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 20-4. Format of DMA Mode Control Register n (DMCn) (1/3)

Address: FFFBAH (DMC0), FFFBBH (DMC1), F017AH (DMC2), F017BH (DMC3) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to the STGn bit when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn[Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of the DWAITn bit to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of the DWAITn bit is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

2. When DMA transfer is held pending while using two or more DMA channels, be sure to hold the DMA transfer pending for all channels (by setting the DWAIT0, DWAIT1, DWAIT2, and DWAIT3 bits to 1).

Remark n: DMA channel number (n = 0 to 3)

Figure 20-4. Format of DMA Mode Control Register n (DMCn) (2/3)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

(When n = 0 or 1)

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note 1]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTTM00	End of timer channel 0 count or capture end interrupt
0	0	1	0	INTTM01	End of timer channel 1 count or capture end interrupt
0	0	1	1	INTTM03	End of timer channel 3 count or capture end interrupt
0	1	0	0	INTTM05	End of timer channel 5 count or capture end interrupt
0	1	0	1	INTTM07	End of timer channel 7 count or capture end interrupt
0	1	1	0	INTTM10	End of timer channel 10 count or capture end interrupt
0	1	1	1	INTTM11	End of timer channel 11 count or capture end interrupt
1	0	0	0	INTTM12	End of timer channel 12 count or capture end interrupt
1	0	0	1	INTIIC11	IIC11 end of transfer interrupt
1	0	1	0	INTLT0	LIN UART0 (UARTF0) transmission interrupt
1	0	1	1	INTLR0	LIN UART0 (UARTF0) reception interrupt
<R> 1	1	0	0	INTCSI00/INTLCDB[Note 2]	CSI00 end of transfer interrupt
1	1	0	1	INTCSI01	CSI01 end of transfer interrupt
1	1	1	0	INTAD	A/D conversion end interrupt
1	1	1	1	Setting prohibited	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

<R> **2.** 128-pin products only. For details, see **20.3 (2) DMA trigger selection register (DMATSEL)**.

Remark n: DMA channel number (n = 0, 1)

Figure 20-4. Format of DMA Mode Control Register n (DMCn) (3/3)

Address: F017AH (DMC2), F017BH (DMC3) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

(When n = 2 or 3)

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note 1]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTTM13	End of timer channel 13 count or capture end interrupt
0	0	1	0	INTTM14	End of timer channel 14 count or capture end interrupt
0	0	1	1	INTTM15	End of timer channel 15 count or capture end interrupt
0	1	0	0	INTTM17	End of timer channel 17 count or capture end interrupt
0	1	0	1	INTTM20	End of timer channel 20 count or capture end interrupt
0	1	1	0	INTTM21	End of timer channel 21 count or capture end interrupt
0	1	1	1	INTTM23	End of timer channel 23 count or capture end interrupt
1	0	0	0	INTTM25	End of timer channel 25 count or capture end interrupt
1	0	0	1	INTTM27	End of timer channel 27 count or capture end interrupt
1	0	1	0	INTLT1	LIN UART1 (UARTF1) transmission interrupt
1	0	1	1	INTLR1	LIN UART1 (UARTF1) reception interrupt
1	1	0	0	INTCSI00	CSI00 end of transfer interrupt
1	1	0	1	INTCSI01	CSI01 end of transfer interrupt
1	1	1	0	INTAD	A/D conversion end interrupt
<R> 1	1	1	1	INTCSI10/INTLCDB[Note 2]	CSI10 end of transfer interrupt

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

<R> **2.** 128-pin products only. For details, see **20.3 (2) DMA trigger selection register (DMATSEL)**.

Remark n: DMA channel number (n = 2, 3)

<R> **(2) DMA trigger selection register (DMATSEL) (128-pin products only)**

The LCD BUS I/F interrupt signal (INTLCDB) is treated as a DMA trigger source. This trigger source shares with INTCSI00/INTCSI10 as one DMA trigger source. This register is used to switch the DMA trigger source.

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

Reset signal generation clears this register to 00H.

Figure 20-5. Format of DMA trigger selection register (DMATSEL))

Address: F001FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
DMATSEL	0	0	0	0	0	0	DMATSL23	DMATSL01

DMATSL23	DMA2/3 trigger (IFCn3 to IFCn0=1111) seletion (n = 2 or 3)
0	INTCSI10
1	INTLCDB

DMATSL01	DMA0/1 trigger (IFCn3 to IFCn0=1100) seletion (n = 0 or 1)
0	INTCSI00
1	INTLCDB

(3) DMA operation control register n (DRCn)

The DRCn register is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 20-6. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1), F017CH (DRC2), F017DH (DRC3) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by the IFCn3 to IFCn0 bits is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set the DSTn bit to 0 and then the DENn bit to 0 (for details, refer to 20.5.5 Forced termination by software).

Remark n: DMA channel number (n = 0 to 3)

(4) DMA all-channel forced wait register (DWAITALL)

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 (PRVARI) of DWAITALL can be rewritten while DMA transfer is in progress without affecting the current transfer.

Figure 20-7. Format of DMA all-channel forced wait register (DWAITALL)

Address: F017FH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DWAITALL	PRVARI	0	0	0	0	0	0	DWAITALL0

PRVARI	Transfer channel priority order
0	Priority order fixed (CH0 → CH1 → CH2 → CH3)
1	Priority order can be changed.

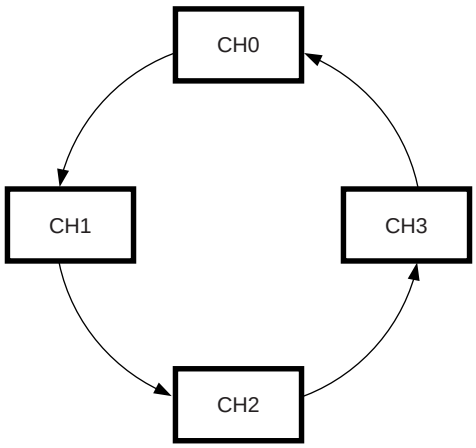
DWAITALL0	All-channel forced wait
0	All channels are operating normally.
1	All channels are being forced to wait.

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.

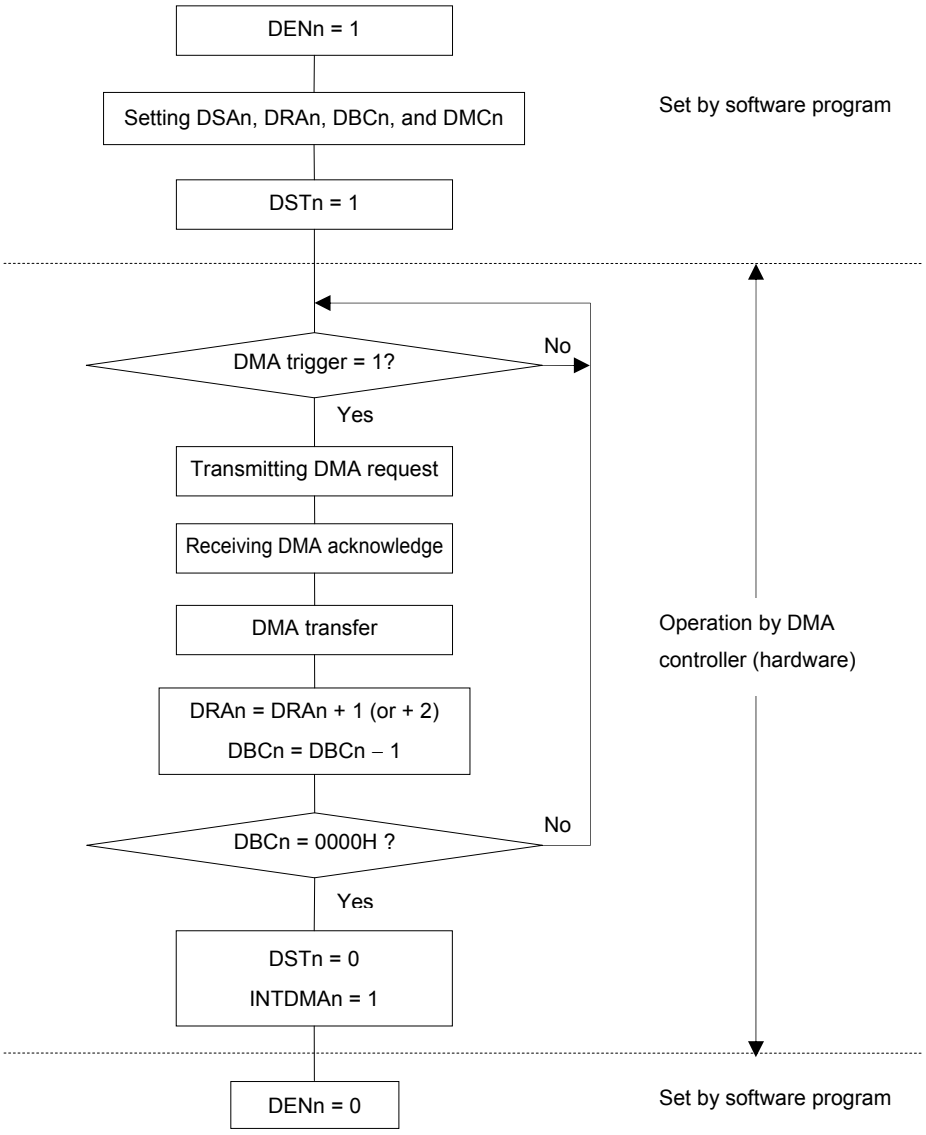


20.4 Operation of DMA Controller

20.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set the DENn bit to 1. Use 80H to write with an 8-bit manipulation instruction.
<2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to DMA SFR address register n (DSAn), DMA RAM address register n (DRAn), DMA byte count register n (DBCn), and DMA mode control register n (DMCn).
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.
<4> When a software trigger (STGn) or a start source trigger specified by the IFCn3 to IFCn0 bits is input, a DMA transfer is started.
<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).
<6> Stop the operation of the DMA controller by clearing the DENn bit to 0 when the DMA controller is not used.

Figure 20-7. Operation Procedure



Remark n: DMA channel number (n = 0 to 3)

20.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of DMA mode control register n (DMCn).

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

20.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, DMA byte count register n (DBCn) and DMA RAM address register n (DRAn) hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0 to 3)

20.5 Example of Setting of DMA Controller

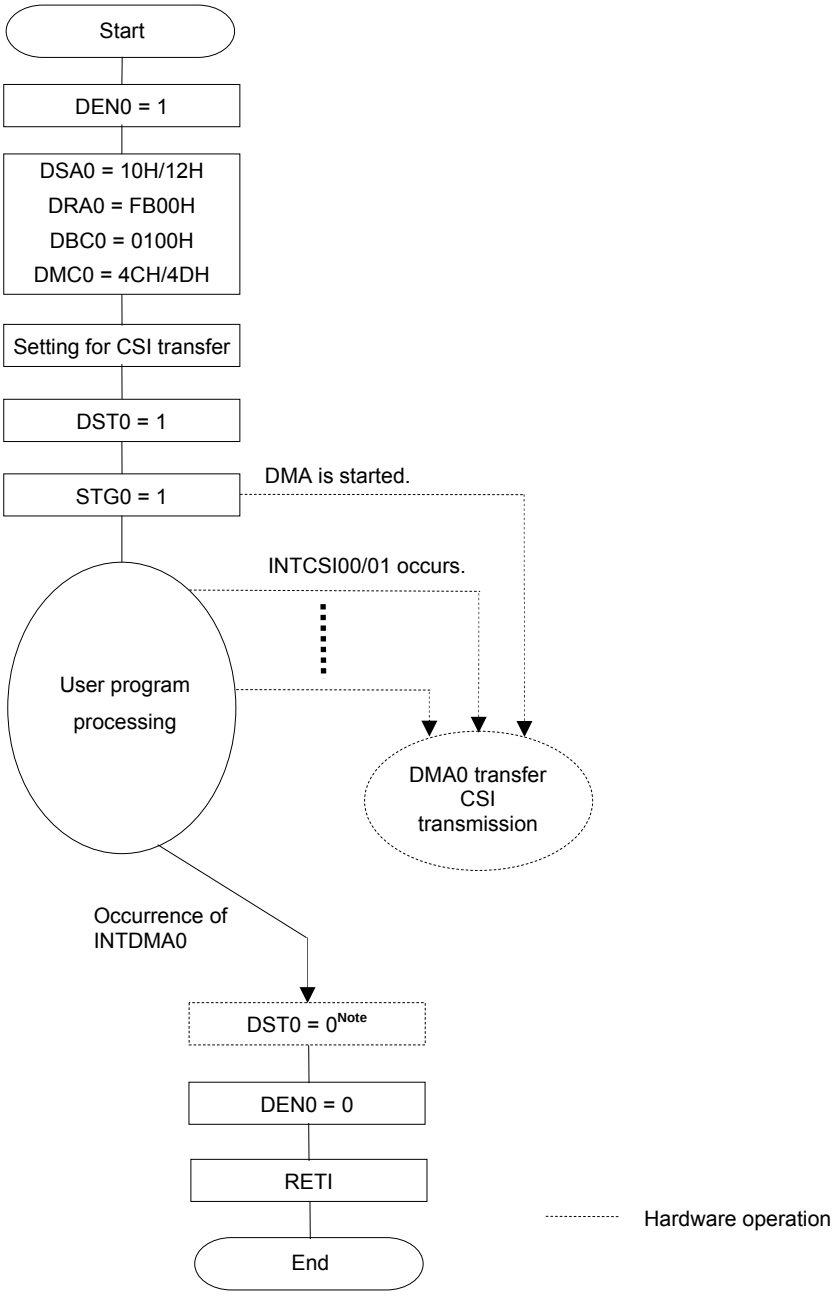
20.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI00 or CSI01 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 or INTCSI01 (software trigger (STG0) only for the first start source)
- <R> Interrupt of CSI00 is specified by IFC03 to IFC00 = 1100B and DMATSL01 = 0B. In case of CSI01, interrupt of CSI01 is specified by IFC03 to IFC00 = 1101B.
- When INTCSI00 is specified as a DMA start source, transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF10H of the serial data register (SDR00L) of CSI. When INTCSI01 is specified, transfers FFB00H to FFBFFH (256 bytes) of <R> RAM to FFF10H of the serial data register (SDR01L) of CSI.

Remark IFC03 to IFC00: Bits 3 to 0 of DMA mode control registers 0 (DMC0)
<R> DMATSL01 : Bit 0 of DMA trigger selection register (DMATSEL)

Figure 20-8. 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 the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **20.5.5 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.

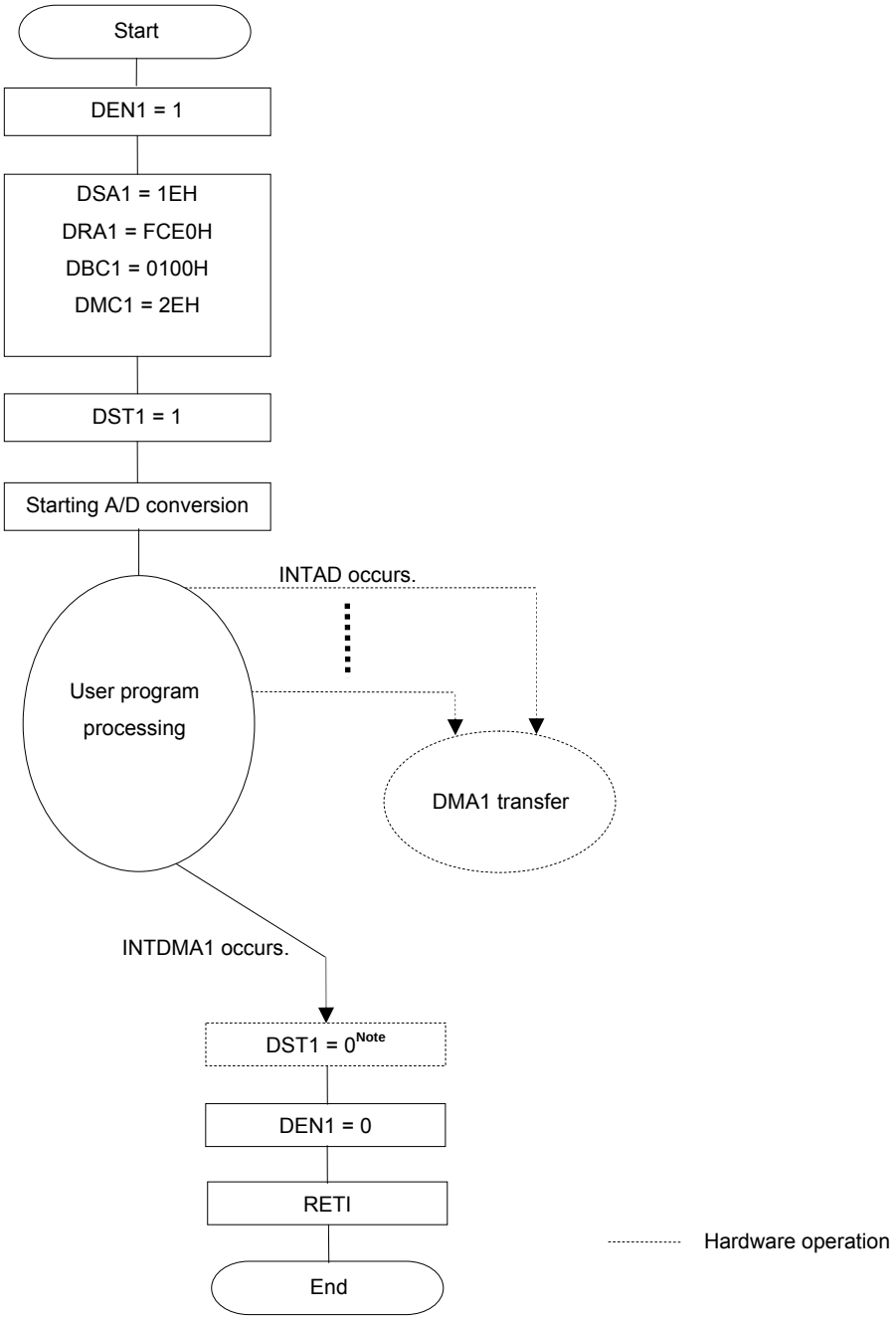
20.5.2 Consecutive capturing of A/D conversion results

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 = 1110B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register (ADCR) to 512 bytes of FFCE0H to FFEDFH of RAM.

Remark IFC13 to IFC10: Bits 3 to 0 of DMA mode control registers 1 (DMC1)

Figure 20-9. Example of Setting of Consecutively Capturing A/D Conversion Results



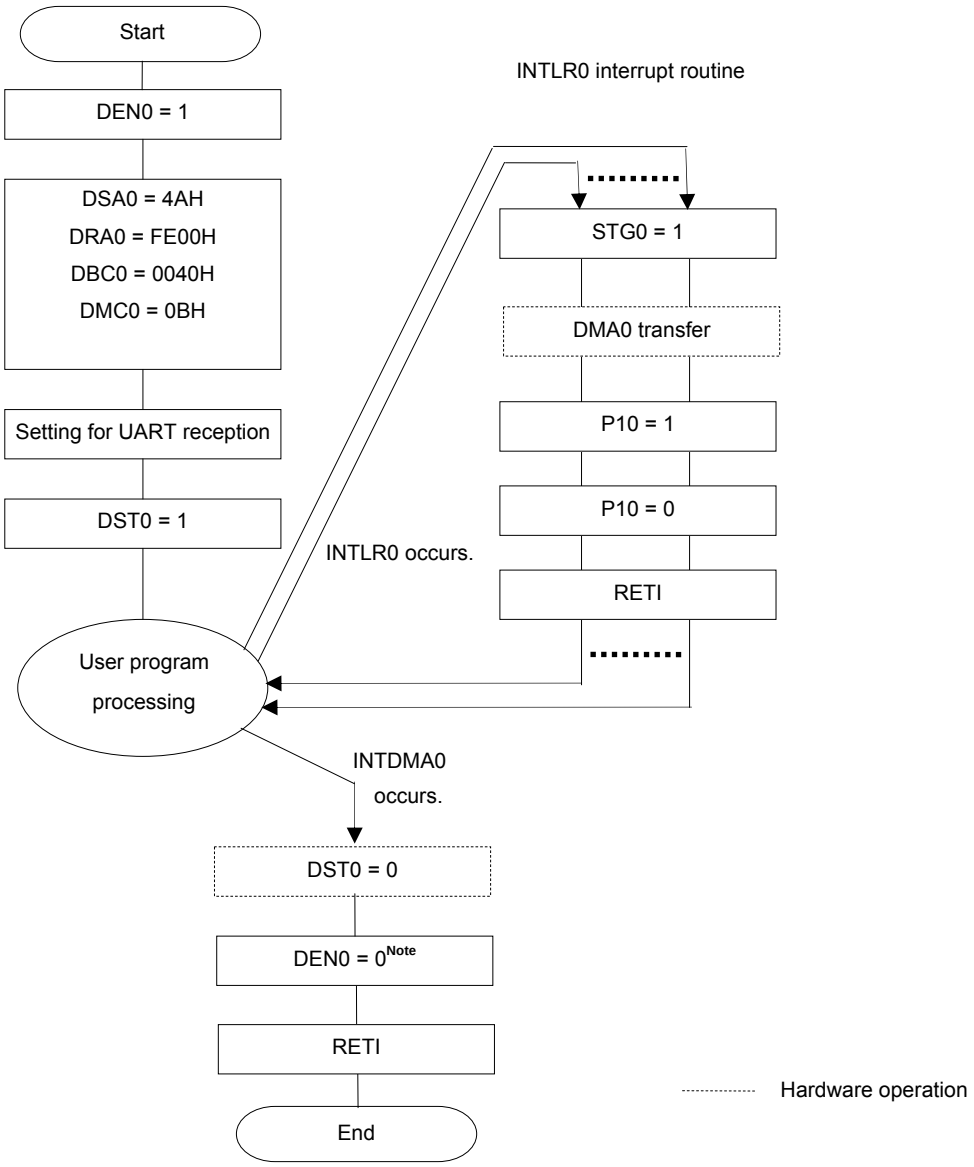
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set the DST1 bit to 0 and then the DEN1 bit to 0 (for details, refer to **20.5.5 Forced termination by software**).

20.5.3 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 UARTF0 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 FFF4AH of UART receive data register 0 (UF0RX) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 20-10. 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 the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **20.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTLR0) can be used to start DMA for data reception.

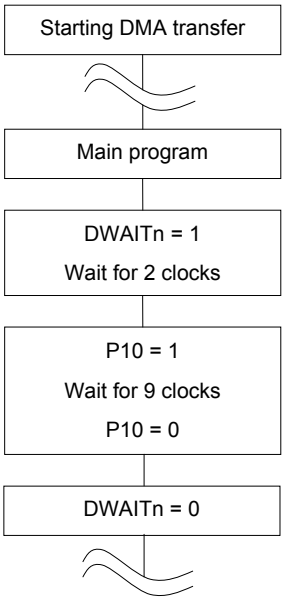
20.5.4 Holding DMA transfer pending by DWAITn bit

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation of the set system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P10 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting the DWAITn bit to 1.

After setting the DWAITn bit to 1, it takes two clocks until a DMA transfer is held pending.

Figure 20-11. Example of Setting for Holding DMA Transfer Pending by DWAITn Bit



Caution When DMA transfer is held pending while using two or more DMA channels, be sure to held the DMA transfer pending for all channels (by setting DWAIT0, DWAIT1, DWAIT2, and DWAIT3 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

Remarks 1. n: DMA channel number (n = 0 to 3)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

20.5.5 Forced termination by software

After the DSTn bit is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and the DSTn bit is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

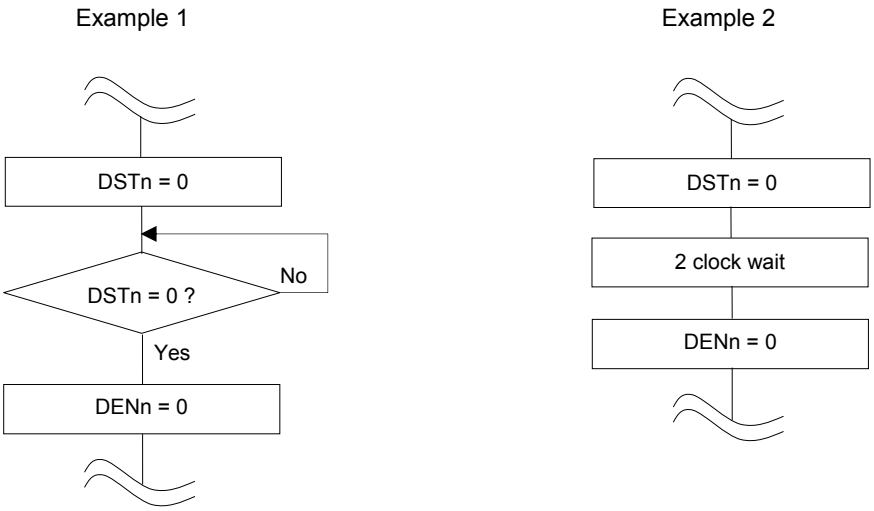
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using two or more DMA channels>

- To forcibly terminate DMA transfer by software when using two or more DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAITn bits of all using channels to 1. Next, clear the DWAITn bits of all using channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 20-12. Forced Termination of DMA Transfer (1/2)



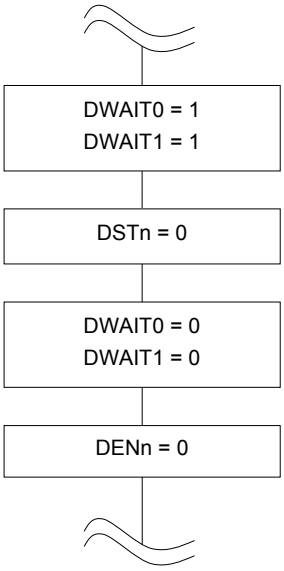
Remarks 1. n: DMA channel number (n = 0 to 3)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

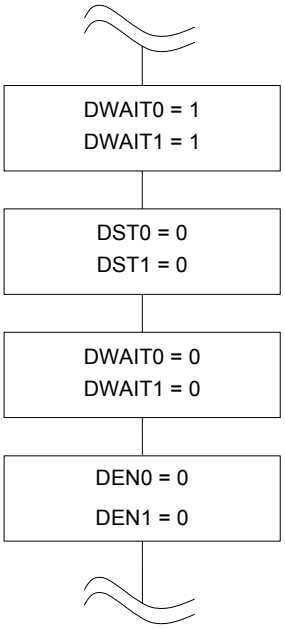
Figure 20-12. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

20.6 Cautions on Using DMA Controller

(1) Priority of DMA

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two or more DMA requests are generated at the same time, however, their priority are DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3. If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 20-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 20.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 20-3. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

(4) DMA pending instruction

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- Write instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L,MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H,PR12L, PR12H, and PR13L each.
- Instruction for accessing the data flash memory

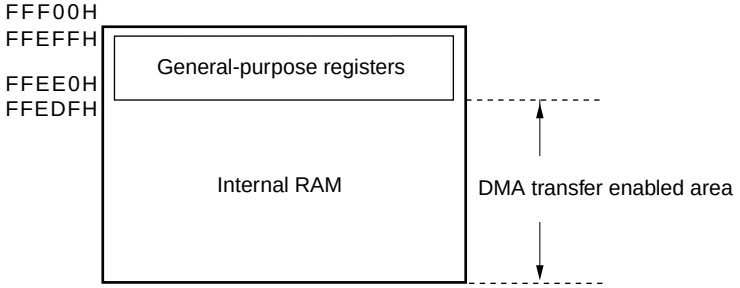
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DMA RAM address register n (DRAn) is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM
  The data of that address is lost.

- In mode of transfer from RAM to SFR
  Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.



(6) Operation if instructions for accessing the data flash area

- Because DMA transfer is suspended to access to the data flash area, be sure to add the DMA pending instruction.
  If the data flash area is accessed after an next instruction execution from start of DMA transfer, a 3-clock wait will be inserted to the next instruction.

  Instruction 1
  DMA transfer
  Instruction 2 ←The wait of three clock cycles occurs.
  MOV A, ! DataFlash area

- The data flash should be read in either of following ways.
  - Use the flash library provided by Renesas (EEL (Pack01) version V1.13 or later).
  - Stop the DMA transfer before reading.

CHAPTER 21 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.

The number of interrupt sources differs, depending on the product.

<R>

		R5F10CGx	R5F10DGx	R5F10CLx	R5F10DLx	R5F10CMx	R5F10CMx	R5F10TPx R5F10DPx	R5F10DPJ	R5F10DSx
Maskable interrupts	External	6		8						
	Internal	39	43	53	26	42	26	49	53	53

21.1 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 21-1**.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

21.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to eight reset sources (see **Table 21-1**). 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.

<R>

Table 21-1. Interrupt Source List (1/2)

Type	Default priority Note1	Interrupt Source		Internal/ External	Vector table address	Basic Configuration type	R5F10CGx	R5F10DGx	R5F10CLx	R5F10DLx	R5F10CMx	R5F10DMx	R5F10DPx, R5F10TPJ	R5F10DPJ	R5F10DSx
		Name	Trigger												
Mask-able	0	INTWDTI	Watchdog timer interval Note2 (75% of overflow time+1/2f_{IL})	Internal	00004H	(A)	√	√	√	√	√	√	√	√	√
	1	INTLVI	Voltage detection Note 3	Internal	00006H		√	√	√	√	√	√	√	√	√
	2	INTP0	External Interrupt 0	External	00008H	(B)	–	–	√	√	√	√	√	√	√
	3	INTP1	External Interrupt 1	External	0000AH		√	√	√	√	√	√	√	√	√
	4	INTP2	External Interrupt 2	External	0000CH		√	√	√	√	√	√	√	√	√
	5	INTP3	External Interrupt 3	External	0000EH		√	√	√	√	√	√	√	√	√
	6	INTP4	External Interrupt 4	External	00010H		√	√	√	√	√	√	√	√	√
	7	INTP5	External Interrupt 5	External	00012H		√	√	√	√	√	√	√	√	√
	8	INTCLM	Clock Monitor interrupt	Internal	00014H	(A)	√	√	√	√	√	√	√	√	√
	9	INTCSI00	End of CSI00 communication	Internal	00016H		√	√	√	√	√	√	√	√	√
		INTST0	UART0 of SAU0 transmission interrupt				–	–	–	–	–	–	–	–	√
	10	INTCSI01	End of CSI01 communication	Internal	00018H		√	√	√	√	√	√	√	√	√
		INTSR0	UART0 of SAU0 reception interrupt				–	–	–	–	–	–	–	–	√
	11	INTDMA0	End of DMA0 transfer	Internal	0001AH		√	√	√	√	√	√	√	√	√
	12	INTDMA1	End of DMA1 transfer	Internal	0001CH		√	√	√	√	√	√	√	√	√
	13	INTRTC	Fixed cycle signal of RTC /Alarm match detection	Internal	0001EH		√	√	√	√	√	√	√	√	√
	14	INTIT	Interval timer interrupt	Internal	00020H		√	√	√	√	√	√	√	√	√
	15	INTLT0	LIN-UART0(UARTF0) transmission interrupt	Internal	00022H		–	–	√	√	√	√	√	√	√
	16	INTLR0	LIN-UART0(UARTF0) reception interrupt	Internal	00024H		–	–	√	√	√	√	√	√	√
	17	INTLS0	LIN-UART0(UARTF0) reception status interrupt	Internal	00026H		–	–	√	√	√	√	√	√	√
	18	INTPLR0	LIN-UART0(UARTF0) reception pin input	External	00028H	(B)	–	–	√	√	√	√	√	√	√
	19	INTSG	Interrupt from SG in ALD mode	Internal	0002AH	(A)	√	√	√	√	√	√	√	√	√
	20	INTTM00	End of TAU 00 count or capture interrupt	Internal	0002CH		√	√	√	√	√	√	√	√	√
	21	INTTM01	End of TAU 01 count or capture interrupt	Internal	0002EH		√	√	√	√	√	√	√	√	√
	22	INTTM02	End of TAU 02 count or capture interrupt	Internal	00030H		√	√	√	√	√	√	√	√	√
	23	INTTM03	End of TAU 03 count or capture interrupt	Internal	00032H		√	√	√	√	√	√	√	√	√
	24	INTAD	End of A/D conversion	Internal	00034H		√	√	√	√	√	√	√	√	√
	25	INTLT1	LIN-UART1(UARTF1) transmission interrupt	Internal	00036H		√	√	√	√	√	√	√	√	√
	26	INTLR1	LIN-UART1(UARTF1) reception interrupt	Internal	00038H		√	√	√	√	√	√	√	√	√
	27	INTLS1	LIN-UART1(UARTF1) reception status interrupt	Internal	0003AH		√	√	√	√	√	√	√	√	√
	28	INTPLR1	LIN-UART1(UARTF1) reception pin input	External	0003CH	(B)	√	√	√	√	√	√	√	√	√
	29	INTCSI10	End of CSI10 communication	Internal	0003EH	(A)	–	–	–	–	–	–	√	√	√
	30	INTIIC11	End of IIC11 communication	Internal	00040H		√	√	√	√	√	√	√	√	√
	31	INTTM04	End of TAU 04 count or capture interrupt	Internal	00042H		√	√	√	√	√	√	√	√	√
	32	INTTM05	End of TAU 05 count or capture interrupt	Internal	00044H		√	√	√	√	√	√	√	√	√
	33	INTTM06	End of TAU 06 count or capture interrupt	Internal	00046H		√	√	√	√	√	√	√	√	√
	34	INTTM07	End of TAU 07 count or capture interrupt	Internal	00048H		√	√	√	√	√	√	√	√	√

Notes 1. 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.

2. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

3 When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

<R>

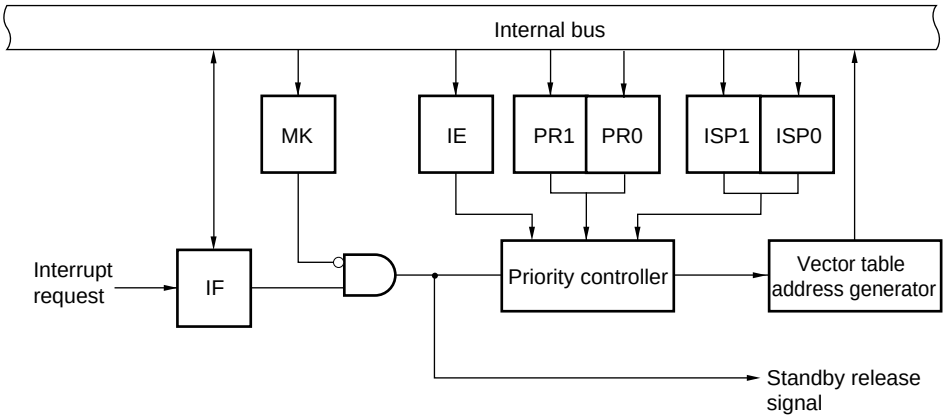
Table 21-1. Interrupt Source List (2/2)

Type	Default priority Note1	Interrupt Source		Internal/ External	Vector table address	Basic Configuration type	R5F10CGx	R5F10DGx	R5F10CLx	R5F10DLx	R5F10CMx	R5F10DMx	R5F10DPx, R5F10TPJ	R5F10DPJ	R5F10DSx
		Name	Trigger												
Mask-able	35	INTC1ERR	CAN1 error interrupt	Internal	0004AH	(A)	–	–	–	–	–	–	–	√	√
	36	INTC1WUP	CAN1 wakeup	Internal	0004CH		–	–	–	–	–	–	–	√	√
	37	INTC0ERR	CAN0 error interrupt	Internal	0004EH		–	√	–	√	–	√	√	√	√
	38	INTC0WUP	CAN0 wakeup	Internal	00050H		–	√	–	√	–	√	√	√	√
	39	INTC0REC	CAN0 reception completion	Internal	00052H		–	√	–	√	–	√	√	√	√
	40	INTC0TRX	CAN0 transmission completion	Internal	00054H		–	√	–	√	–	√	√	√	√
	41	INTTM10	End of TAU 10 count or capture interrupt	Internal	00056H		√	√	√	√	√	√	√	√	√
	42	INTTM11	End of TAU 11 count or capture interrupt	Internal	00058H		√	√	√	√	√	√	√	√	√
	43	INTTM12	End of TAU 12 count or capture interrupt	Internal	0005AH		√	√	√	√	√	√	√	√	√
	44	INTTM13	End of TAU 13 count or capture interrupt	Internal	0005CH		√	√	√	√	√	√	√	√	√
	45	INTMD	End of division operation/Overflow occur	Internal	0005EH		√	√	√	√	√	√	√	√	√
	46	INTC1REC	CAN1 reception completion	Internal	00060H		–	–	–	–	–	–	–	√	√
	47	INTFL	End of sequencer interrupt(Flash programming)	Internal	00062H		√	√	√	√	√	√	√	√	√
	48	INTC1TRX	CAN1 transmission completion	Internal	00064H		–	–	–	–	–	–	–	√	√
	49	INTTM14	End of TAU 14 count or capture interrupt	Internal	00066H		√	√	√	√	√	√	√	√	√
	50	INTTM15	End of TAU 15 count or capture interrupt	Internal	00068H		√	√	√	√	√	√	√	√	√
	51	INTTM16	End of TAU 16 count or capture interrupt	Internal	0006AH		√	√	√	√	√	√	√	√	√
	52	INTTM17	End of TAU 17 count or capture interrupt	Internal	0006CH		√	√	√	√	√	√	√	√	√
	53	INTTM20	End of TAU 20 count or capture interrupt	Internal	0006EH		√	√	√	√	√	√	√	√	√
	54	INTTM21	End of TAU 21 count or capture interrupt	Internal	00070H		√	√	√	√	√	√	√	√	√
	55	INTTM22	End of TAU 22 count or capture interrupt	Internal	00072H		√	√	√	√	√	√	√	√	√
	56	INTTM23	End of TAU 23 count or capture interrupt	Internal	0074H		√	√	√	√	√	√	√	√	√
	57	INTTM24	End of TAU 24 count or capture interrupt	Internal	0076H		√	√	√	√	√	√	√	√	√
	58	INTTM26	End of TAU 26 count or capture interrupt	Internal	0078H		√	√	√	√	√	√	√	√	√
	59	INTDMA2	End of DMA2 transfer	Internal	007AH		–	–	–	–	–	–	√	√	√
	60	INTDMA3	End of DMA3 transfer	Internal	007CH		–	–	–	–	–	–	√	√	√
Soft-ware	-	BRK	Execution of BRK instruction	-	007EH	(C)	√	√	√	√	√	√	√	√	√
Reset	-	RESET	$\overline{\text{RESET}}$ pin input	-	0000H	-	√	√	√	√	√	√	√	√	√
		POR	Power-on-reset				√	√	√	√	√	√	√	√	√
		LVD	Voltage detection Note				√	√	√	√	√	√	√	√	√
		WDT	Overflow of watchdog timer				√	√	√	√	√	√	√	√	√
		TRAP	Execution of illegal instruction				√	√	√	√	√	√	√	√	√
		IAW	Illegal-memory access				√	√	√	√	√	√	√	√	√
		RPE	RAM parity error				√	√	√	√	√	√	√	√	√

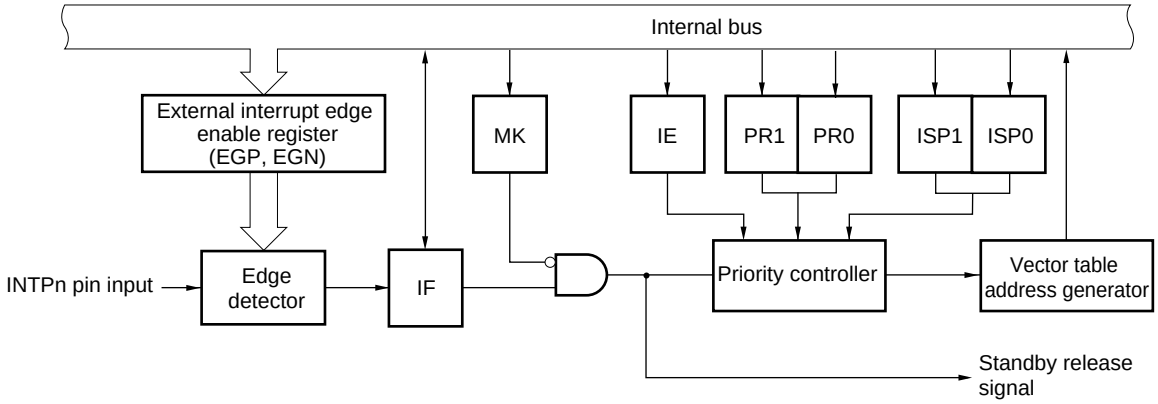
Note The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

Figure 21-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



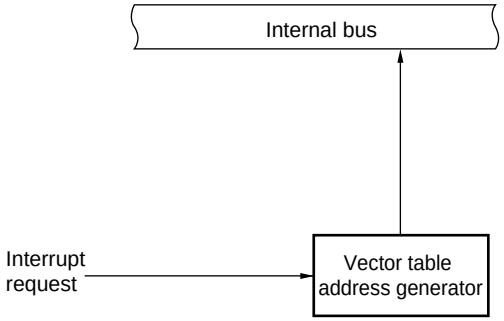
(B) External maskable interrupt (INTPn)



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1
INTPn = INTP0 to INTP5, INTPLR0, INTPLR1

Figure 21-1. Basic Configuration of Interrupt Function (2/2)

(C) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

21.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, 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 register 0 (EGP0)
- External interrupt falling edge enable register 0 (EGN0)
- Program status word (PSW)

Table 21-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 21-2. Flags Corresponding to Interrupt Request Sources

Name	Interrupt Request Flag				Interrupt Mask Flag				Priority Specification Flag0				Priority Specification Flag1			
		Register		Bit		Register		Bit		Register		Bit		Register		Bit
INTWDTI	WDTIIF	IF0	IF0L	0	WDTIMK	MK0	MK0L	0	WDTIPR0	PR00	PR00L	0	WDTIPR1	PR10	PR10L	0
INTLVI	LVIIF			1	LVIMK			1	LVIPR0			1	LVIPR1			1
INTP0	PIF0			2	PMK0			2	PPR00			2	PPR10			2
INTP1	PIF1			3	PMK1			3	PPR01			3	PPR11			3
INTP2	PIF2			4	PMK2			4	PPR02			4	PPR12			4
INTP3	PIF3			5	PMK3			5	PPR03			5	PPR13			5
INTP4	PIF4			6	PMK4			6	PPR04			6	PPR14			6
INTP5	PIF5			7	PMK5			7	PPR05			7	PPR15			7
INTCLM	CLMIF		IF0H	0	CLMMK		MK0H	0	CLMPR0		PR00H	0	CLMPR1		PR10H	0
INTCSI00	CSIIF00			1	CSIMK00			1	CSIPR000			1	CSIPR100			1
<R> INTST0	STIF0				STMK0				STPR00				STPR10			
INTCSI01	CSIIF01			2	CSIMK01			2	CSIPR001			2	CSIPR101			2
<R> INTSR0	SRIF0				SRMK0				SRPR00				SRPR10			
INTDMA0	DMAIF0			3	DMAMK0			3	DMAPR00			3	DMAPR10			3
INTDMA1	DMAIF1			4	DMAMK1			4	DMAPR01			4	DMAPR11			4
INTRTC	RTCIF			5	RTCMK			5	RTCPR0			5	RTCPR1			5
INTIT	ITIF			6	ITMK			6	ITPR0			6	ITPR1			6
INTLT0	LTIF0			7	LTMK0			7	LTPR00			7	LTPR10			7
INTLR0	LRIF0	IF1	IF1L	0	LRMK0	MK1	MK1L	0	LRPR00	PR01	PR01L	0	LRPR10	PR11	PR11L	0
INTLS0	LSIF0			1	LSMK0			1	LSPR00			1	LSPR10			1
INTPLR0	PIFLR0			2	PMKLR0			2	PPR0LR0			2	PPR1LR0			2
INTSG	SGIF			3	SGMK			3	SGPR0			3	SGPR1			3
INTTM00	TMIF00			4	TMMK00			4	TMPR000			4	TMPR100			4
INTTM01	TMIF01			5	TMMK01			5	TMPR001			5	TMPR101			5
INTTM02	TMIF02			6	TMMK02			6	TMPR002			6	TMPR102			6
INTTM03	TMIF03			7	TMMK03			7	TMPR003			7	TMPR103			7
INTAD	ADIF		IF1H	0	ADMK		MK1H	0	ADPR0		PR01H	0	ADPR1		PR11H	0
INTLT1	LTIF1			1	LTMK1			1	LTPR01			1	LTPR11			1
INTLR1	LRIF1			2	LRMK1			2	LRPR01			2	LRPR11			2
INTLS1	LSIF1			3	LSMK1			3	LSPR01			3	LSPR11			3
INTPLR1	PIFLR1			4	PMKLR1			4	PPR0LR1			4	PPR1LR1			4
INTCSI10	CSIIF10			5	CSIMK10			5	CSIPR010			5	CSIPR110			5
INTIIC11	IICIF11			6	IICMK11			6	IICPR011			6	IICPR111			6
INTTM04	TMIF04			7	TMMK04			7	TMPR004			7	TMPR104			7
INTTM05	TMIF05	IF2	IF2L	0	TMMK05	MK2	MK2L	0	TMPR005	PR02	PR02L	0	TMPR105	PR12	PR12L	0
INTTM06	TMIF06			1	TMMK06			1	TMPR006			1	TMPR106			1
INTTM07	TMIF07			2	TMMK07			2	TMPR007			2	TMPR107			2
INTC1ERR	C1ERRIF			3	C1ERRMK			3	C1ERRPR0			3	C1ERRPR1			3
INTC1WUP	C1WUPIF			4	C1WUPMK			4	C1WUPPR0			4	C1WUPPR1			4
INTC0ERR	C0ERRIF			5	C0ERRMK			5	C0ERRPR0			5	C0ERRPR1			5
INTC0WUP	C0WUPIF			6	C0WUPMK			6	C0WUPPR0			6	C0WUPPR1			6
INTC0REC	C0RECIF			7	C0RECMK			7	C0RECPR0			7	C0RECPR1			7
INTC0TRX	C0TRXIF		IF2H	0	C0TRXMK		MK2H	0	C0TRXPR0		PR02H	0	C0TRXPR1		PR12H	0
INTTM10	TMIF10			1	TMMK10			1	TMPR010			1	TMPR110			1
INTTM11	TMIF11			2	TMMK11			2	TMPR011			2	TMPR111			2
INTTM12	TMIF12			3	TMMK12			3	TMPR012			3	TMPR112			3
INTTM13	TMIF13			4	TMMK13			4	TMPR013			4	TMPR113			4
INTMD	MDIF			5	MDMK			5	MDPR0			5	MDPR1			5
INTC1REC	C1RECIF			6	C1RECMK			6	C1RECPR0			6	C1RECPR1			6
INTFL	FLIF			7	FLMK			7	FLPR0			7	FLPR1			7
INTC1TRX	C1TRXIF	IF3	IF3L	0	C1TRXMK	MK3	MK3L	0	C1TRXPR0	PR03	PR03L	0	C1TRXPR1	PR13	PR13L	0
INTTM14	TMIF14			1	TMMK14			1	TMPR014			1	TMPR114			1
INTTM15	TMIF15			2	TMMK15			2	TMPR015			2	TMPR115			2
INTTM16	TMIF16			3	TMMK16			3	TMPR016			3	TMPR116			3
INTTM17	TMIF17			4	TMMK17			4	TMPR017			4	TMPR117			4
INTTM20	TMIF20			5	TMMK20			5	TMPR020			5	TMPR120			5
INTTM21	TMIF21			6	TMMK21			6	TMPR021			6	TMPR121			6
INTTM22	TMIF22			7	TMMK22			7	TMPR022			7	TMPR122			7
INTTM23	TMIF23		IF3H	0	TMMK23		MK3H	0	TMPR023		PR03H	0	TMPR123		PR13H	0
INTTM24	TMIF24			1	TMMK24			1	TMPR024			1	TMPR124			1
INTTM26	TMIF26			2	TMMK26			2	TMPR026			2	TMPR126			2
INTDMA2	DMAIF2			3	DMAMK2			3	DMAPR02			3	DMAPR12			3
INTDMA3	DMAIF3			4	DMAMK3			4	DMAPR03			4	DMAPR13			4
-	0			5	1			5	1			5	1			5
-	0			6	1			6	1			6	1			6
-	0			7	1			7	1			7	1			7

(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.

The IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, and IF3H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, the IF2L and IF2H registers, and the IF3L and IF3H registers 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 21-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (R5F10DSx) (1/2)

<R>

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	ITIF	RTCIF	DMAIF1	DMAIF0	CSIIF01 SRIF0	CSIIF00 STIF0	CLMIF

<R>

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	SGIF	PIFLR0	LSIF0	LRIF0

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	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	C1WUPIF	C1ERRIF	TMIF07	TMIF06	TMIF05

Figure 21-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (R5F10DSx) (2/2)

<R>

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	FLIF	C1RECIF	MDIF	TMIF13	TMIF12	TMIF11	TMIF10	C0TRXIF

Address: FFFD2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	TMIF22	TMIF21	TMIF20	TMIF17	TMIF16	TMIF15	TMIF14	C1TRXIF

Address: FFFD3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF3H	0	0	0	DMAIF3	DMAIF2	TMIF26	TMIF24	TMIF23

xxIFx	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

<R> **Cautions 1. The above is the bit layout for the R5F10DSx. The available bits differ depending on the product. For details about the bits available for each product, see Table 21-1 and 21-2. Be sure to clear bits that are not available to 0.**

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(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.

The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, and MK3H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, the MK2L and MK2H registers, and the MK3L and MK3H registers 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 21-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (R5F10DSx)

<R>

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	ITMK	RTCMK	DMAMK1	DMAMK0	CSIMK01 SRMK0	CSIMK00 STMK0	CLMMK

<R>

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	SGMK	PMKLR0	LSMK0	LRMK0

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	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	C1WUPMK	C1ERRMK	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	FLMK	C1RECMK	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	C1TRXMK

Address: FFFD7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK3H	1	1	1	DMAMK3	DMAMK2	TMMK26	TMMK24	TMMK23

xxMKx	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The above is the bit layout for the R5F10DSx. The available bits differ depending on the product. For details about the bits available for each product, see Table 21-1 and 21-2. 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, or 2H).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, the PR12L and PR12H registers, and the PR13L and PR13H registers are combined to form 16-bit registers PR00, PR01, PR02, 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 21-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (R5F10DSx) (1/3)

<R>

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	ITPR0	RTCPR0	DMAPR01	DMAPR00	CSIPR001	CSIPR000	CLMPR0

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	LTPR10	ITPR1	RTCPR1	DMAPR11	DMAPR10	CSIPR101 SRPR10	CSIPR100 STPR10	CLMPR1

<R>

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	SGPR0	PPR0LR0	LSPR00	LRPR00

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	SGPR1	PPR1LR0	LSPR10	LRPR10

Figure 21-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (R5F10DSx) (2/3)

<R>

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	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	IICPR111	CSIPR110	PPR1LR1	LSPR11	LRPR11	LTPR11	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	C0RECPR0	C0WUPPR0	C0ERRPR0	C1WUPPR0	C1ERRPR0	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	C0RECPR1	C0WUPPR1	C0ERRPR1	C1WUPPR1	C1ERRPR1	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	FLPR0	C1RECPR0	MDPR0	TMPR013	TMPR012	TMPR011	TMPR010	C0TRXPR0

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	FLPR1	C1RECPR1	MDPR1	TMPR113	TMPR112	TMPR111	TMPR110	C0TRXPR1

Figure 21-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (R5F10DSx) (3/3)

<R>

Address: FFFDAH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	TMPR022	TMPR021	TMPR020	TMPR017	TMPR016	TMPR015	TMPR014	C1TRXPR0

Address: FFFDEH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	TMPR122	TMPR121	TMPR120	TMPR117	TMPR116	TMPR115	TMPR114	C1TRXPR1

Address: FFFDBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR03H	1	1	1	DMAPR03	DMAPR02	TMPR026	TMPR024	TMPR023

Address: FFFDFH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR13H	1	1	1	DMAPR13	DMAPR12	TMPR126	TMPR124	TMPR123

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)

<R> **Caution The above is the bit layout for the R5F10DSx. The available bits differ depending on the product. For details about the bits available for each product, see Table 20-1 and 20-2. Be sure to set bits that are not available to 1.**

<R> **(4) External interrupt rising edge enable register 0 (EGP0) and external interrupt falling edge enable register 0 (EGN0)**

These registers specify the valid edge for external interrupt, INTP0 to INTP5, INTPLR0, and INTPLR1.

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

Reset signal generation clears these registers to 00H.

Figure 21-5. Format of External Interrupt Rising Edge Enable Register 0 (EGP0) and External Interrupt Falling Edge Enable Register 0 (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	Valid edge selection of external interrupt
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 20-3 shows the ports corresponding to the EGPn and EGNn bits.

<R> **Table 21-3. Ports Corresponding to EGPn and EGNn bits**

Register name	Bit	External interrupt name	Edge detection port	128-pin	100-pin	80-pin	64-pin	48-pin
EGP0, EGN0	0	INTP0	P17	√	√	√	√	–
	1	INTP1	P60	√	√	√	√	√
	2	INTP2	P12	√	√	√	√	√
	3	INTP3	P61	√	√	√	√	√
	4	INTP4	P10	√	√	√	√	√
	5	INTP5	P137	√	√	√	√	√
	6	INTPLR0	P70	√	√	√	√	–
			P70 or P14	√	√	√	√	–
	7	INTPLR1	P11	√	√	√	√	√
			P11 or P132	√	√	–	–	–

Caution Select the port mode by clearing the EGPn and EGNn bits to 0 because an edge may be detected when the external interrupt function is switched to the port function.

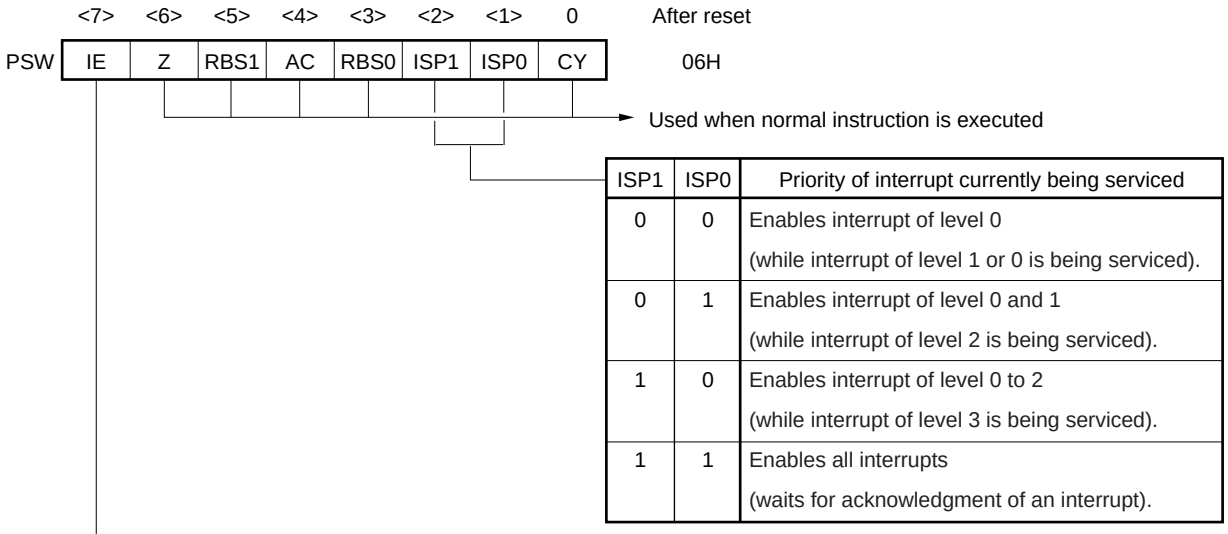
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. 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 21-6. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

21.4 Interrupt Servicing Operations

21.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 20-4 below.

For the interrupt request acknowledgment timing, see **Figures 21-8** and **21-9**

Table 21-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

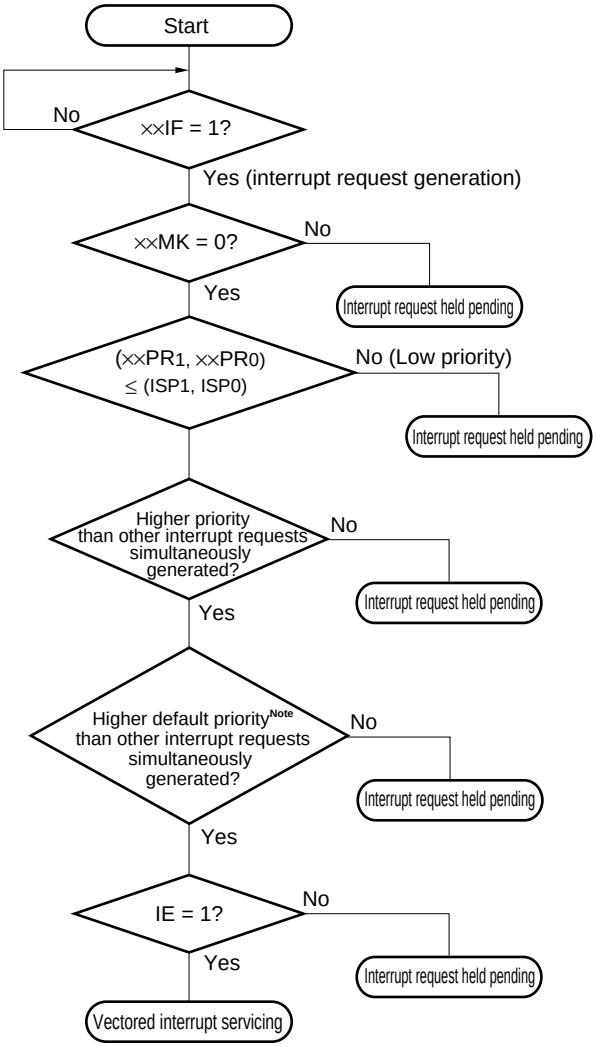
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 20-7shows 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.

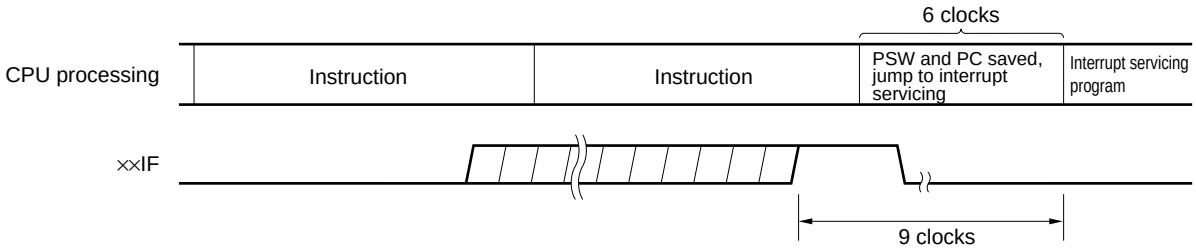
Figure 21-7. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 21-6**)

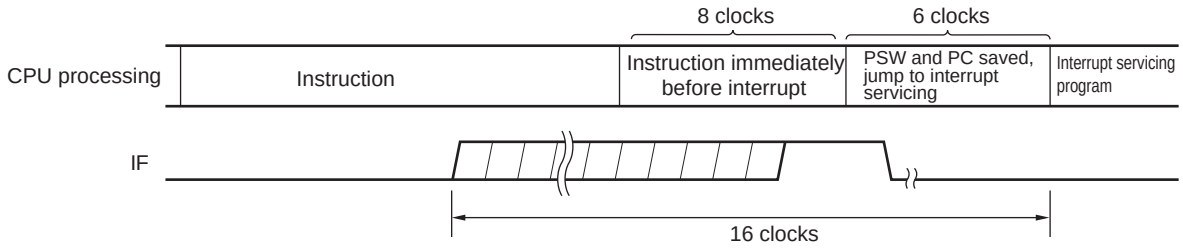
Note For the default priority, refer to **Table 21-1 Interrupt Source List**.

Figure 21-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 21-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

21.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

21.4.3 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 20-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 20-10 shows multiple interrupt servicing examples.

Table 21-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	√	-	-	-	-	-	-	-	√
	ISP1 = 0, ISP0 = 1	√	-	√	-	×	-	-	-	√
	ISP1 = 1, ISP0 = 0	√	-	√	-	√	-	-	-	√
Software interrupt		√	-	√	-	√	-	√	-	√

Remarks 1. √: Multiple interrupt servicing enabled

2. -: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

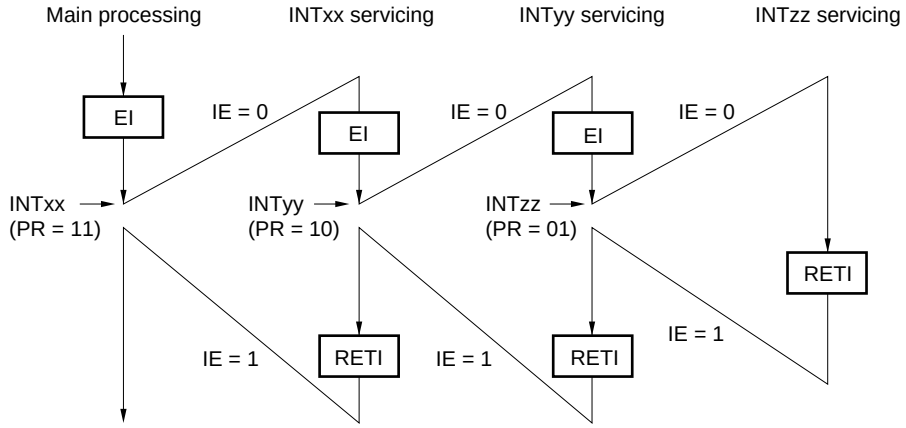
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

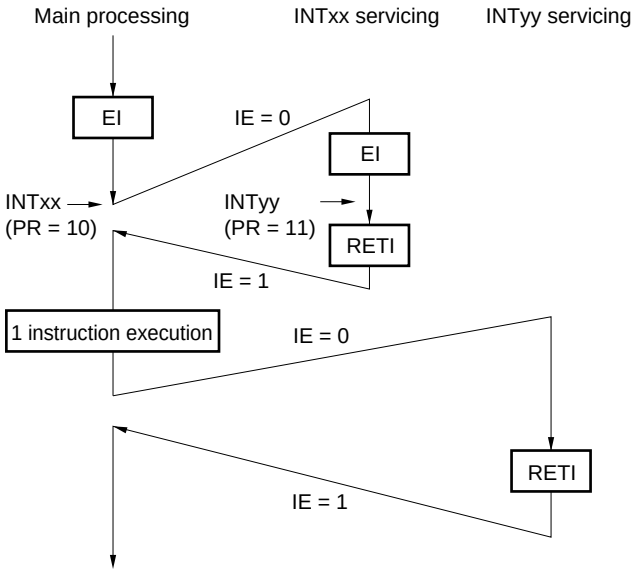
Figure 21-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

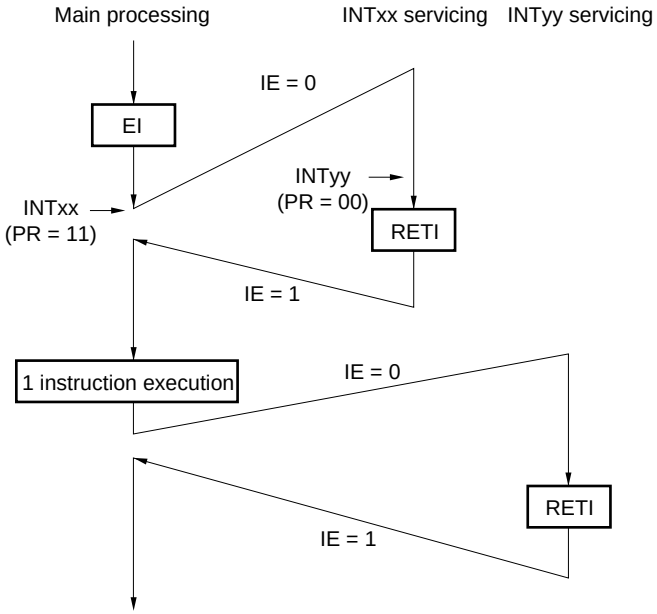


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 21-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

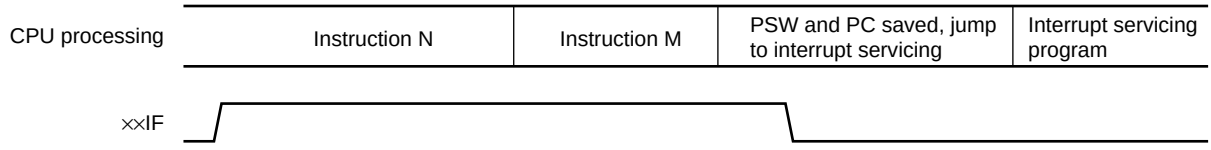
21.4.4 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

Figure 20-11 shows the timing at which interrupt requests are held pending.

Figure 21-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 22 STANDBY FUNCTION

22.1 Standby Function and Configuration

22.1.1 Standby function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, high-speed on-chip oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of an A/D conversion request by the hardware trigger signal from external pin (ADTRG), the STOP mode is exited, A/D conversion is performed without operating the CPU. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 11.3 Registers Used in A/D Converter.

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 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 low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 28 OPTION BYTE.

22.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- <R> STOP status output control register (STPSTC) **Note**

Note 128-pin products only.

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

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

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.
The X1 clock oscillation stabilization time can be checked in the following case.

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

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

When reset is released (reset by $\overline{\text{RESET}}$ input, POR, LVD, WDT, and executing an illegal instruction), the STOP instruction and MSTOP bit (bit 7 of clock operation status control register (CSC)) = 1 clear this register to 00H.

Figure 22-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

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

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

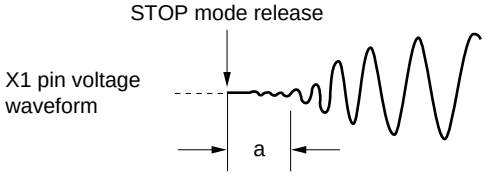
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS). If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC register oscillation stabilization time ≤ Oscillation stabilization time set by OSTS register**

Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register 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 f_X: X1 clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

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

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

When the high-speed on-chip oscillator clock is selected as the CPU clock, confirm with the oscillation stabilization time counter status register (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 the OSTC register.

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

Reset signal generation sets this register to 07H.

Figure 22-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS.2	OSTS.1	OSTS.0

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

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

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

3. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

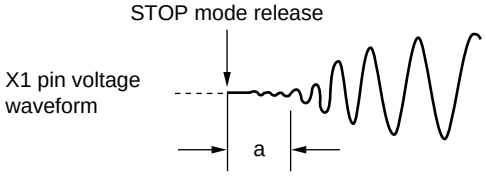
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register. If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC register oscillation stabilization time ≤ Oscillation stabilization time set by OSTS register**

Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after STOP mode is released.

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



Remark f_X: X1 clock oscillation frequency

<R> **(3) STOP status output control register (STPSTC) (128-pin products only)**

This register controls the output of STOP status.

Once STOP release triggers occurs, or SNOOZE released to normal mode occurs, P41 pin level is inverted. This function is incorporated to only 128-pin products.

When a STOP released or SNOOZE released to normal mode, this register can enable to output the inverted signal STOPST from P41.

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

Reset signal generation sets this register to 00H.

Caution When using STOPST output, P41 should be set to output mode and the port latch of P41 should be set to "0" in advance.

Figure 22-3. Format of STOP status output control register (STPSTC)

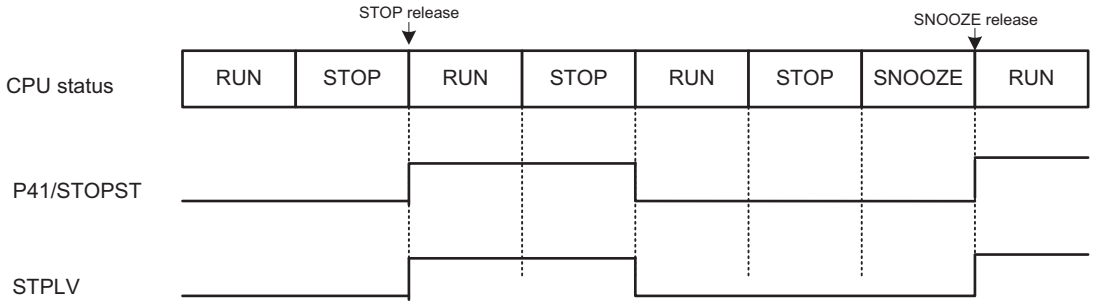
Address: F0016H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
STPSTC	STPOEN	0	0	STPLV	0	0	0	0

STPOEN	Operation when STOP released
0	P41/STOPST performs no operation when STOP released or SNOOZE released to normal mode
1	STPLV can be output as STOPST from P41 when STOP released or SNOOZE released to normal mode.
The STPOEN bit controls output of the STPLV from P41.	

STPLV	Data control when STOP released or SNOOZE released to normal mode
0	Low output (will invert to high at next STOP released or SNOOZE released to normal mode)
1	High output (will invert to low at next STOP released or SNOOZE released to normal mode)
The STPLV bit is inverted when STOP released or SNOOZE released to normal mode, regardless of the STPOEN status. See Figure 22-4	

Figure 22-4. Timings of STPLV, P41/STOPST



22.2 Standby Function Operation

22.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 22-1. Operating Statuses in HALT Mode (1/3)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-speed on-chip oscillator Clock (f_{IH}) or f_{IH}+PLL	When CPU Is Operating on X1 Clock (f_X) or f_X +PLL	When CPU Is Operating on External Main System Clock (f_{EX}) or f_{EX} +PLL
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops		
PLL			Status before HALT mode was set is retained		
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM			Operation stopped (however, operable when DMA is executed)		
CREG			Status before HALT mode was set is retained		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
Real-time clock (RTC)					
Interval timer					
Watchdog timer			See CHAPTER 10 WATCHDOG TIMER		
CLM			Operable if f_{IL} is not stopped		
PCL			Operable		
A/D converter					
<R> SAU (CSI, I^2C, UART)					
Serial interface LIN-UART (UARTF)					
CAN controller					
LCD controller/driver					
<R> LCD Bus interface					
Sound generator					
Stepper motor controller/driver (with ZPD)					
Multiplier and divider/multiply-accumulator					
DMA controller					
Power-on-reset function					
Voltage detection function					
External interrupt			Acceptable		
Internal interrupt					
CRC operation function	High-speed CRC		Operable		
	General-purpose CRC				

Table 22-1. Operating Statuses in HALT Mode (2/3)

HALT Mode Setting / Item	When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
	When CPU Is Operating on High-speed on-chip oscillator Clock (f_{IH}) or f_{IH}+PLL	When CPU Is Operating on X1 Clock (f_X) or f_X +PLL	When CPU Is Operating on External Main System Clock (f_{EX}) or f_{EX} +PLL
Illegal access detection function	Operation stopped (however, it is possible when DMA is executed)		
RAM parity check function			
RAM guard function			
SFR guard function			
BCD	Operation stopped		

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation must be stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock f_{XT}: XT1 clock
f_X: X1 clock

Table 22-1. Operating Statuses in HALT Mode (3/3)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on XT1 Clock (f_{XT})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}	Operation disabled
		f_X	
		f_{EX}	
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops
PLL			Operation disabled
CPU			Operation stopped
Code flash memory			
Data flash memory			
RAM			Operation stopped (however, operable when DMA is executed)
CREG			Status before HALT mode was set is retained
Port (latch)			Status before HALT mode was set is retained (CPU is stopped, while input/output function is possible by DMA access during HALT mode)
Timer array unit			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))
Real-time clock (RTC)			Operable
Interval timer			
Watchdog timer			See CHAPTER 10 WATCHDOG TIMER
CLM			Operation stopped
PCL			Operable (Operation is disabled while in the low consumption RTC mode)
A/D converter			Operation disabled
<R> SAU (CSI, I^2C, UART)			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))
Serial interface LIN-UART (UARTF)			
CAN controller			
LCD controller/driver			
<R> LCD Bus interface			Operation disabled
Sound generator			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))
Stepper motor controller/driver (with ZPD)			
Multiplier and divider/multiply-accumulator			
DMA controller			
Power-on-reset function			Operable
Voltage detection function			
External interrupt			Acceptable
Internal interrupt			
CRC operation function	High-speed CRC		Operation disabled
	General-purpose CRC		Operable
Illegal access detection function			Operation stopped (however, it is possible when DMA is executed)
RAM parity check function			
RAM guard function			
SFR guard function			
BCD			Operation stopped

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation must be stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock f_{XT}: XT1 clock f_X: X1 clock

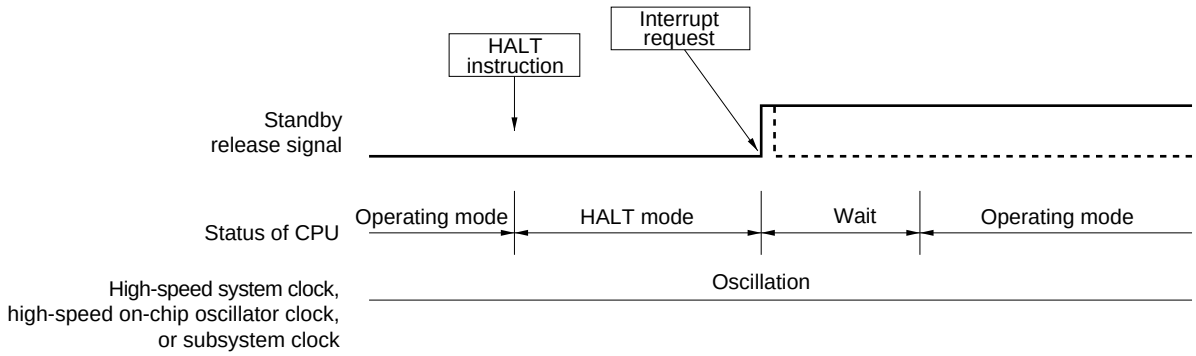
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 22-5. HALT Mode Release by Interrupt Request Generation



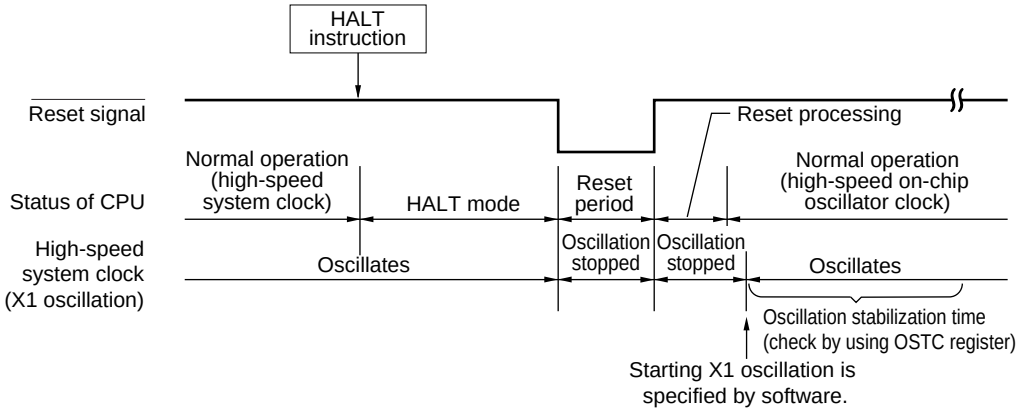
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

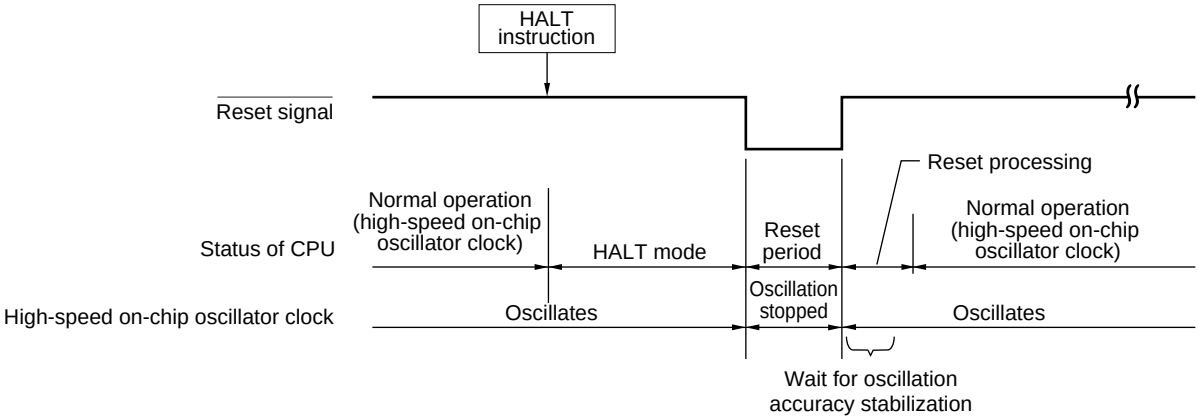
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 22-6 HALT Mode Release by Reset

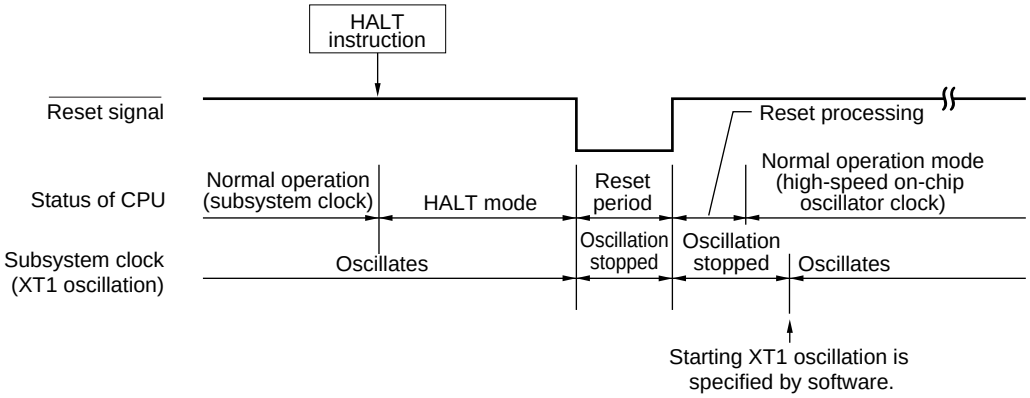
(1) When high-speed system clock is used as CPU clock



When high-speed on-chip oscillator clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

22.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the main system clock.

Cautions 1. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

2. When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 11.3 Registers Used in A/D Converter.

The operating statuses in the STOP mode are shown below.

Table 22-2. Operating Statuses in STOP Mode (1/2)

Item \ STOP Mode Setting	When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
	When CPU Is Operating on High-speed on-chip oscillator Clock (f_{IH}) or f_{IH}+PLL	When CPU Is Operating on X1 Clock (f_X) or f_X+PLL	When CPU Is Operating on External Main System Clock (f_{EX}) or f_{EX}+PLL
System clock	Clock supply to the CPU is stopped		
Main system clock (f_{IH})	Stopped		
Main system clock (f_X)			
Main system clock (f_{EX})			
Subsystem clock (f_{XT})	Status before STOP mode was set is retained		
f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops		
PLL	Operation disabled		
CPU	Operation stopped		
Code flash memory			
Data flash memory	Operation stopped (Executing the STOP instruction is disabled during data flash programming)		
RAM	Operation stopped		
CREG	Low power mode		
Port (latch)	Status before STOP mode was set is retained		
Timer array unit	Operation disabled		
Real-time clock (RTC)	Operable by f_{XT} or f_{IL}		
Interval timer			
Watchdog timer	See CHAPTER 10 WATCHDOG TIMER		
CLM	Operation stopped		
PCL	Operable only when f_{XT} clock is selected		
A/D converter	Wakeup operation is enabled (switching to the SNOOZE mode)		
<R> SAU (CSI, I^2C, UART)	Operation stopped		
Serial interface LIN-UART (UARTF)	Operation disabled (STOP release by INTPLRx is possible)		
CAN controller	Operation disabled (STOP release by INTCxWUP during CAN sleep mode is possible)		
LCD controller/driver	Operable by f_{XT} or f_{IL}		
<R> LCD bus interface	Operation stopped		
Sound generator			
Stepper motor controller/driver (with ZPD)			
Multiplier and divider/multiply-accumulator	Operation disabled		
DMA controller			
Power-on-reset function	Operable		
Voltage detection function			
External interrupt	Acceptable		
Internal interrupt	Interrupts from operable peripherals are acceptable		

Table 22-2. Operating Statuses in STOP Mode (2/2)

STOP Mode Setting / Item		When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
		When CPU Is Operating on High-speed on-chip oscillator Clock (f_{IH}) or f_{IH}+PLL	When CPU Is Operating on X1 Clock (f_X) or f_X+PLL	When CPU Is Operating on External Main System Clock (f_{EX}) or f_{EX}+PLL
CRC operation function	High-speed CRC	Operation stopped		
	General-purpose CRC			
Illegal access detection function				
RAM parity check function				
RAM guard function				
SFR guard function				
BCD				

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation must be stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock

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 low-speed on-chip oscillator clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.

3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction. Before changing the CPU clock from the high-speed on-chip oscillator 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

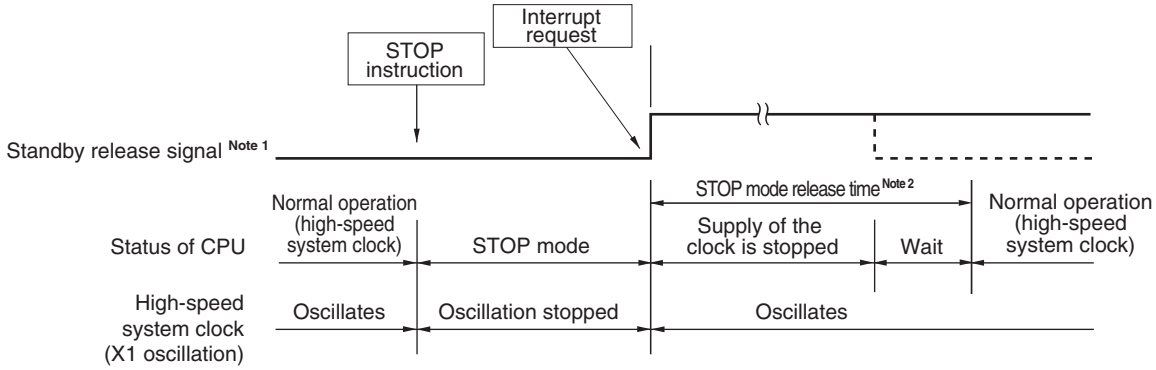
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 22-7. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 21-1**.

2. STOP mode release time

Supply of the clock is stopped: 18 μs to whichever is longer 65 μs and the oscillation stabilization time (set by OSTS) (additional wait cycles are required when using a PLL.)

Wait

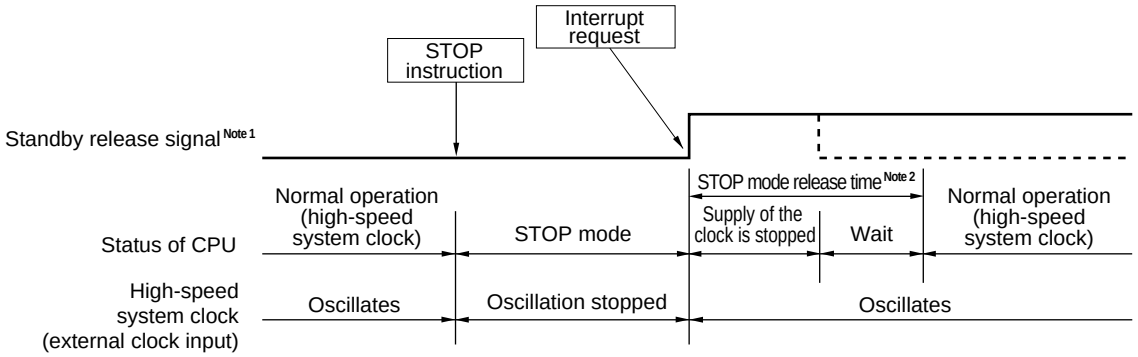
- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

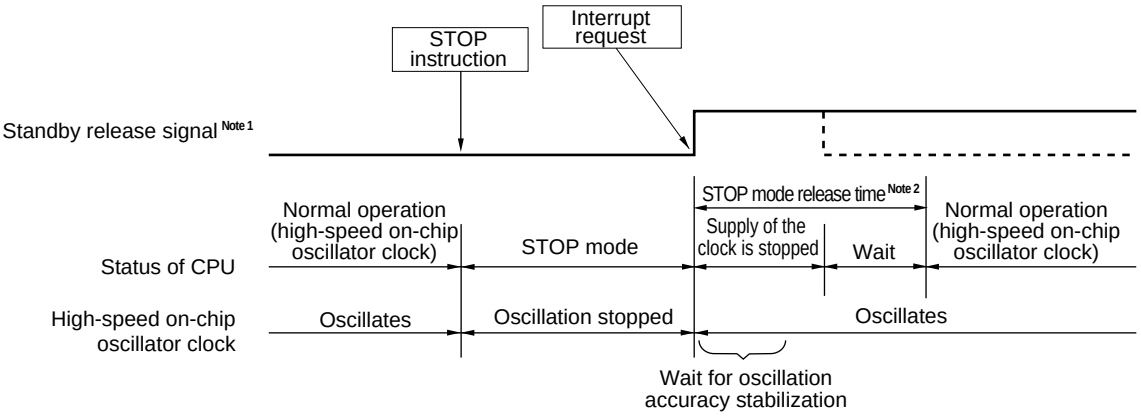
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 22-7. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (external clock input) is used as CPU clock



(3) When high-speed on-chip oscillator clock is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 21-1**.

2. STOP mode release time

Supply of the clock is stopped: 18 μs to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

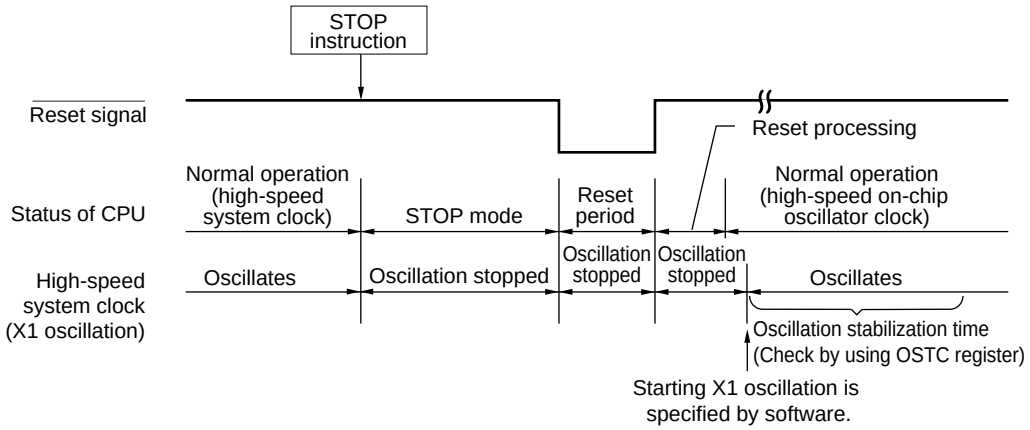
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

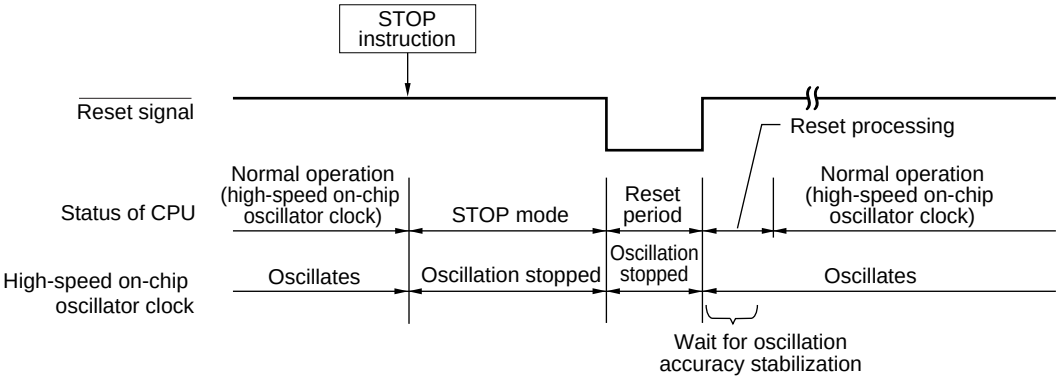
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 22-8. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When high-speed on-chip oscillator clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

22.2.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for the A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see **11.3 Registers Used in A/D Converter**.

The transition time of going into and getting out from SNOOZE mode is as following.

Transition time from STOP mode to SNOOZE mode

18 to 65 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation

- When vectored interrupt servicing is carried out:
  4.99 to 9.44 μs + 7 clocks
- When vectored interrupt servicing is not carried out:
  4.99 to 9.44 μs + 1 clock

The operating statuses in the SNOOZE mode are shown below.

Table 22-3. Operating Statuses in SNOOZE Mode (1/2)

Item \ SNOOZE Mode Setting		When Inputting A/D Converter Timer Trigger Signal While in SNOOZE Mode
		When CPU Is Operating on High-speed on-chip oscillator Clock (f_{IH})
System clock		Clock supply to the CPU is stopped
Main system clock	f_{IH}	Operation started
	f_X	Stopped
	f_{EX}	
Subsystem clock	f_{XT}	Use of the status while in the STOP mode continues
f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops
PLL		Operation stopped
CPU		
Code flash memory		
Data flash memory		
RAM		
CREG		Low power mode
Port (latch)		Use of the status while in the STOP mode continues
Timer array unit		Operation disabled
Real-time clock (RTC)		Operable
Interval timer		
Watchdog timer		See CHAPTER 10 WATCHDOG TIMER
CLM		Operation stopped
A/D converter		Operable
<R> SAU (CSI, I^2C, UART)		Operation disabled
Serial interface LIN-UART (UARTF)		
CAN controller		
LCD controller/driver		
<R> LCD bus interface		
Sound generator		
Stepper motor controller/driver (with ZPD)		
Multiplier and divider/multiply-accumulator		
DMA controller		
Power-on-reset function		Operable
Voltage detection function		
External interrupt		
Internal interrupt		

Table 22-3. Operating Statuses in SNOOZE Mode (2/2)

SNOOZE Mode Setting / Item		When Inputting A/D Converter Timer Trigger Signal While in SNOOZE Mode
		When CPU Is Operating on High-speed on-chip oscillator Clock (f_{IH})
CRC operation function	High-speed CRC	Operation disabled
	General-purpose CRC	
Illegal access detection function		
RAM parity check function		
RAM guard function		
SFR guard function		
BCD		

Remark Operation stopped: Operation is automatically stopped before switching to the SNOOZE mode.
Operation disabled: Operation must be stopped before switching to the SNOOZE mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock

CHAPTER 23 RESET FUNCTION

The following eight operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction[Note]
(6) Internal reset by RAM parity error
(7) Internal reset by detection of main clock oscillation stop via clock monitoring
(8) Internal reset by 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, or by POR and LVD circuit voltage detection, execution of illegal instruction[Note], RAM parity error, detection of main clock oscillation stop via clock monitoring, or illegal-memory access, and each item of hardware is set to the status shown in Tables 22-1.

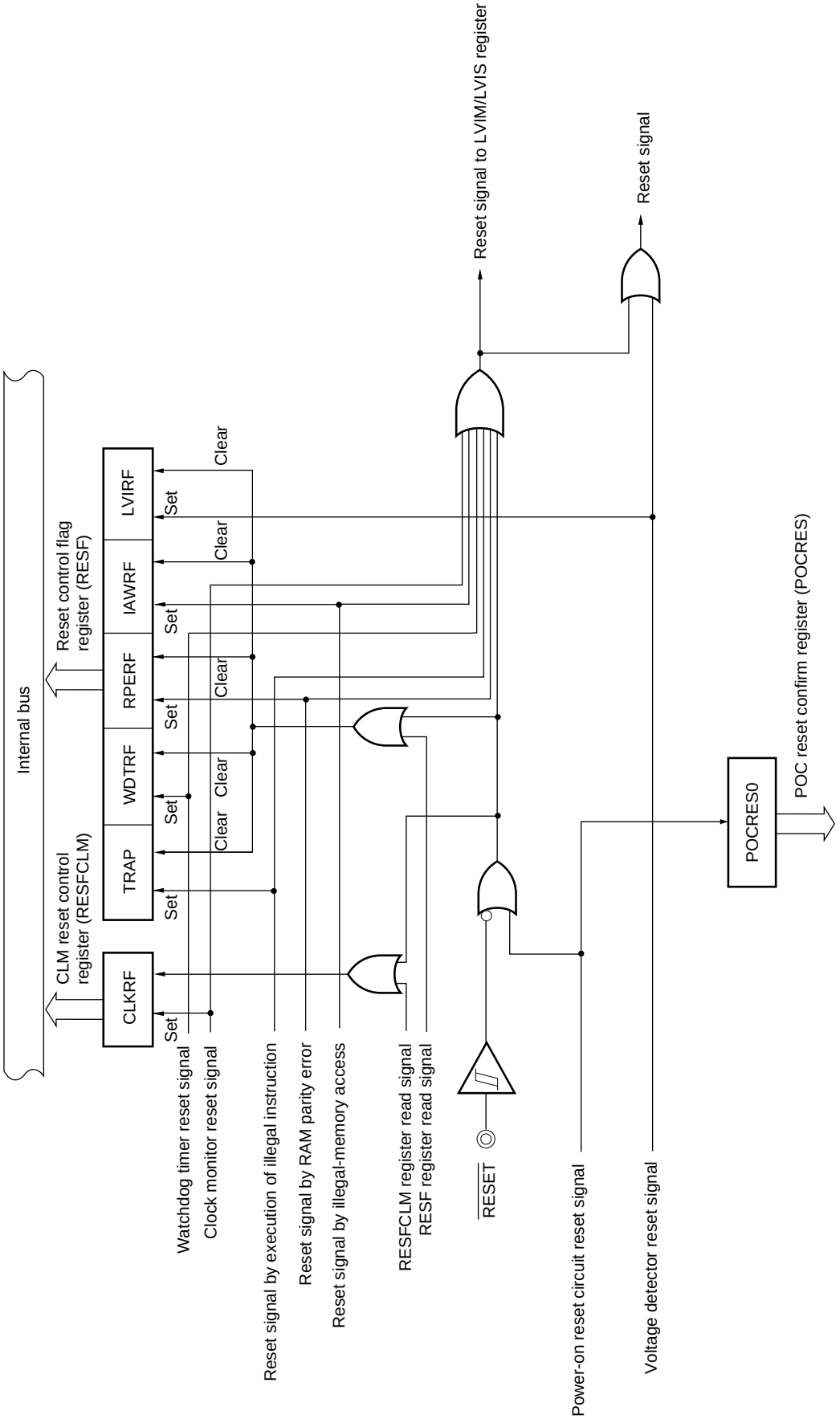
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the high-speed on-chip oscillator clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the high-speed on-chip oscillator clock (see **Figures 23-2** to **23-4**) after reset processing. Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset, and program execution starts using the high-speed on-chip oscillator clock (see **CHAPTER 24 POWER-ON-RESET CIRCUIT** and **CHAPTER 25 VOLTAGE DETECTOR**) after reset processing.

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

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
(To perform an external reset upon power application, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range ($V_{DD} \geq 2.7$ V).)
2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input becomes invalid.
3. When reset is effected, port pin P130 is set to low-level output and other port pins become high-impedance, because each SFR and 2nd SFR are initialized.

Remark V_{POR}: POR power supply rise detection voltage

Figure 23-1. Block Diagram of Reset Function



Caution An LVD circuit internal reset does not reset the LVD circuit.

Remarks 1. LVIM: Voltage detection register

2. LVIS: Voltage detection level register

Figure 23-2. Timing of Reset by $\overline{\text{RESET}}$ Input

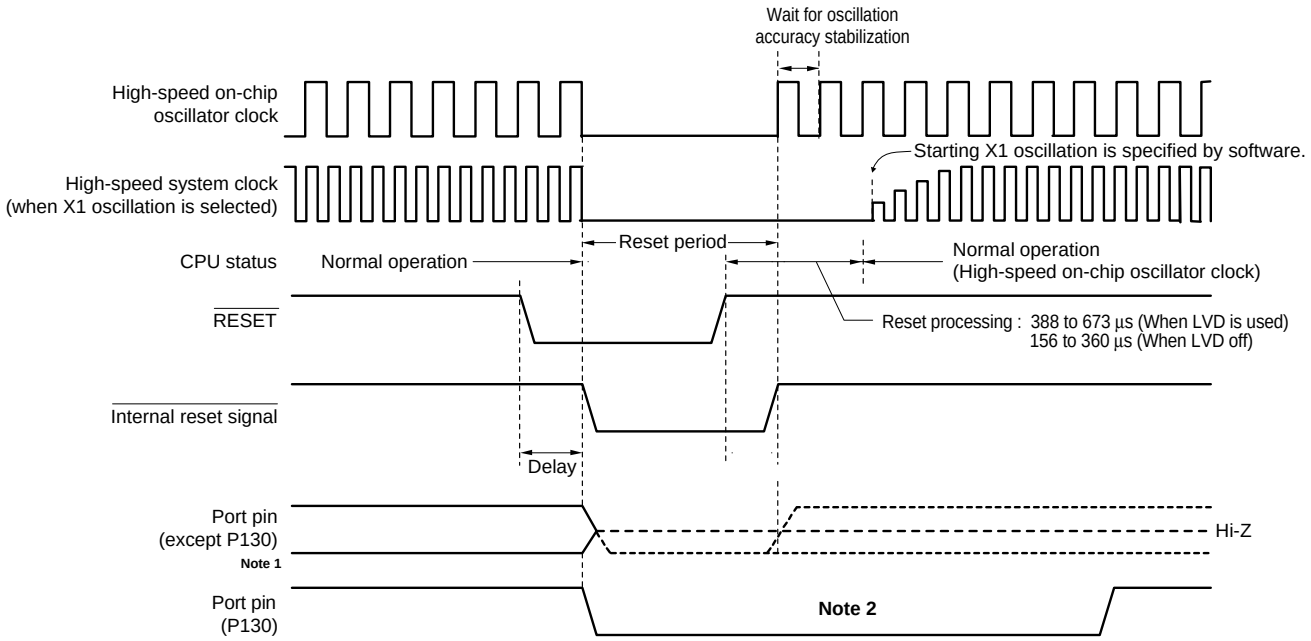
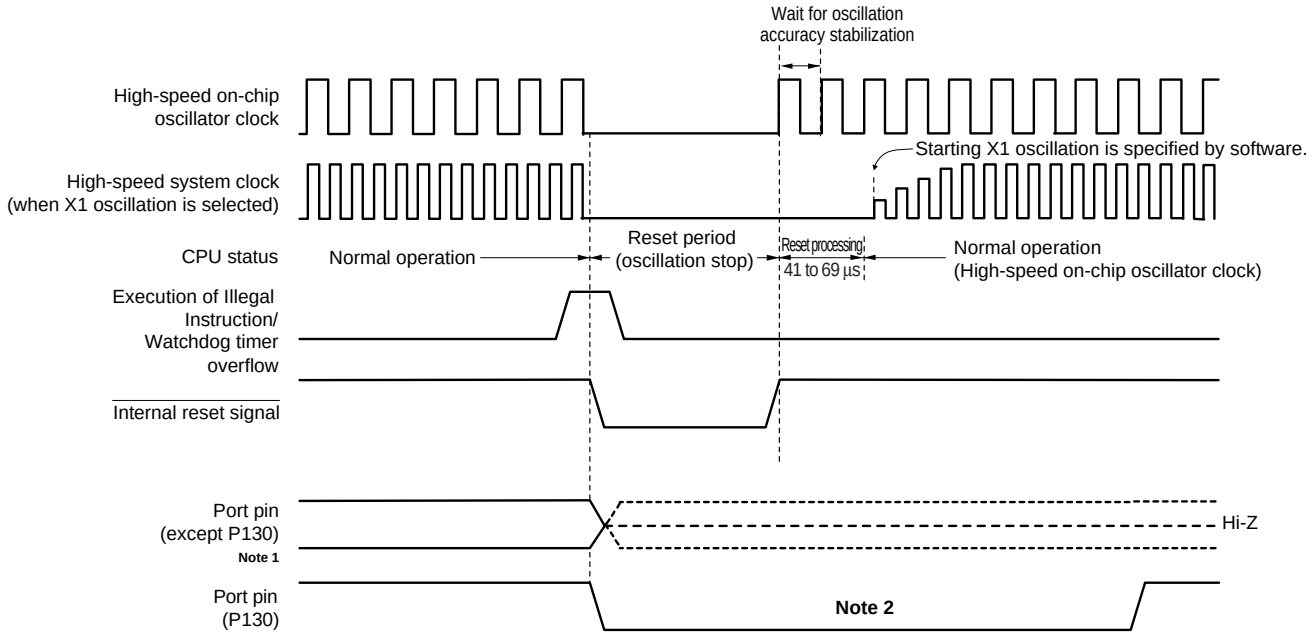


Figure 23-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow

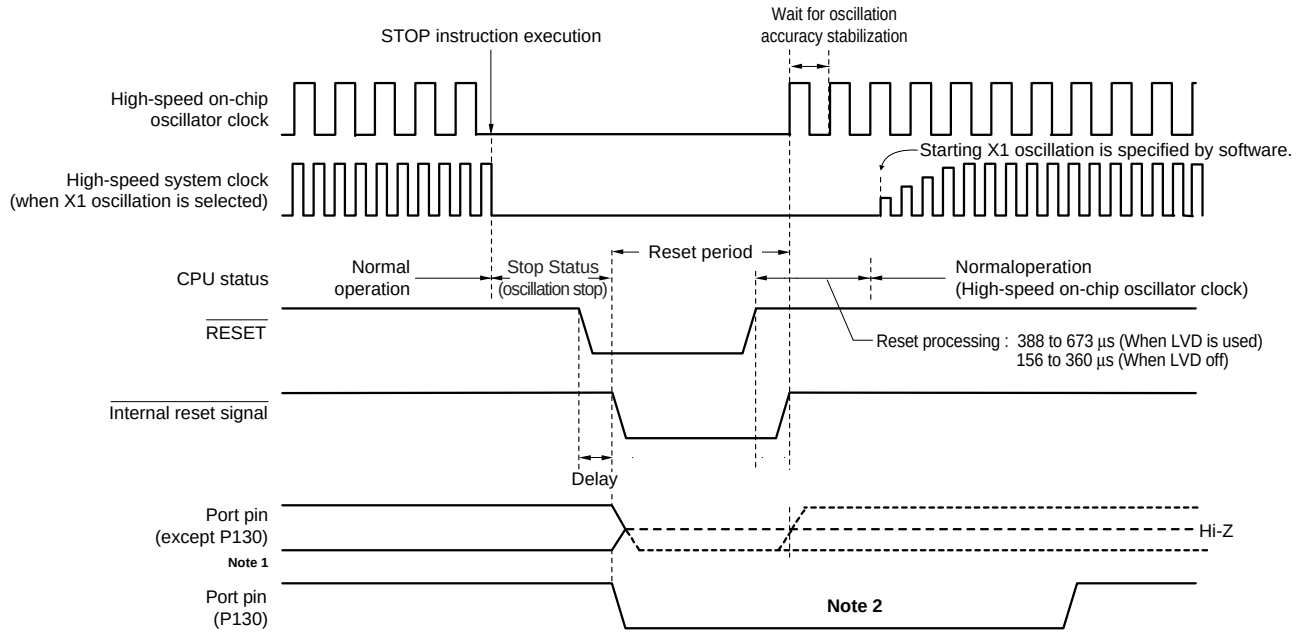


Notes 1. Segment output pin is pull-downed while POR reset or $\overline{\text{RESET}}$ input is stay active.

2. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

Caution A watchdog timer internal reset resets the watchdog timer.

Figure 23-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Notes 1. Segment output pin is pull-downed while POR reset or $\overline{\text{RESET}}$ input is stay active.

2. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

Remark For the reset timing of the power-on-reset circuit and voltage detector, see **CHAPTER 24 POWER-ON-RESET CIRCUIT** and **CHAPTER 25 VOLTAGE DETECTOR**.

Table 23-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped
		f_{EX}	Operation stopped
	Subsystem clock	f_{XT}	Operation stopped
	f_{IL}		Operation stopped
PLL			
CPU			
Code flash memory			Operation stopped (The system operates in the LV (low voltage main) mode after reading the option byte)
Data flash memory			Operation stopped
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-on-reset detection voltage.)
Port (latch)			See **CHAPTER 2 PIN FUNCTIONS**
Timer array unit			Operation stopped
Real-time clock (RTC)			
Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Serial interface LIN-UART (UARTF)			
CAN controller			
LCD controller/driver			
<R> LCD Bus I/F			
Sound generator			
Stepper motor controller/driver (with ZPD)			
Multiplier & divider, multiply-accumulator			
DMA controller			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation stopped (LVD detection is possible after reading the option byte)
External interrupt			Operation stopped
CRC operation function	High-speed CRC		
	General-purpose CRC		
Illegal access detection function			
RAM parity check function			
RAM guard function			
SFR guard function			

Remark f_{IH}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{IL}: Low-speed on-chip oscillator clock

Table 23-2. Hardware Statuses After Reset Acknowledgment (1/7)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P9, P13 to P15) (output latches)		00H
Port mode registers 0 to 9, 13 to 15 (PM0 to PM9, PM13 to PM15)		FFH[Note3]
Port input mode registers 0, 1, 3, 5 to 7, 13 (PIM0, PIM1, PIM3, PIM5 to PIM7, PIM13)		00H
Port output mode registers (POM)		00H
Pull-up resistor option registers 0, 1, 3 to 9, 13, 14 (PU0, PU1, PU3 to PU9, PU13, PU14)		00H (PU4 is 01H)
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
<R> STOP status output control register (STPSTC)		00H
System clock control register (CKC)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Peripheral enable registers 0, 1 (PER0, PER1)		00H
<R> High-speed on-chip oscillator trimming register (HIOTRM)		Undefined
PLL control register (PLLCTL)		00H
PLL status register (PLLSTS)		00H
Peripheral clock select register (PCKSEL)		00H
FMP clock division selection register (MDIV)		00H
Timer input select registers 00, 01, 10, 11, 20, 21 (TIS00, TIS01, TIS10, TIS11, TIS20, TIS21)		00H
Timer array unit	Timer data registers 00 to 07, 10 to 17, 20 to 27 (TDR00 to TDR07, TDR10 to TDR17, TDR20 to TDR27)	0000H
	Timer mode registers 00 to 07, 10 to 17, 20 to 27 (TMR00 to TMR07, TMR10 to TMR17, TMR20 to TMR27)	0000H
	Timer status registers 00 to 07 (TSR00 to TSR07)	0000H
	Timer counter registers 00 to 07, 10 to 17, 20 to 27 (TCR00 to TCR07, TCR10 to TCR17, TCR20 to TCR27)	FFFFH
	Timer channel enable status register 0 (TE0)	0000H
	Timer channel start register 0 (TS0)	0000H
	Timer channel stop register 0 (TT0)	0000H
	Timer clock select register 0 to 2(TPS0 to TPS2)	0000H
	Timer output register 0 (TO0)	0000H
	Timer output enable register 0 (TOE0)	0000H
	Timer output level register 0 (TOL0)	0000H
	Timer output mode registers 0 to 2 (TOM0 to TOM2)	0000H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Value afte reset is FEH only for PM3.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (2/7)

Hardware		Status After Reset Acknowledgment[Note 1]
Timer input select else register (TISELSE)		00H
Sound generator pin select register (SGSEL)		00H
Timer output select register 00, 01, 10, 11, 20, 21 (TOS00, TOS01, TOS10, TOS11, TOS20, TOS21)		00H
Timer array unit	Noise filter enable register for each channel of TAU unit0 to 2 BCD correction result register (TNFEN0BCDAJ to TNFEN2)	00H
	Sampling clock select of noise filter for unit0 to 2 (2 set) (TNFSMP0, TNFSMP1, TNFSMP2)	00H
	Noise filter clock select register for each channel of TAU unit0 to 2 (TNFCS0, TNFCS1, TNFCS2)	00H
Real-time clock	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD, SUBCUDW)	00H, 0000H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register ALARMWW)	00H
	Control register 0 (RTCC0)	00H
	Control register 1 (RTCC1)	00H
	RTC clock selection register (RTCCL)	00H
	RTC1Hz pin select register (RTCSEL)	00H
Interval timer	Interval timer control register (ITMC)	7FFFH
Clock output/buzzer output controller	Clock output select register 0 (CKS0)	00H
Watchdog timer	Watchdog timer enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode registers 0 to 2 (ADM0 to ADM2)	00H
	Conversion result comparison upper limit setting register (ADUL)	FFH
	Conversion result comparison lower limit setting register (ADLL)	00H
	A/D test register (ADTES)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

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

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (3/7)

Hardware		Status After Reset Acknowledgment[Note 1]
Serial array unit (SAU)	Serial data registers 00, 01, 10, 11 (SDR00, SDR01, SDR10, SDR11)	0000H
	Serial status registers 00, 01, 10, 11 (SSR00, SSR01, SSR10, SSR11)	0000H
	Serial flag clear trigger registers 00, 01, 10, 11 (SIR00, SIR01, SIR10, SIR11)	0000H
	Serial mode registers 00, 01, 10, 11 (SMR00, SMR01, SMR10, SMR11)	0020H
	Serial communication operation setting registers 00, 01, 10, 11 (SCR00, SCR01, SCR10, SCR11)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop registers 0, 1 (ST0, ST1)	0000H
	Serial clock select registers 0, 1 (SPS0, SPS1)	0000H
	Serial output registers 0, 1 (SO0, SO1)	0303H
	Serial output enable registers 0, 1 (SOE0, SOE1)	0000H
	Serial output level registers 0, 1 (SOL0, SOL1)	0000H
	Serial communication pin select register 0, 1(STSEL0, STSEL1)	00H
DMA controller	DMA SFR address registers 0 to 3 (DSA0 to DSA3)	00H
	DMA RAM address registers 0 to 3 (DRA0 to DRA3)	00H
	DMA byte count registers 0 to 3 (DBC0 to DBC3)	00H
	DMA mode control registers 0 to 3 (DMC0 to DMC3)	00H
	DMA operation control registers 0 to 3 (DRC0 to DRC3)	00H
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 register 0 (EGP0)	00H
	External interrupt falling edge enable register 0 (EGN0)	00H

Note 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.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (4/7)

Hardware		Status After Reset Acknowledgment[Note 1]
UART	LIN-UART0 control register 0, 1 (UF0CTL0, UF0CTL1)	10H, 0FFFH
	LIN-UART0 option control registers 0, to 2 (UF0OPT0 to UF0OPT2)	14H, 00H
	LIN-UART0 status register (UF0STR)	0000H
	LIN-UART0 status clear register (UF0STC)	0000H
	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 to 8 (UF0BUF0 to UF0BUF8)	00H
	LIN-UART0 buffer control register (UF0BUCTL)	0000H
	LIN-UART0 transmit data register (UF0TX)	0000H
	LIN-UART0 8-bit transmit data register (UF0TXB)	00H
	LIN-UART0 receive data register (UF0RX)	0000H
	LIN-UART0 receive data register (UF0RXB)	00H
	LIN-UART1 control register 0 to 1 (UF1CTL0, UF1CTL1)	10H, 0FFFH
	LIN-UART1 option control registers 0 to 2 (UF1OPT0 to UF1OPT2)	14H, 00H
	LIN-UART1 status register (UF1STR)	0000H
	LIN-UART1 status clear 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 to 8 (UF1BUF0 to UF1BUF8)	00H
	LIN-UART1 buffer control register (UF1BUCTL)	0000H
	LIN-UART1 transmit data register (UF1TX)	0000H
	LIN-UART1 8-bit transmit data register (UF1TXB)	00H
	LIN-UART1 receive data register (UF1RX)	0000H
	LIN-UART1 receive data register (UF1RXB)	00H

Note 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.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (5/7)

Hardware		Status After Reset Acknowledgment[Note 1]
CAN controller	CAN0 global module control register (C0GMCTRL)	0000H
	CAN0 global block transmission control register (C0GMABT)	0000H
	CAN0 global block transmission delay setting register (C0GMABTD)	00H
	CAN0 global module clock select register (C0GMCS)	0FH
	CAN0 module mask 1 to 4 register L (C0MASK1L to C0MASK4L)	Undefined
	CAN0 module mask 1 to 4 register H (C0MASK1H to C0MASK4H)	Undefined
	CAN0 module control register (C0CTRL)	0000H
	CAN0 module last error information register (C0LEC)	00H
	CAN0 module information register (C0INFO)	00H
	CAN0 module error counter register (C0ERC)	0000H
	CAN0 module interrupt enable register (C0IE)	0000H
	CAN0 module interrupt status register (C0INTS)	0000H
	CAN0 module bit rate prescaler register (C0BRP)	FFH
	CAN0 module bit rate register (C0BTR)	370FH
	CAN0 module last in-pointer register (C0LIPT)	Undefined
	CAN0 module receive history list register (C0RGPT)	xx02H
	CAN0 module last out-pointer register (C0LOPT)	Undefined
	CAN0 module transmit history list register (C0TGPT)	xx02H
	CAN0 module time stamp register (C0TS)	0000H
	CAN0 message data byte 01 register 00 to 15 (C0MDB0100 to C0MDB0115)	Undefined
	CAN0 message data byte 23 register 00 to 15 (C0MDB2300 to C0MDB2315)	Undefined
	CAN0 message data Byte 45 register 00 to 15 (C0MDB4500 to C0MDB4515)	Undefined
	CAN0 message data byte 67 register 00 to 15 (C0MDB6700 to C0MDB6715)	Undefined
	CAN0 message data length register 00 to 15 (C0MDLC00 to C0MDLC15)	0xH
	CAN0 message configuration register 00 to 15 (C0MCONF00 to C0MCONF15)	Undefined
	CAN0 message ID register 00L to 15L (C0MIDL00 to C0MIDL15)	Undefined
	CAN0 message ID register 00H to 15H (C0MIDH00 to C0MIDH15)	Undefined
	CAN0 message control register 00 to 15 (C0MCTRL00 to C0MCTRL15)	Undefined
	CAN1 global module control register (C1GMCTRL)	0000H
	CAN1 global module clock select register (C1GMCS)	0FH
	CAN1 global block transmission control register (C1GMABT)	0000H
	CAN1 global block transmission delay setting register (C1GMABTD)	00H
	CAN1 module mask 1 to 4 register L (C1MASK1L to C1MASK4L)	Undefined
	CAN1 module mask 1 to 4 register H (C1MASK1H to C1MASK4H)	Undefined
	CAN1 module control register (C1CTRL)	0000H
	CAN1 module last error information register (C1LEC)	00H
	CAN1 module information register (C1INFO)	00H
	CAN1 module error counter register (C1ERC)	0000H
	CAN1 module interrupt enable register (C1IE)	0000H
	CAN1 module interrupt status register (C1INTS)	0000H

Note 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.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (6/7)

Hardware		Status After Reset Acknowledgment[Note 1]
CAN controller	CAN1 module bit rate prescaler register (C1BRP)	FFH
	CAN1 module bit rate register (C1BTR)	370FH
	CAN1 module last in-pointer register (C1LIPT)	Undefined
	CAN1 module receive history list register (C1RGPT)	xx02H
	CAN1 module last out-pointer register (C1LOPT)	Undefined
	CAN1 module transmit history list register (C1TGPT)	xx02H
	CAN1 module time stamp register (C1TS)	0000H
	CAN1 message data byte 01 register 00 to 15 (C1MDB0100 to C1MDB0115)	Undefined
	CAN1 message data byte 23 register 00 to 15 (C1MDB2300 to C1MDB2315)	Undefined
	CAN1 message data byte 45 register 00 to 15 (C1MDB4500 to C1MDB4515)	Undefined
	CAN1 message data byte 67 register 00 to 15 (C1MDB6700 to C1MDB6715)	Undefined
	CAN1 message data length register 00 to 15 (C1MDLC00 to C1MDLC15)	0xH
	CAN1 message Configuration register 00 to 15 (C1MCONF00 to C1MCONF15)	Undefined
	CAN1 message ID register 00L to 15L (C1MIDL00 to C1MIDL15)	Undefined
	CAN1 message ID register 00H to 15H (C1MIDH00 to C1MIDH15)	Undefined
	CAN1 message control register 00 to 15 (C1MCTRL00 to C1MCTRL15)	Undefined
Stepper motor controller/driver	Timer mode control register 0 (MCNTC0)	00H
	Combined compare registers 1HW to 4HW (MCMP1HW to MCMP4HW)	0000H
	Compare registers for sine side (MCMP10, MCMP20, MCMP30, MCMP40)	00H
	Compare registers for cosine side (MCMP11, MCMP21, MCMP31, MCMP41)	00H
	Compare control registers 1 to 4 (MCMPC1 to MCMPC4)	00H
	Stepper motor port control register (SMPC)	00H
	ZPD detection voltage setting registers 0, 1 (ZPDS0, ZPDS1)	00H
	ZPD flag detection clock setting register (CMPCTL)	00H
	ZPD operation control register (ZPDEN)	00H
LCD controller	LCD display data memory 0 to 52 (SEG0 to SEG52)	00H
	LCD port function registers 0, 1, 3, 5, 7 to 9, 13 (LCDPF0, LCDPF1, LCDPF3, LCDPF5, LCDPF7 to LCDPF9, LCDPF13)	00H
	LCD mode register (LCDMD)	00H
	LCD display mode register (LCDM)	00H
	LCD clock control register (LCDC0)	00H
Sound generator	Control register (SG0CTL)	0000H
	Frequency register SG0FL (SG0FL)	0000H
	Frequency register SG0FH (SG0FH)	0000H
	Amplitude register (SG0PWM)	0000H
	Duration factor register (SG0SDF)	00H
	Interrupt threshold register (SG0ITH)	0000H

Note 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.

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 23-2. Hardware Statuses After Reset Acknowledgment (7/7)

Hardware		Status After Reset Acknowledgment[Note 1]
Multiplier & divider, multiply-accumulator	Multiplication/division data register A (MDAL, MDAH)	0000H
	Multiplication/division data register B (MDBL, MDBH)	0000H
	Multiplication/division data register C (MDCL, MDCH)	0000H
	Multiplication/division control register (MDUC)	00H
Reset function	Reset control flag register (RESF)	Undefined[Note 2]
	CLM reset control flag register (RESFCLM)	00H[Note 2]
	POR reset confirm register (POCRES)	00H
Voltage detector	Voltage detection register (LVIM)	00H[Note 2]
	Voltage detection level register (LVIS)	00H/01H/81H[Notes 2, 3]
Safety functions	Flash memory CRC control register (CRC0CTL)	00H
	Flash memory CRC operation result register (PGCRCL)	0000H
	CRC input register (CRCIN)	00H
	CRC data register (CRCD)	0000H
	Invalid memory access detection control register (IAWCTL)	00H
	RAM parity error control register (RPECTL)	00H
	Specific register manipulation protection register (GUARD)	00H
Flash memory	Data flash control register (DFLCTL)	00H
BCD correction circuit	BCD correction result register (BCDADJ)	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 reset source.

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

3. The generation of reset signal other than an LVD reset sets as follows.

- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

23.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the RL78/D1A. The reset control flag register (RESF) is used to store which source has generated the reset request.

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

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 23-5. 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	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source.

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

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. An instruction code fetched from RAM is not subject to parity error detection while it is being executed. However, the data read by the instruction is subject to parity error detection.

3. Because the RL78's CPU executes lookahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, when enabling RAM parity error resets (RPERDIS = 1), be sure to initialize the used RAM area + 10 bytes.

The status of the RESF register when a reset request is generated is shown in Table 23-3.

Table 23-3. RESF and RESFCLM Register Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD	Reset by clock monitor
TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held	Held
WDTRF bit			Held	Set (1)	Held	Held	Held	Held
RPERF bit			Held	Held	Set (1)	Held	Held	Held
IAWRF bit			Held	Held	Held	Set (1)	Held	Held
LVIRF bit			Held	Held	Held	Held	Set (1)	Held
CLKRF bit			Held	Held	Held	Held	Held	Set (1)

23.2 CLM Reset Control Flag Register

The CLM Reset Control Flag Register (RESFCLM) checks whether an internal reset generates. The CLKRF bit is cleared when the RESFCLM is cleared. The register can be accessed in 8-bit unit. The CLKRF bit is cleared by RESET input, power-on-reset (POR) circuit, and reading the register.

Figure 23-6. Format of CLM Reset Control Flag Register (RESFCLM)

Address: F00FAH After reset: 00H Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESFCLM	0	0	0	0	0	0	0	CLKRF

CLKRF	Internal reset request by clock monitor
0	Internal reset request is not generated, or the RESFCLM is cleared.
1	Internal reset request is generated.

Note 1. The value after reset varies depending on the reset source.

23.3 POR Reset Flag Register

The POC reset register (POCRES) checks the generation of POR reset, For POCRES, only writing ”1” is valid and writing ”0” is invalid, Only a reset by the Power-on-Reset (POR) circuit can clear this register to 00H. If use the flag, it is necessary to preset POCRES0 to ”1”.

Figure 23-7. Format of POR reset confirm register (POCRES)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POCRES	0	0	0	0	0	0	0	POCRES0

POCRES0	Internal reset request by POR reset
0	POR reset is generated or writing is not performed.
1	POR reset is not generated.

CHAPTER 24 POWER-ON-RESET CIRCUIT

24.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds 1.51 V ±0.06 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.50 V ±0.06 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

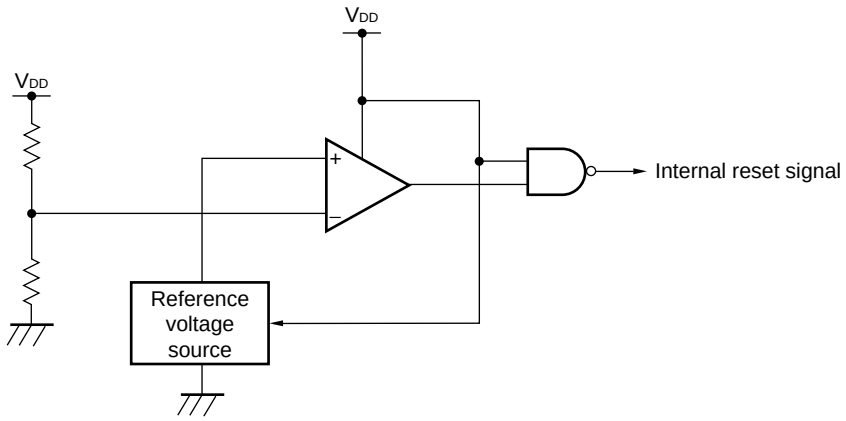
Caution If an internal reset signal is generated in the POR circuit, TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags of the reset control flag register (RESF) is cleared.

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), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 23 RESET FUNCTION**.

24.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 24-1.

Figure 24-1. Block Diagram of Power-on-reset Circuit



24.3 Operation of Power-on-reset Circuit

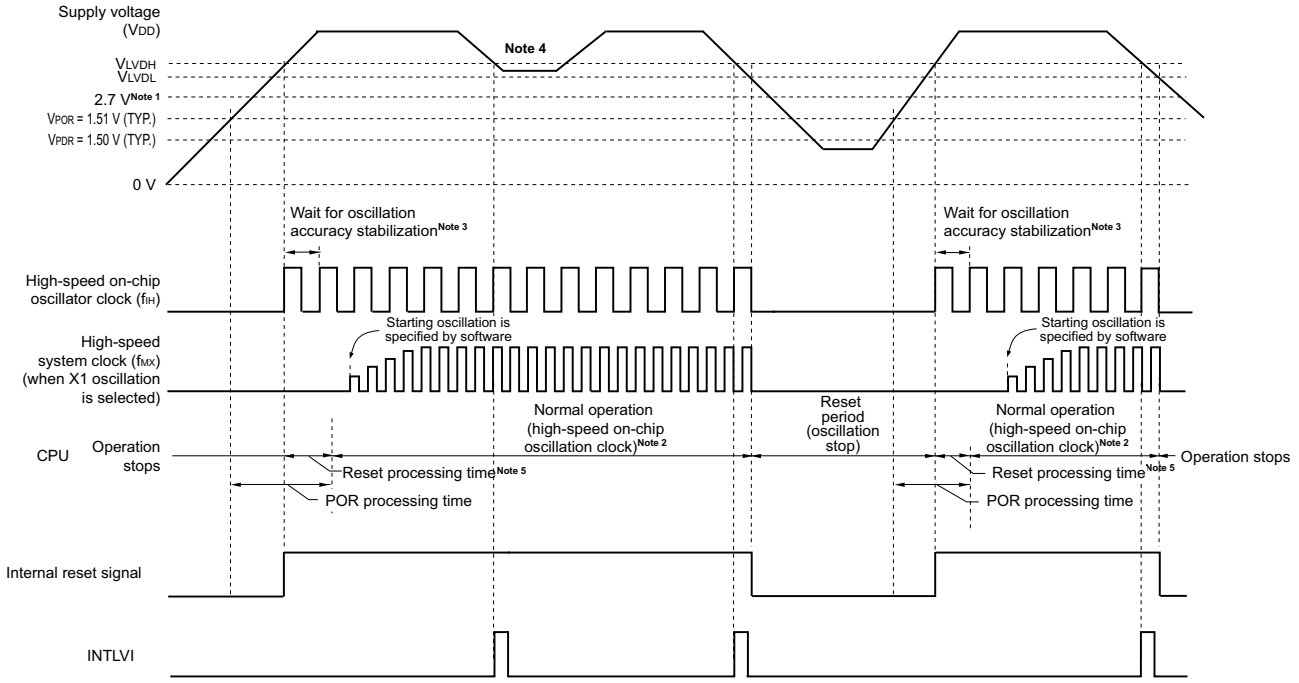
- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{PDR} = 1.51 V ±0.06 V), the reset status is released.

- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.50 V ±0.06 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-reset circuit and voltage detector is shown below.

Figure 24-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2)

(a) When LVD is interrupt & reset mode (V_{LVDL} = 2.75 V, V_{LVDH} = 2.92 V (option byte 000C1/020C1H = 7AH)



Notes 1. The operation guaranteed range is 2.7 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 2.7 V reset state when the supply voltage falls, use the reset function of the voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

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

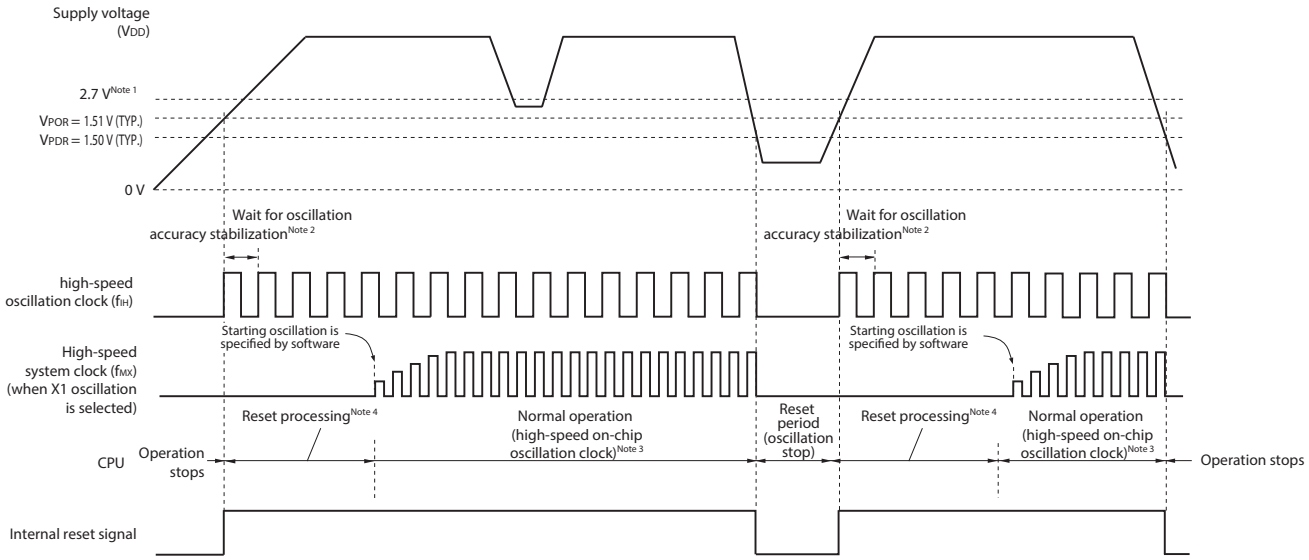
4. After the first interrupt request signal (INTLVI) is generated, the LVIL and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. If the operating voltage returns to 2.7 V or higher without falling below the voltage detection level (V_{LVDL}), after INTLVI is generated, perform the required backup processing, and then use software to specify the following settings in order (**see Figure 25-8. Initial Setting of Interrupt and Reset Mode**).

5. Reset processing time: 497 to 720 μ s

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 24-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)

(b) When LVD is OFF (option byte 000C1H/020C1H: VPOC2 = 1B)



Notes 1. The operation guaranteed range is 2.7 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 2.7 V reset state when the supply voltage falls, use the reset function of the voltage detector, or input the low level to the RESET pin.

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

3. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

4. For details about the reset processing time, see **Figure 5-18**.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

24.4 Cautions for Power-on-reset Circuit

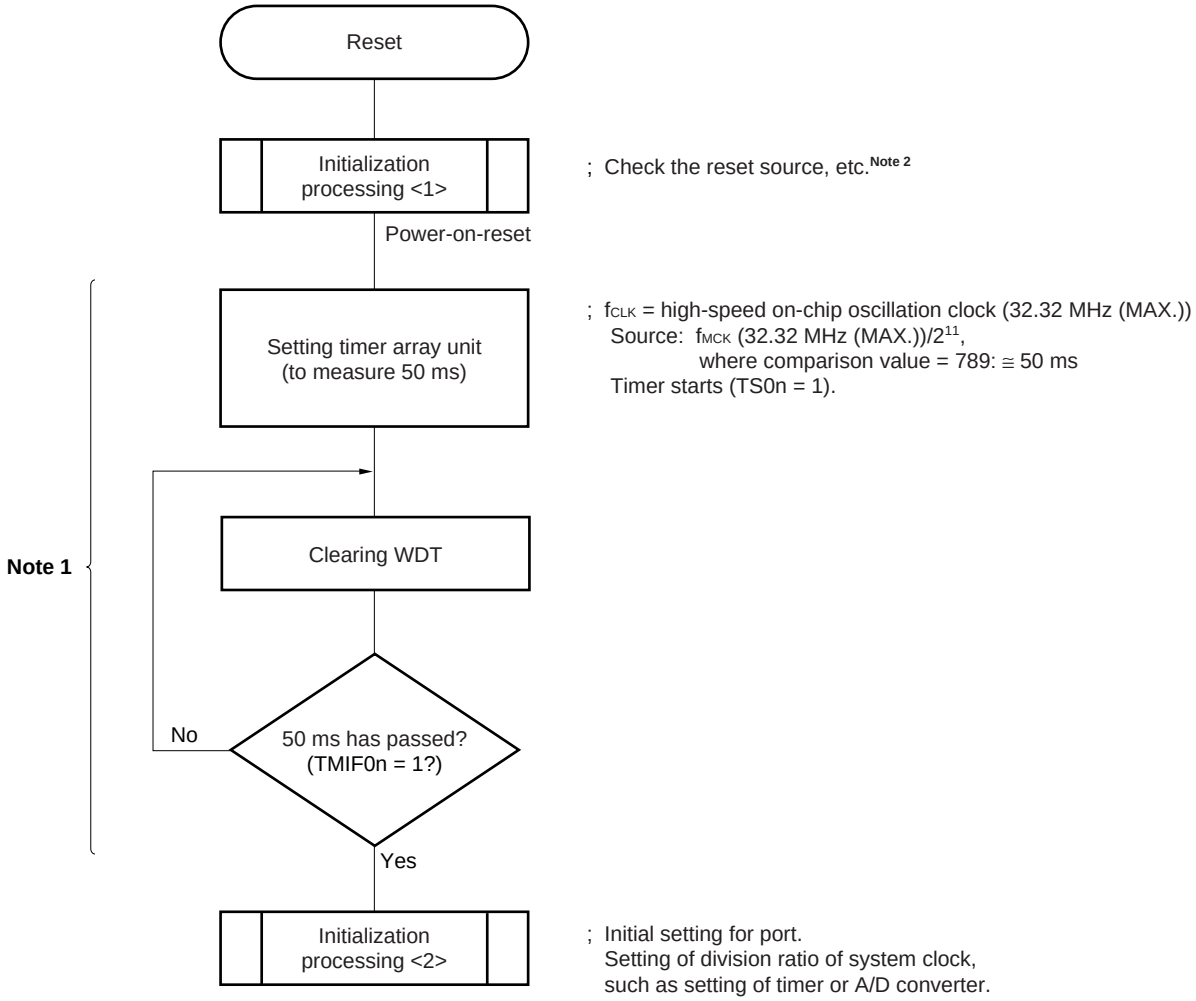
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POR detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 24-3. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POR detection voltage



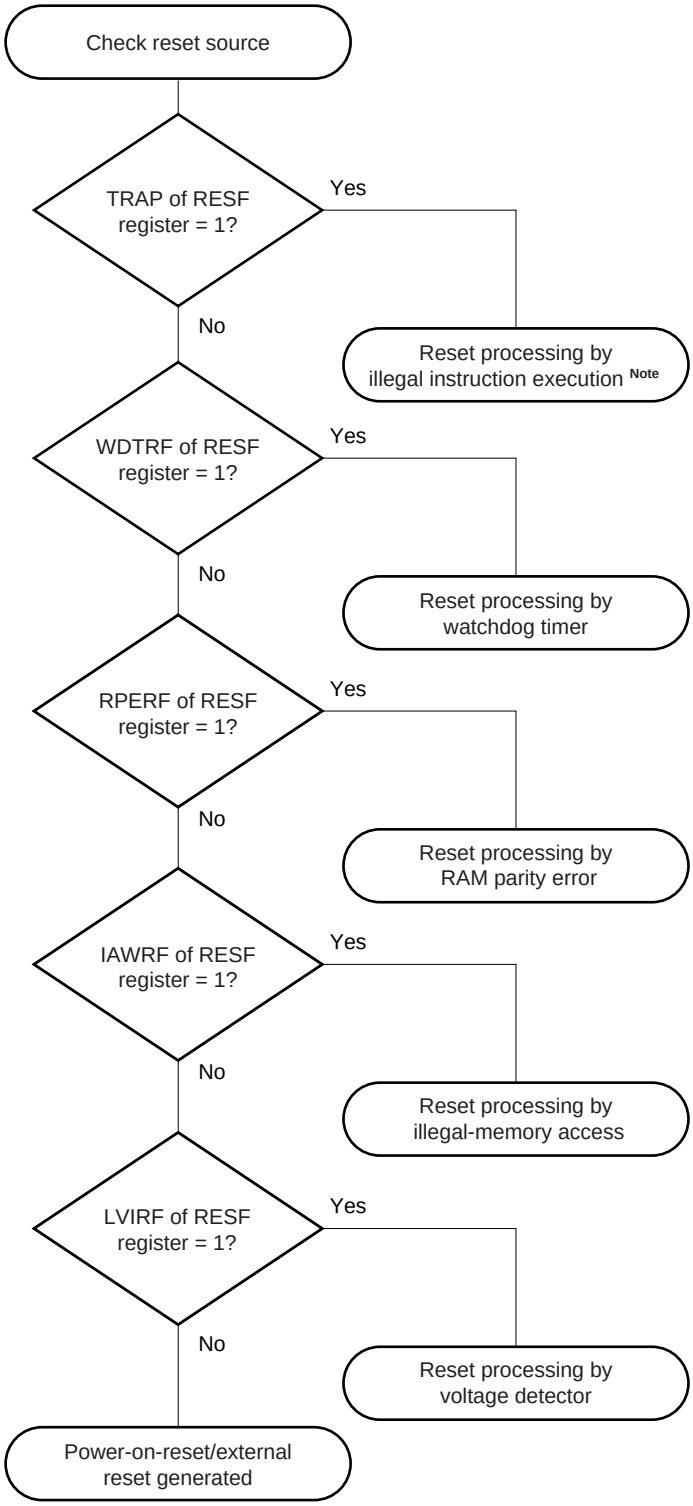
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 0 to 7

Figure 24-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



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

CHAPTER 25 VOLTAGE DETECTOR

25.1 Functions of Voltage Detector

The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}) can be selected by using the option byte as one of 6 levels (For details, see **CHAPTER 28 OPTION BYTE**).
- Operable in STOP mode.
- The following three operation modes can be selected by using the option byte.

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)

For the two detection voltages selected by the option byte 000C1H/020C1H, the high-voltage detection level (V_{LVDH}) is used for generating interrupts and ending resets, and the low-voltage detection level (V_{LVDL}) is used for triggering resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)

The detection voltage (V_{LVD}) selected by the option byte 000C1H/020C1H is used for triggering and ending resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)

The detection voltage (V_{LVD}) selected by the option byte 000C1H/020C1H is used for generating interrupts.

Two detection voltages (V_{LVDH}, V_{LVDL}) can be specified in the interrupt & reset mode, and one (V_{LVD}) can be specified in the reset mode and interrupt mode.

The reset and interrupt signals are generated as follows according to the option byte (LVIMDS0, LVIMDS1) selection.

Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an internal interrupt signal when $V_{DD} < V_{LVDH}$, and an internal reset when $V_{DD} < V_{LVDL}$. Releases the reset signal when $V_{DD} \geq V_{LVDH}$.	Generates an internal reset signal when $V_{DD} < V_{LVD}$ and releases the reset signal when $V_{DD} \geq V_{LVD}$.	The state of an internal reset by LVD is retained until $V_{DD} \geq V_{LVD}$ immediately after reset generation. The internal reset is released when $V_{DD} \geq V_{LVD}$ is detected. After that, an interrupt request signal (INTLVI) is generated when $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ is detected.

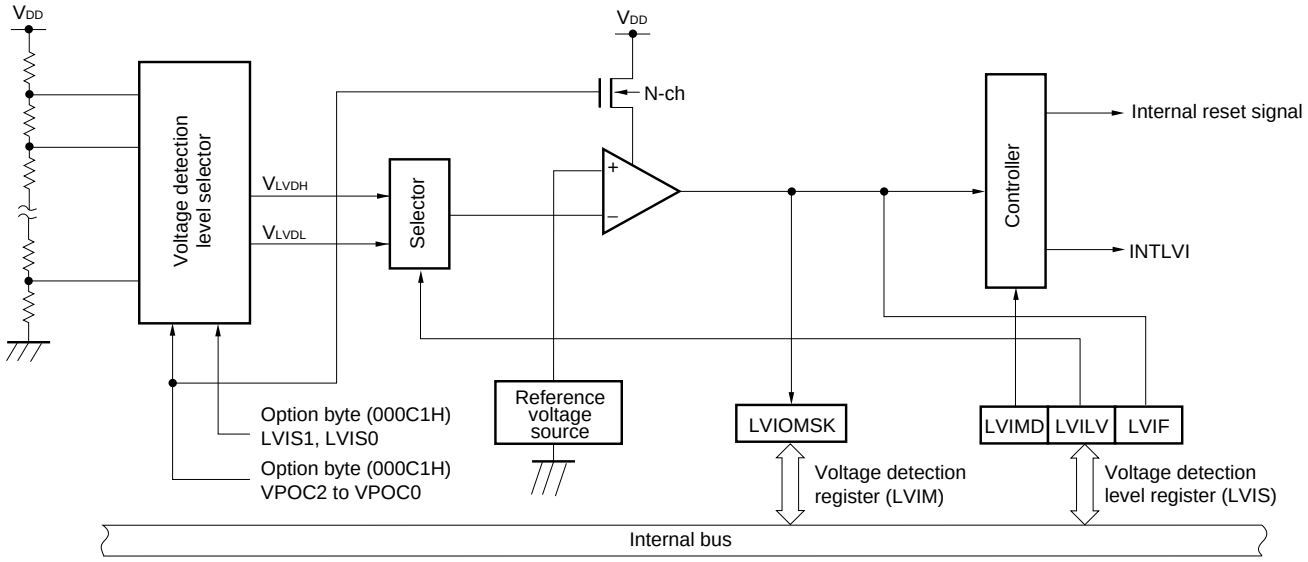
While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 23 RESET FUNCTION**.

25.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 25-1.

Figure 25-1. Block Diagram of Voltage Detector



25.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

(1) Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

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

Figure 25-2. Format of Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN	0	0	0	0	0	LVIOMSK	LVIF

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling rewriting
1	Enabling rewriting Note 3

LVIOMSK	Mask status flag of LVD output
0	Mask is invalid
1	Mask is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. Only the bit 7 of this register is cleared by “0”, if a reset other than by LVD is effective.

2. Bits 0 and 1 are read-only.

3. This can only be set when LVIMDS1 and LVIMDS0 are set to 1 and 0 (interrupt and reset mode) by the option byte.

4. LVIOMSK bit is automatically set to “1” in the following periods and reset or interruption by LVD is masked.

- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable.

(2) Voltage detection level register (LVIS)

This register selects the voltage detection level.

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

Reset signal generation input sets this register to 00H/01H/81H [Note1].

Figure 25-3. Format of Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 00H/01H/81H [Note 1] R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD[Note 2]	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV[Note 2]	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

2. Writing “0” can only be allowed when LVIMDS1 and LVIMDS0 are set to 1 and 0 (interrupt and reset mode) by the option byte. In other cases, writing is not allowed and the value is switched automatically when reset or interrupt is generated.

Cautions 1. Only rewrite the value of the LVIS register after setting the LVISEN bit (bit 7 of the LVIM register) to 1.

2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte (000C1H). Table 25-1 shows the option byte (000C1H) settings. For details about the option byte, see CHAPTER 28 OPTION BYTE.

Table 25-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H/020C1H)

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	Mode setting						
Rise	Fall	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
2.92 V	2.86 V	2.75 V	1	0	0	1	1	1	0
3.02 V	2.96 V							0	1
4.06 V	3.98 V							0	0
Other than the above			Setting prohibited						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		Mode setting						
Rise	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
2.81 V	2.75 V	1	1	0	1	1	1	1
2.92 V	2.86 V				1	1	1	0
3.02 V	2.96 V				1	1	0	1
3.13 V	3.06 V				0	1	0	0
3.75 V	3.67 V				1	0	0	0
4.06 V	3.98 V				1	1	0	0
Other than the above		Setting prohibited						

Remark ×: Don't care.

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		Mode setting						
Rise	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
2.81 V	2.75 V	0	1	0	1	1	1	1
2.92 V	2.86 V				1	1	1	0
3.02 V	2.96 V				1	1	0	1
3.13 V	3.06 V				0	1	0	0
3.75 V	3.67 V				1	0	0	0
4.06 V	3.98 V				1	1	0	0
Other than the above		Setting prohibited						

Remark ×: Don't care.

Caution External $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

- LVD setting (LVD off)

Detection voltage		Option byte setting value						
V_{LVD}		Mode setting						
Rise	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
–	–	0/1	1	1	×	×	×	×
Other than the above		Setting prohibited						

Remark ×: Don't care.

Caution External $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

25.4 Operation of Voltage Detector

25.4.1 When used as reset mode

- When starting operation

  Start in the following initial setting state.

  Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H/020C1H.

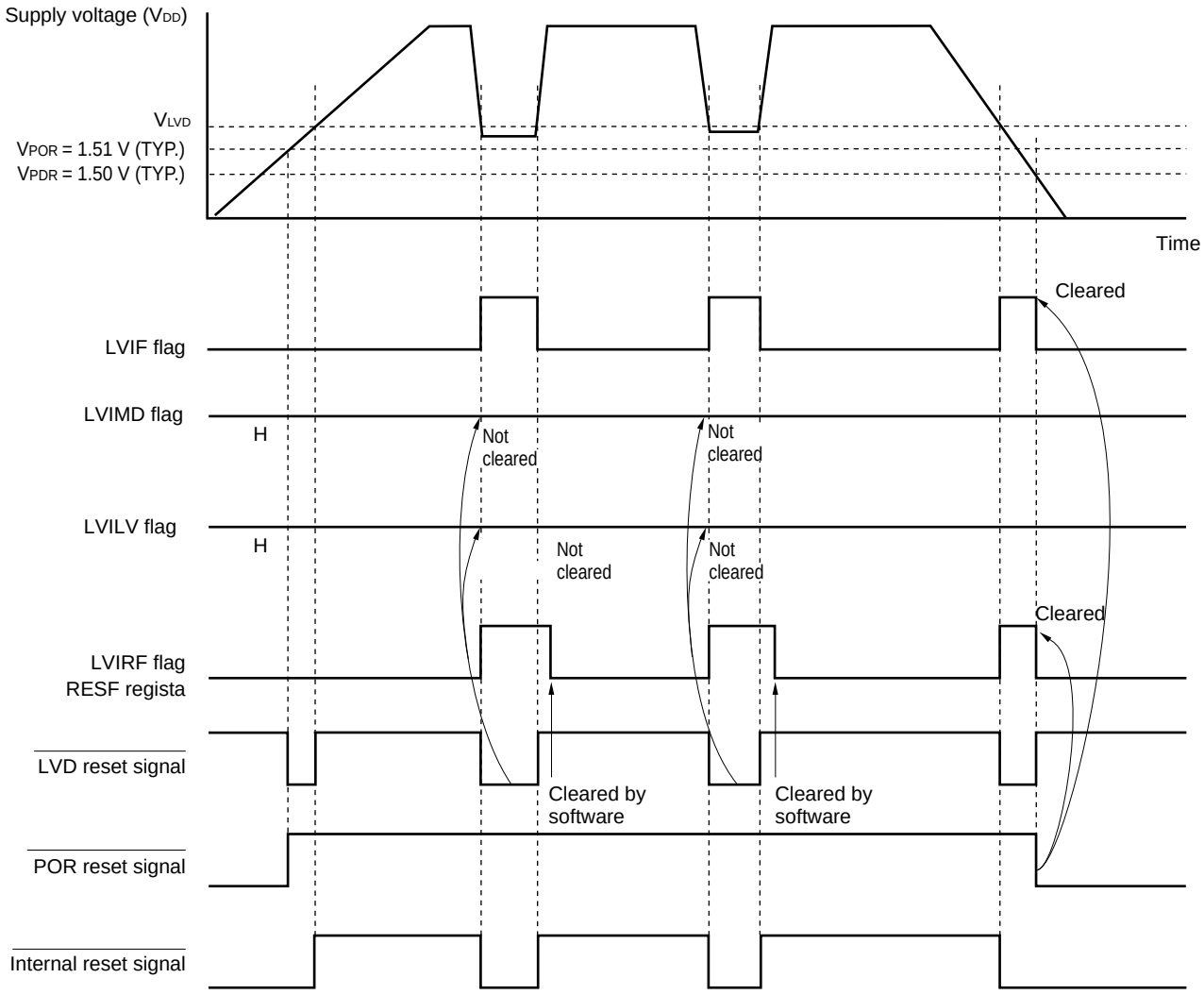
  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS)).
  - When the option byte LVIMDS1 and LVIMDS0 are set to 1, the initial value of the LVIS register is set to 81H.

    Bit 7 (LVIMD) is 1 (reset mode).

    Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

  Figure 25-4 shows the timing of the internal reset signal generated by the voltage detector.

Figure 25-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

25.4.2 When used as interrupt mode

- When starting operation

  Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H/020C1H.

  Start in the following initial setting state.

  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS)).
  - When the option byte LVIMDS1 is clear to 0 and LVIMDS0 is set to 1, the initial value of the LVIS register is set to 01H.

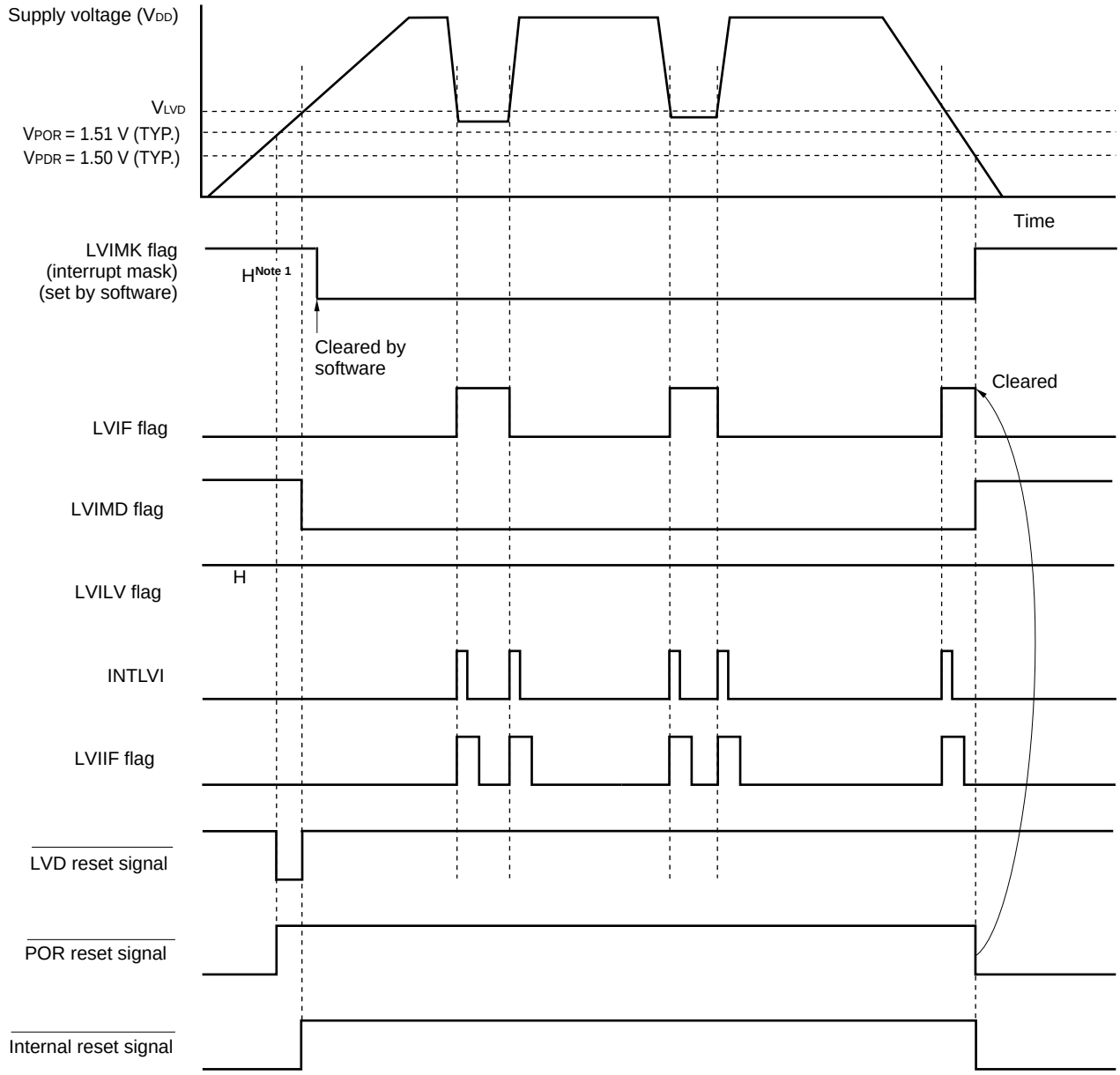
    Bit 7 (LVIMD) is 0 (interrupt mode).

    Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

  Figure 25-5 shows the timing of the internal interrupt signal generated by the voltage detector.

  Caution External $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

Figure 25-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note The LVIMK flag is set to “1” by reset signal generation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

25.4.3 When used as interrupt and reset mode

- When starting operation

  Specify the operation mode (the interrupt and reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H/020C1H.

  Start in the following initial setting state.

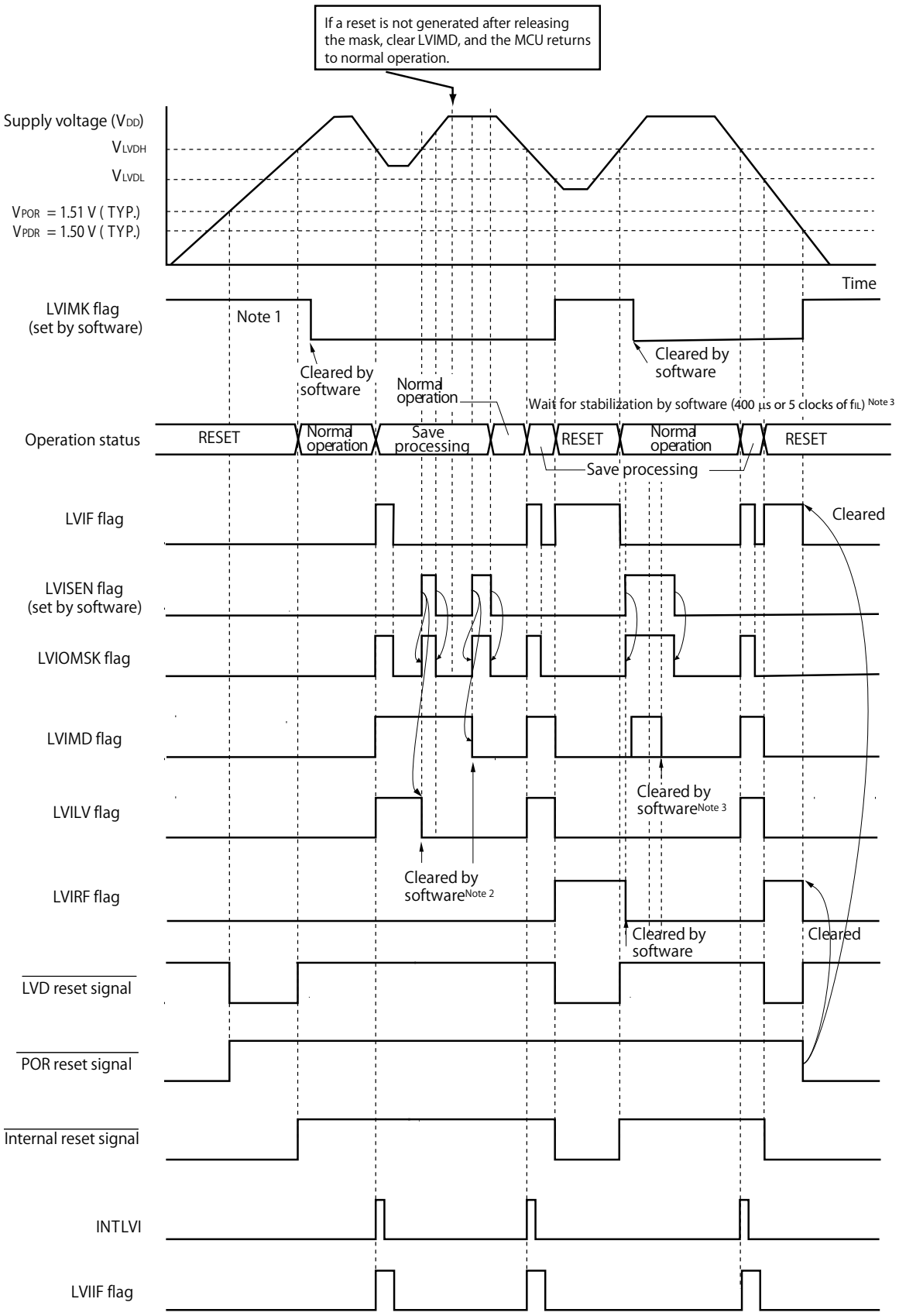
  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS)).
  - When the option byte LVIMDS1 is set to 1 and LVIMDS0 is cleared to 0, the initial value of the LVIS register is set to 00H.

    Bit 7 (LVIMD) is 0 (interrupt mode).

    Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

  Figure 25-6 shows the timing of the internal reset signal and interrupt signal generated by the voltage detector. Perform the processing according to **Figure 25-7 Processing Procedure after an Interrupt is Generated** and **Figure 25-8 Initial Setting of Interrupt and Reset Mode.**

Figure 25-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



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

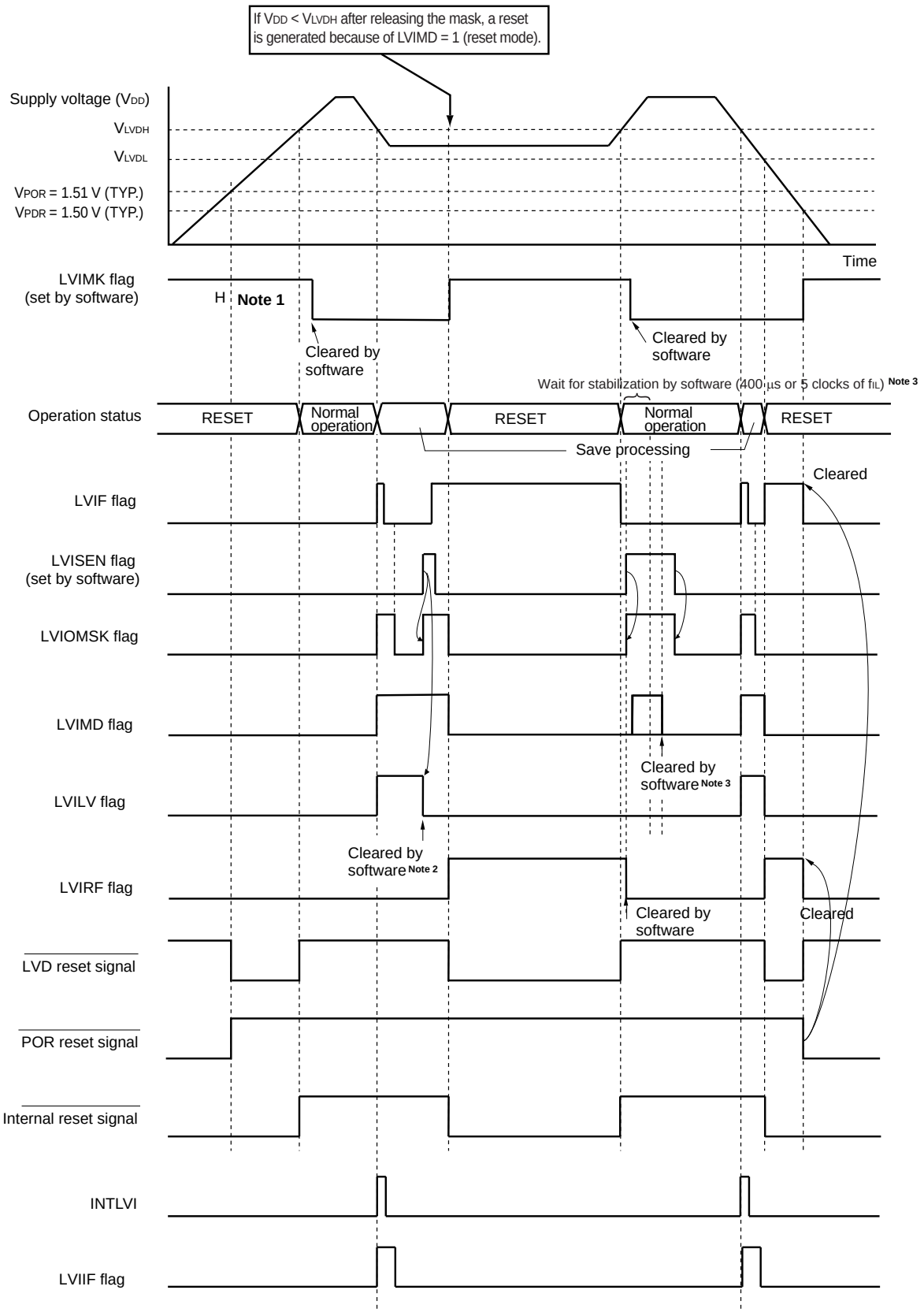
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 25-7 Processing Procedure after an Interrupt is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 25-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 25-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



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

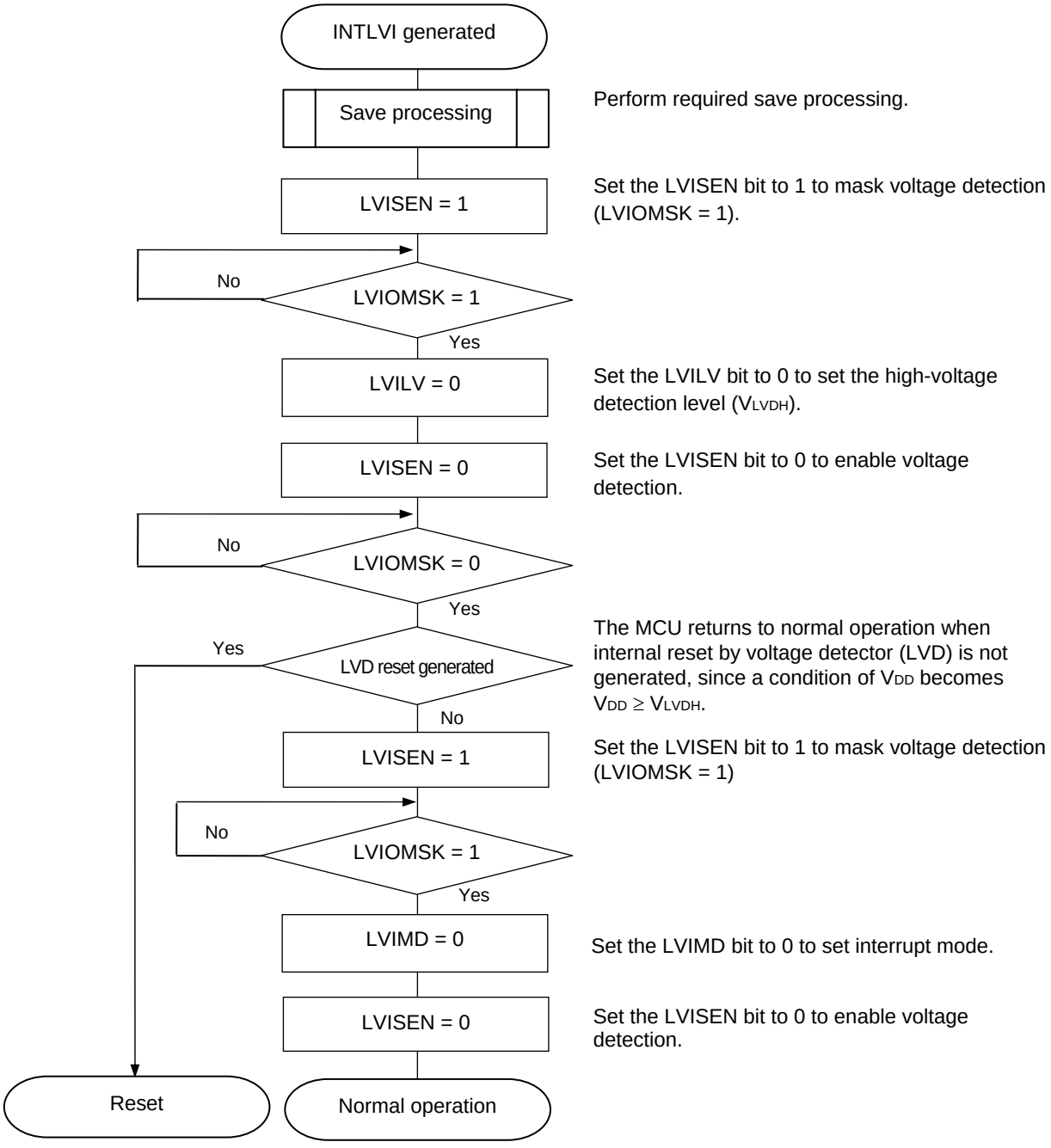
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 25-7 Processing Procedure after an Interrupt is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 25-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

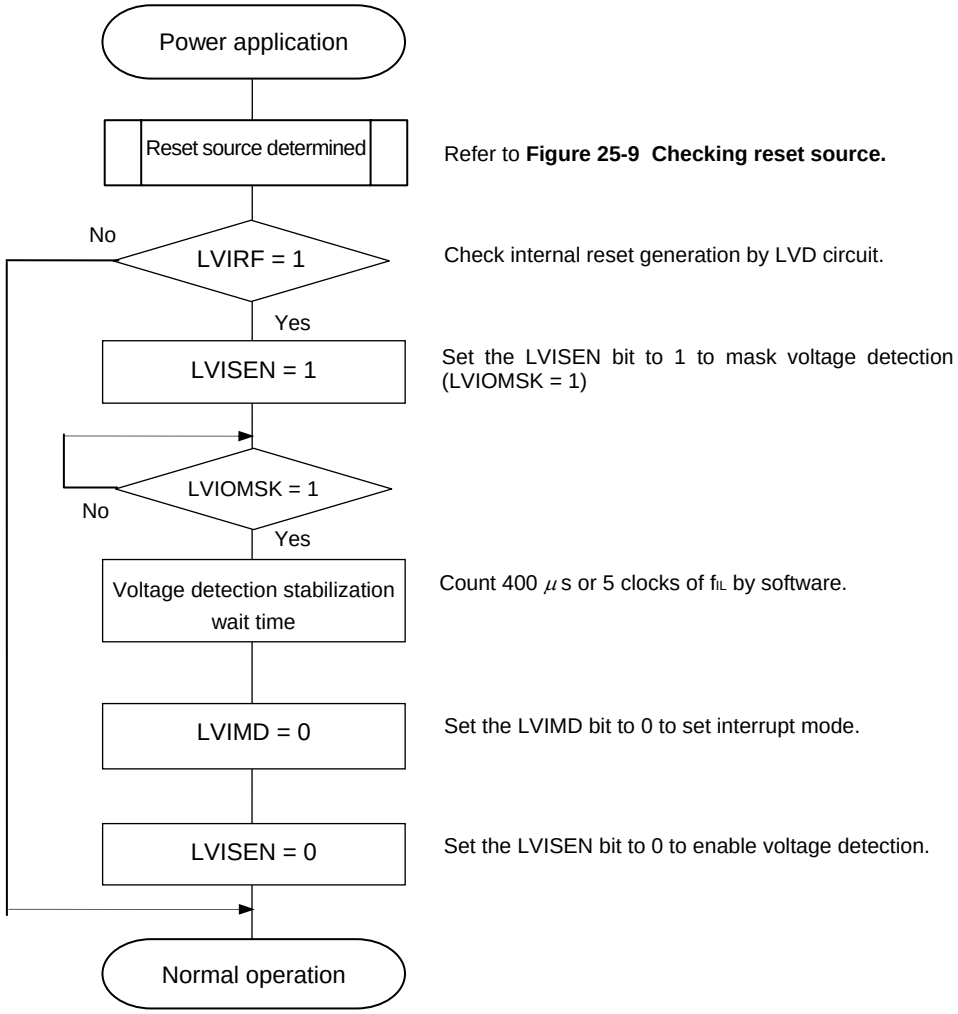
Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 25-7. Processing Procedure after an Interrupt is Generated



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 μs or 5 clocks of f_{IL} is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, clear the LVIMD bit to 0 for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.
Figure 24-8 shows the procedure for initial setting of interrupt and reset mode.

Figure 25-8. Initial Setting of Interrupt and Reset Mode



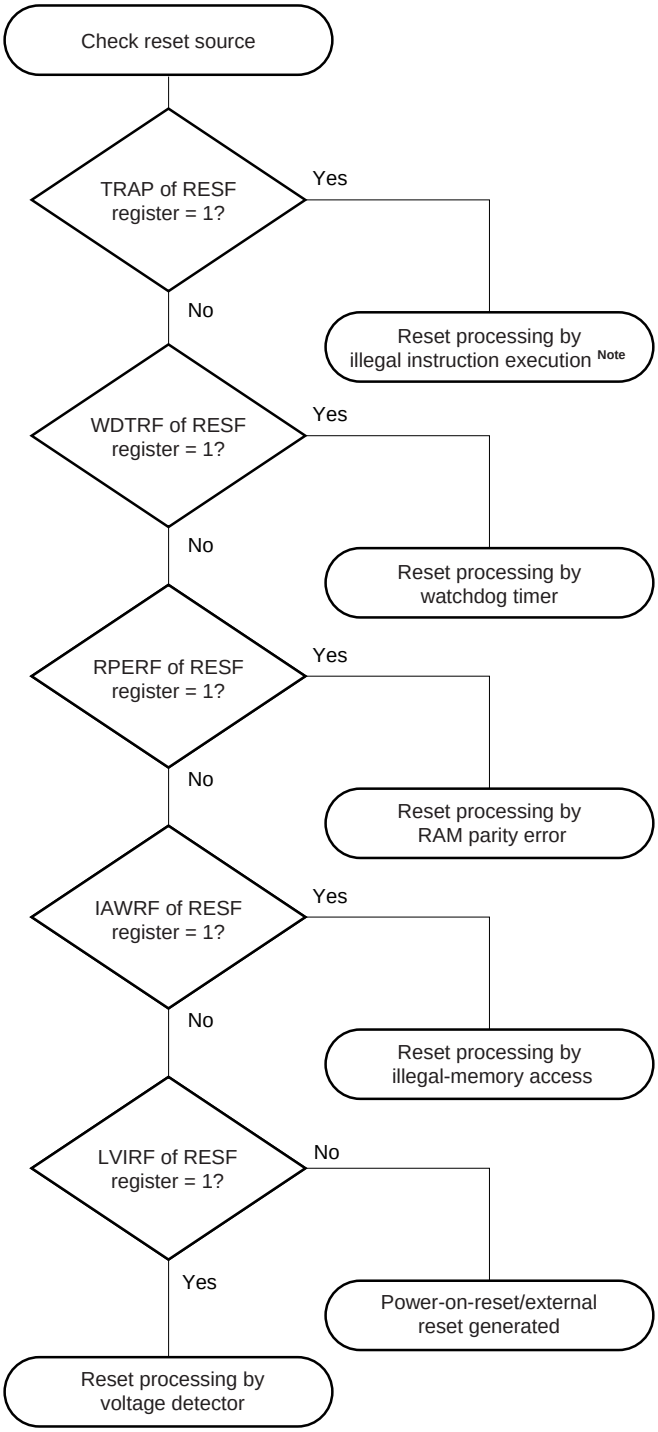
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

25.5 Cautions for Voltage Detector

(1) Checking reset source

When a reset occurs, check the reset source by using the following method.

Figure 25-9. Checking reset source



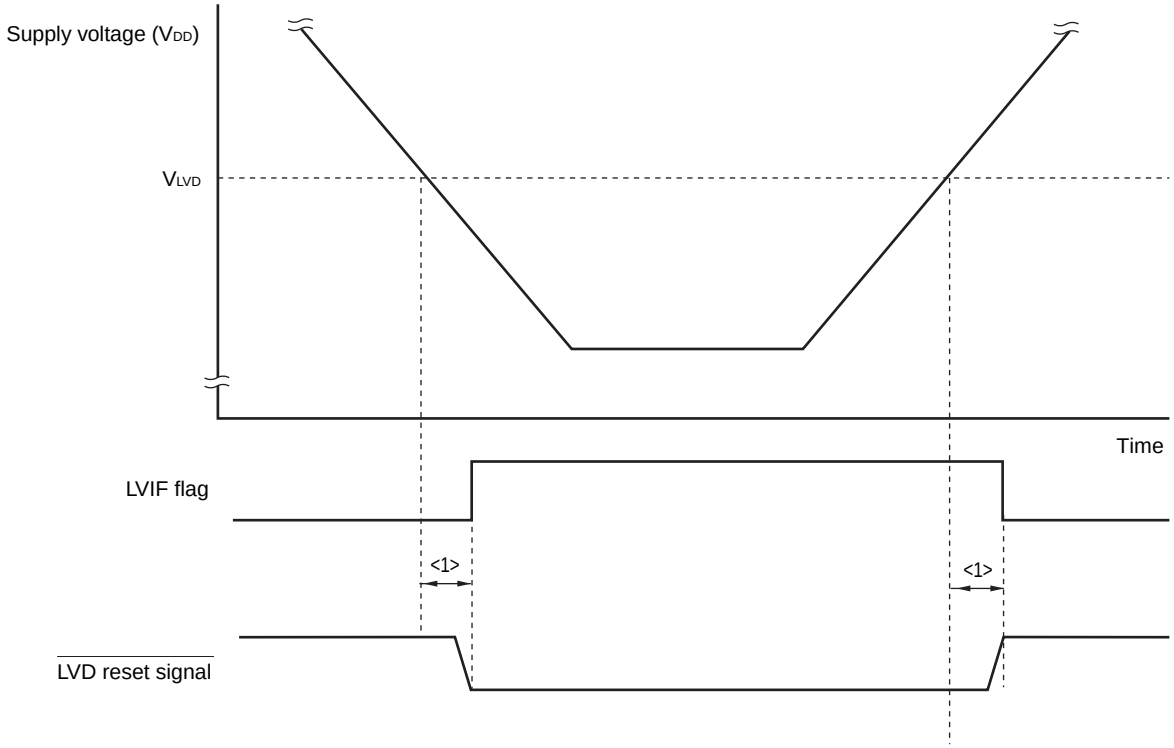
Note When instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 25-10**).

Figure 25-10. Delay from the Time LVD Reset Source is Generated until the Time LVD Reset Has been Generated or Released



<R> <1>: Detection delay (300 μs (MAX.))

CHAPTER 26 SAFETY FUNCTIONS

26.1 Overview of Safety Functions

The RL78/D1A is provided with the following safety functions to meet the IEC 60730 and IEC 61508 safety standards. The functions are intended to detect failures through microcomputer's self-diagnosis and to stop the system safely.

(1) Flash memory CRC operation function (high-speed CRC and general-purpose CRC)

This detects errors associated to data in the flash memory by performing CRC operations.
Either of the CRC operations below can be used according to the application and conditions of use.

- High-speed CRC: Can be used for high-speed check of the entire code flash memory area while the CPU is stopped in the initialization routine.
- General-purpose CRC: Can be used for not only check of code flash memory area but versatile check while the CPU is running.

(2) RAM parity error detection function

This detects parity errors when the RAM is read as data.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function

This detects access to invalid memory areas (non-existent areas or access-restricted areas).

(6) Frequency detection function

This uses TAU to detect the oscillation frequency.

(7) A/D test function

This is used to perform a self-check of A/D conversion by performing A/D conversion on the internal reference voltage.

(8) Clock monitor function

This detects the stop of oscillation of main system clock (f_{MAIN}) and PLL clock (f_{PLL}).For more details, see **5.4 Clock monitor (CLM)**.

26.2 Registers Used by Safety Functions

Each safety function uses the following registers.

Register Name	Safety Functions
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function SFR guard function Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Analog input channel specification register (ADS)	Specification of the analog voltage input channel

For details of the registers, refer to **26.3 Operations of Safety Functions**.

26.3 Operations of Safety Functions

26.3.1 Flash Memory CRC Operation Function (High-Speed CRC)

The IEC 60730 standard requires verification of data in flash memory and recommends CRC as the means for verification. With the high-speed CRC operation function, the entire code flash memory area can be checked in the initialization routine. This function is available only in HALT mode of the main system clock through the program on RAM.

With the high-speed CRC operation function, the CPU is stopped and a 32-bit data unit is read from the flash memory in a clock cycle to perform operation on the data. Therefore, check can be completed in a short time (for example, 64-Kbyte flash memory checked in 512 μs at 32 MHz frequency).

The CRC generator polynomial is $X^{16} + X^{12} + X^5 + 1$ of CRC-16-CCITT.

Operation is done with the MSB first, i.e., from bit 31 to bit 0.

Caution Since the monitor program is allocated for on-chip debugging, a different operation result is obtained.

Remark Since the general-purpose CRC uses the LSB-fist method, a different operation result is obtained.

26.3.1.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range.

The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 26-1. Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High- speed CRC operation range
0	0	0	0	0	0	0 to 3FFBH (16 K - 4 bytes)
0	0	0	0	0	1	0 to 7FFBH (32 K - 4 bytes)
0	0	0	0	1	0	0 to BFFBH (48 K - 4 bytes)
0	0	0	0	1	1	0 to FFFBH (64 K - 4 bytes)
0	0	0	1	0	0	0 to 13FFBH (80 K to 4 bytes)
0	0	0	1	0	1	0 to 17FFBH (96 K to 4 bytes)
0	0	0	1	1	0	0 to 1BFFBH (112 K to 4 bytes)
0	0	0	1	1	1	0 to 1FFFBH (128 K to 4 bytes)
0	0	1	0	0	0	0 to 23FFBH (144 K to 4 bytes)
0	0	1	0	0	1	0 to 27FFBH (160 K to 4 bytes)
0	0	1	0	1	0	0 to 2BFFBH (176 K to 4 bytes)
0	0	1	0	1	1	0 to 2FFFBH (192 K to 4 bytes)
0	0	1	1	0	0	0 to 33FFBH (208 K to 4 bytes)
0	0	1	1	0	1	0 to 37FFBH (224 K to 4 bytes)
0	0	1	1	1	0	0 to 3BFFBH (240 K to 4 bytes)
0	0	1	1	1	1	0 to 3FFFBH (256 K to 4 bytes)
0	1	0	0	0	0	0 to 43FFBH (272 K to 4 bytes)
0	1	0	0	0	1	0 to 47FFBH (288 K to 4 bytes)
0	1	0	0	1	0	0 to 4BFFBH (304 K to 4 bytes)
0	1	0	0	1	1	0 to 4FFFBH (320 K to 4 bytes)
0	1	0	1	0	0	0 to 53FFBH (336 K to 4 bytes)
0	1	0	1	0	1	0 to 57FFBH (352 K to 4 bytes)
0	1	0	1	1	0	0 to 5BFFBH (368 K to 4 bytes)
0	1	0	1	1	1	0 to 5FFFBH (384 K to 4 bytes)
0	1	1	0	0	0	0 to 63FFBH (400 K to 4 bytes)
0	1	1	0	0	1	0 to 67FFBH (416 K to 4 bytes)
0	1	1	0	1	0	0 to 6BFFBH (432 K to 4 bytes)
0	1	1	0	1	1	0 to 6FFFBH (448 K to 4 bytes)
0	1	1	1	0	0	0 to 73FFBH (464 K to 4 bytes)
0	1	1	1	0	1	0 to 77FFBH (480 K to 4 bytes)
0	1	1	1	1	0	0 to 7BFFBH (496 K to 4 bytes)
0	1	1	1	1	1	0 to 7FFFBH (512 K to 4 bytes)
Other than the above						Setting prohibited

Remark In the last 4 bytes of the flash memory, store in advance the expected value of the CRC operation result for comparison. The above table thus shows the operation range that is smaller by 4 bytes.

26.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 26-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

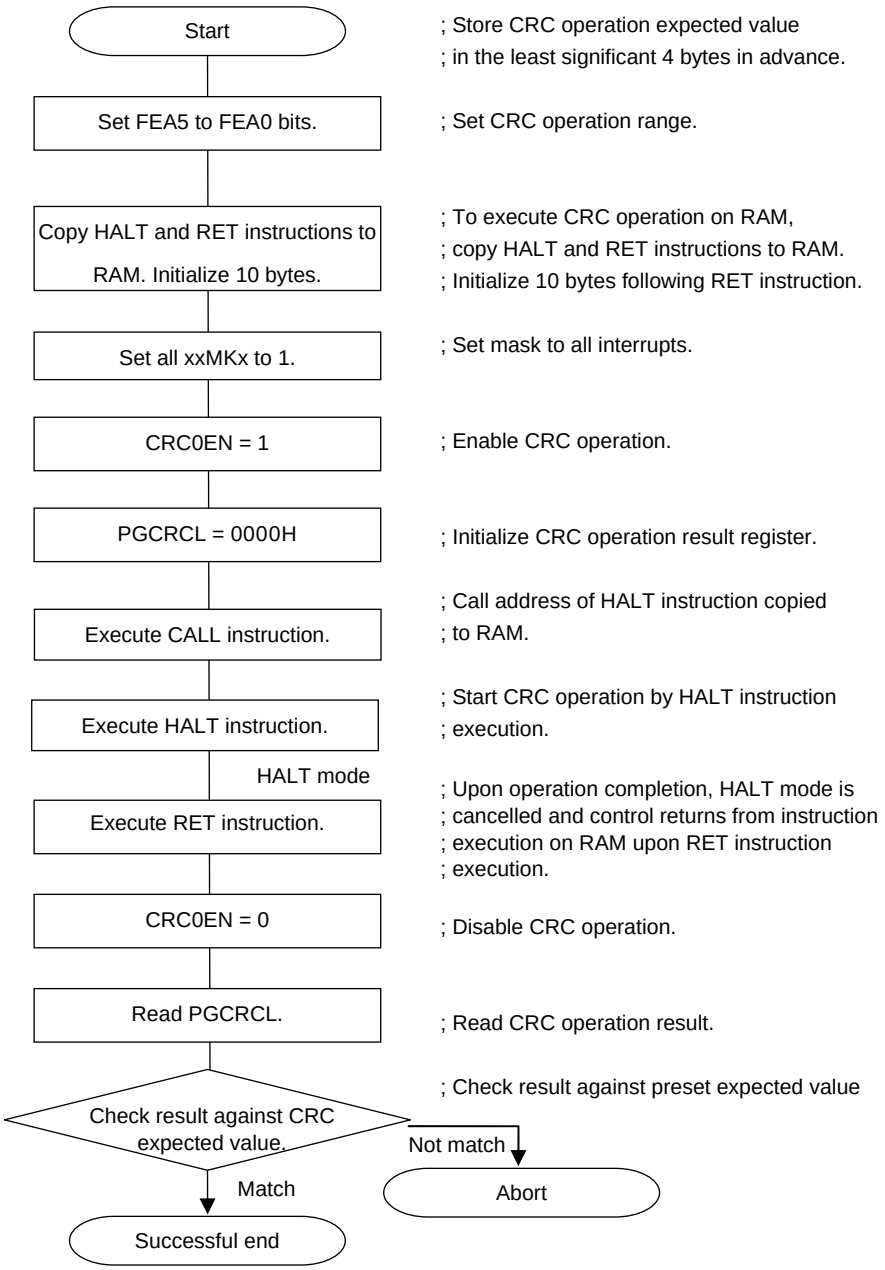
Symbol	7	6	5	4	3	2	1	0
PGCRCL	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written to if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 26-3 shows a flowchart of the flash memory CRC operation function (high-speed CRC).

Figure 26-3. Flowchart of Flash Memory CRC Operation Function (High-Speed CRC)



Cautions 1. Only code flash memory is subject to CRC operation.

2. Store the CRC operation expected value in the area following the operation range of the code flash memory.

3. Boot swapping is not performed during CRC operation.

4. Executing the HALT instruction in the RAM area enables CRC operation; be sure to execute the HALT instruction in the RAM area.

The CRC expected value can be calculated using the development environment CubeSuite+ or the equivalents (refer to the CubeSuite+ user's manual.

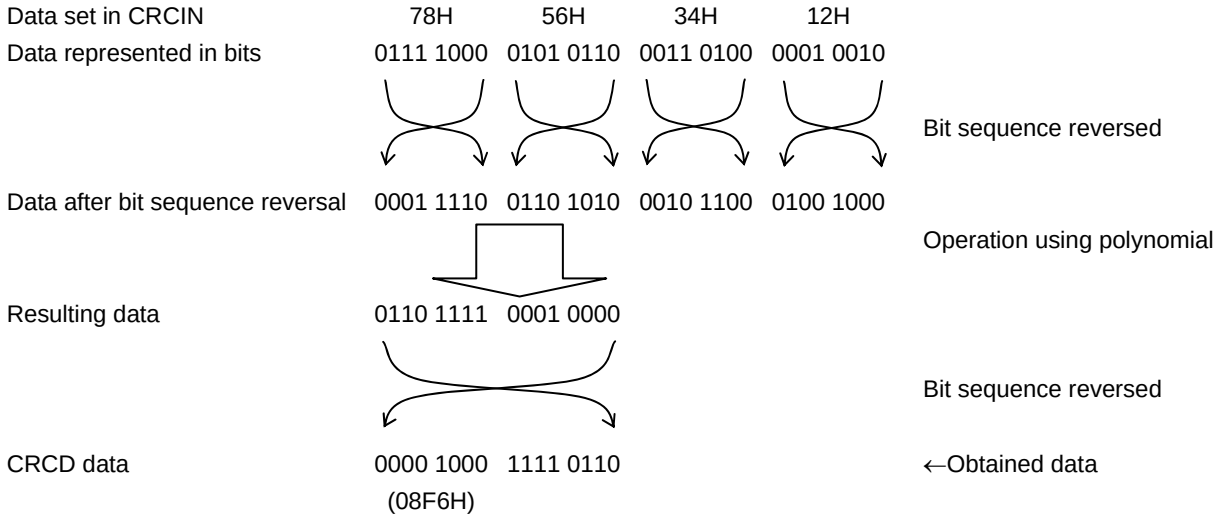
26.3.2 CRC Operation Function (General-Purpose CRC)

The IEC 61508 standard requires safety to be guaranteed during operation and thus the means for data verification during CPU operation is necessary.

With the general-purpose CRC operation function, the CRC operation is possible as a peripheral function during CPU operation. The general-purpose CRC operation function can be used for not only check of code flash memory area but versatile check. Data to be checked is specified with the user program. In HALT mode, the CRC operation function is available only during DMA transfer.

The CRC operation function is available both in main and subsystem clock operation modes.

The CRC generator polynomial is $X^{16} + X^{12} + X^5 + 1$ of CRC-16-CCITT. Operation is done after the input data is reversed in bit sequence to accommodate to the LSB-first communication. For example, when data 12345678H is to be transmitted with the LSB first, writing 78H, 56H, 34H, and 12H in CRCIN register in this order allows value 08E6H to be obtained from the CRCD register. This value is obtained as shown below, where data 12345678H is reversed in bit sequence and then the resulting bit strings are subjected to CRC operation.



Caution During program execution, the debugger replaces the line in which software break is set with the break instruction; therefore, setting a software break in the area subject to CRC operation causes a different operation result to be obtained.

26.3.2.1 CRC input register (CRCIN)

This is an 8-bit register used to set the data for general-purpose CRC operation.

The possible setting range is 00H to FFH.

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

Reset signal generation clears this register to 00H.

Figure 26-4. Format of CRC Input Register (CRCIN)

Address: FFFACH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input

26.3.2.2 CRC data register (CRCD)

This register is used to store the general-purpose CRC operation result.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written to, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 26-5. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

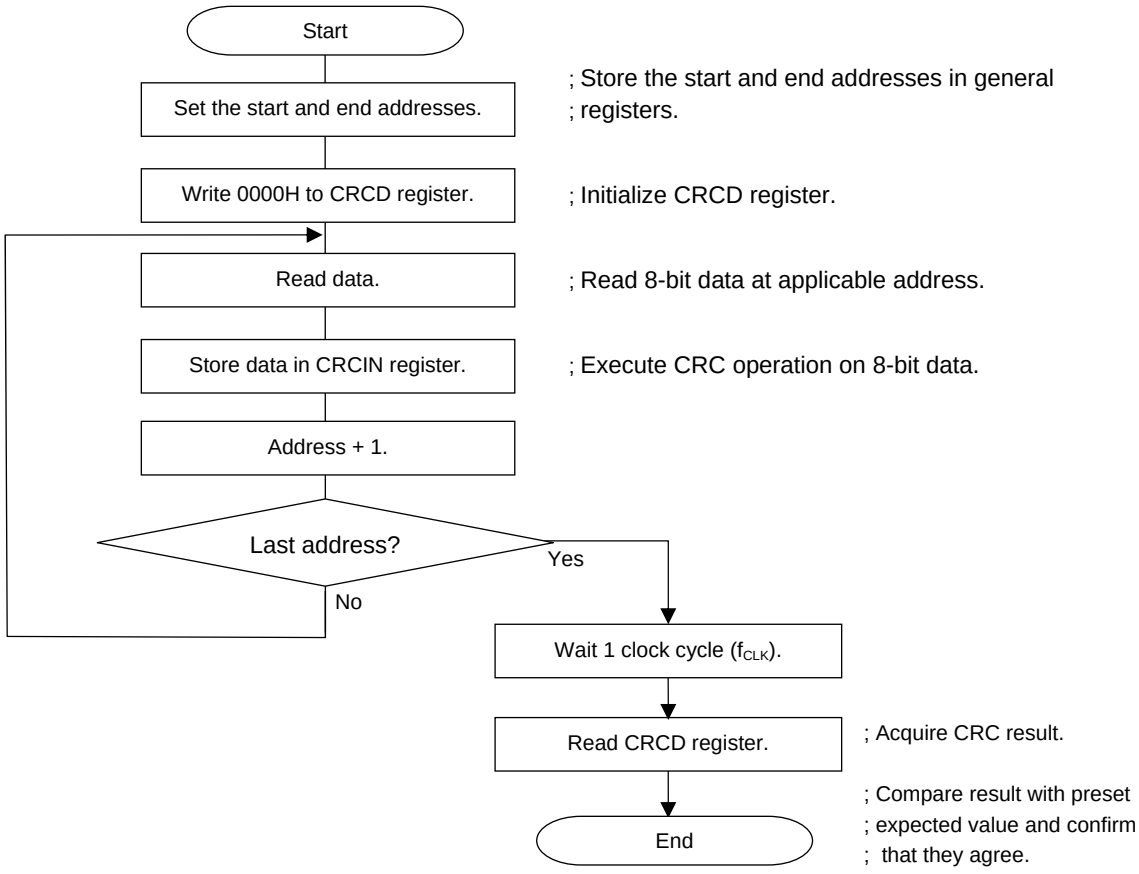
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If writing and storing operation result to CRCD register conflict, the writing is ignored.

<Operation flow>

Figure 26-6. Flowchart of CRC Operation Function (General-Purpose CRC)



26.3.3 RAM Parity Error Detection Function

The IEC 60730 standard requires verification of RAM data. To meet the requirement, one parity bit is appended to each 8-bit data in the RL78/D1A RAM. With the RAM parity error detection function, a parity is written when data is written and the parity is checked when data is read out. A reset can be generated upon occurrence of a parity error.

26.3.3.1 RAM parity error control register (RPECTL)

This register is used to check occurrence of a parity error and control resets due to parity errors.

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

Reset signal generation clears this register to 00H.

Figure 26-7. Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Parity error reset mask flag
0	Enables parity error resets.
1	Disables parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution With the RL78, the CPU performs read-ahead for pipeline operation to read the RAM area not yet initialized that follows the currently used RAM area, which may cause a RAM parity error. Therefore, when RAM parity error reset generation is enabled (RPERDIS = 0), be sure to initialize the RAM area to be used and 10 more bytes. When the self-programming function is used, be sure to initialize the RAM area to be rewritten to and 10 more bytes before rewrite.
Parity error detection is performed on RAM data read during RAM instruction fetching.

Remarks 1. The RAM parity check function is always on and the check results can be read from the RPEF flag.

2. In the initial state, parity error reset generation is enabled (RPERDIS = 0). Even if parity error reset generation is disabled (RPERDIS = 1), the RPEF flag is set (1) when a parity error occurs.

3. The RPEF flag is set (1) by RAM parity errors and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

26.3.4 RAM Guard Function

The IEC 61508 standard requires safety to be guaranteed during operation and thus it is necessary to protect significant data stored in RAM when the CPU freezes.

The RAM guard function protects data in the specified space.

Setting this function disables writing to RAM in the specified space but enables reading normally.

26.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard.

The RAM guard function uses the GRAM1 and GRAM0 bits.

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

Reset signal generation clears this register to 00H.

Figure 26-8. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space[Note]
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the lower RAM address
1	0	The 256 bytes starting at the lower RAM address
1	1	The 512 bytes starting at the lower RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product.

26.3.5 SFR Guard Function

The IEC 61508 standard requires safety to be guaranteed during operation and thus it is necessary to prevent significant SFRs from being erroneously rewritten when the CPU freezes.

The SFR guard function protects data in the registers used to control the port function, interrupt function, clock control function, voltage detection circuits, and RAM parity error detection function.

Setting this function disables writing to guarded SFRs but enables reading normally.

26.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard.

The SFR guard function uses the GPORT, GINT, and GCSC bits.

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

Reset signal generation clears this register to 00H.

Figure 26-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Port function control register guard
0	Disabled. Port function control registers can be read or written to.
1	Enabled. Writing to port function control registers is disabled. Reading is enabled. Guarded SFRs: PMxx, PUxx, PIMxx, POMxx, TISxx, TOSxx, TISELSE, STSELx, SGSEL, RTCSEL, LCDPFxx, SMPC and ADPC Note 1

GINT	Interrupt function control register guard
0	Disabled. Interrupt function control registers can be read or written to.
1	Enabled. Writing to interrupt function control registers is disabled. Reading is enabled. Guarded SFRs: IFxx, MKxx, PRxx, EGPx, EGNx

GCSC Note 2	Clock control function, voltage detection circuit, and RAM parity error detection function control register guard
0	Disabled. Registers to control port function, interrupt function, clock control function, voltage detection circuits, and RAM parity error detection function can be read or written to.
1	Enabled. Writing to registers to control port function, interrupt function, clock control function, voltage detection circuits, and RAM parity error detection function is disabled. Reading is enabled. Guarded SFRs: CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, PCKSEL, MDIV and RPECTL

Notes 1. Pxx (port registers) are not guarded.

2. Set GCSC to 0 for self-programming.

26.3.5.2 Specific register manipulation protection register (GUARD)

This register is used to control the guard function.

GDRTC and GDPLL bits are used in SFR guard function.

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

Reset signal generation clears this register to 00H.

Figure 26-10. Format of Specific Register Manipulation Protection Register (GUARD)

Address: F00FCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
GUARD	0	0	0	0	0	0	GDRTC	GDPLL

GDRTC	Protection against RTC clock selection register (RTCCL)
0	RTCCL register can be accessed.
1	Disables RTCCL register (GUARD is effective).

GDPLL	Protection against manipulation of PLL control register (PLLCTL)
0	Enables manipulation of PLLCTL register.
1	Disables manipulation of PLLCTL register (GUARD is effective).

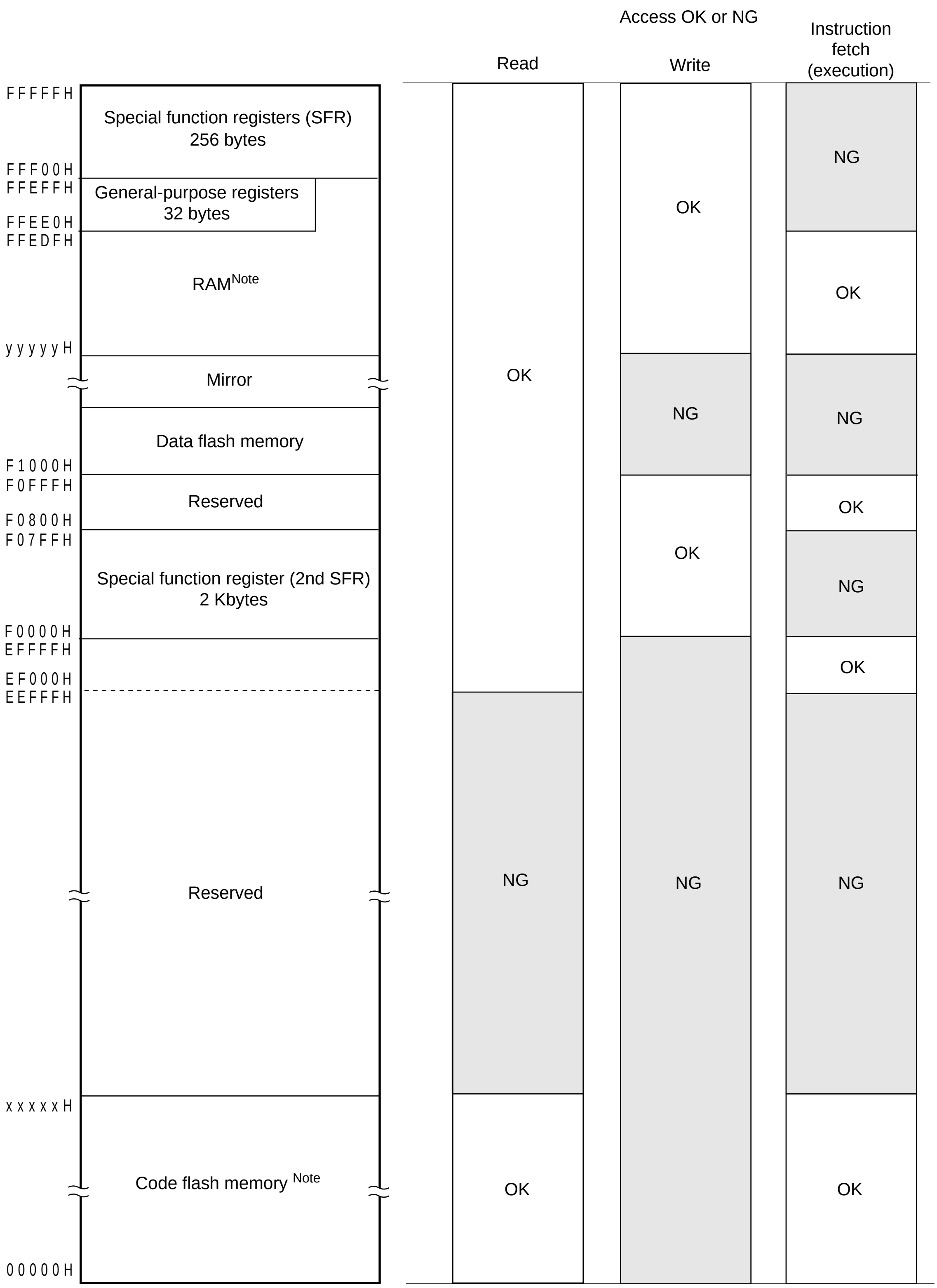
26.3.6 Invalid Memory Access Detection Function

The IEC 60730 standard requires verification of correct operations of the CPU and interrupts.

The invalid memory access detection function allows a reset to be generated when the specified invalid memory access detection space is accessed.

The space in which invalid memory access is to be detected is indicated as NG in figure 26-11.

Figure 26-11. Invalid Memory Access Detection Space



Note The addresses of the RAM and code flash memory are shown below according to the product type.

Products	Code flash memory (00000H to xxxxxH)	RAM (yyyyyH to FFEFFH)	Reserved	
			Specific area	Other area
			Read access: No error is detected. But value 0FFH is read always Fetching instruction: Illegal instruction execution is detected and internal reset generated (RESF.TRAP bit flag set to 1) Write access: Invalid access is detected.	Read access: Invalid access is detected. Fetching instruction: Invalid access is detected. Write access: Invalid access is detected.
R5F10CGBxFB	24576 × 8 bits (00000H to 05FFFH)	2048 × 8 bits (FF700H to FFEFFH)	40960 x 8 bits (06000H to 0FFFFH)	All other “Reserved” area than “Specific area”
R5F10CGCxFB, R5F10DGCxFB	32768 × 8 bits (00000H to 07FFFH)	2048 × 8 bits (FF700H to FFEFFH)	32768 × 8 bits (08000H to 0FFFFH)	All other “Reserved” area than “Specific area”
R5F10CGDxFB, R5F10DGDxFB, R5F10CLDxFB, R5F10DLDxFB, R5F10CMDxFB, R5F10DMDxFB	49152 × 8 bits (00000H to 0BFFFH)	3072 × 8 bits (FF300H to FFEFFH)	16384 × 8 bits (0C000H to 0FFFFH)	All other “Reserved” area than “Specific area”
R5F10DGExFB, R5F10DLExFB, R5F10CMExFB, R5F10DMExFB, R5F10DPExFB	65536 × 8 bits (00000H to 0FFFFH)	4096 × 8 bits (FEF00H to FFEFFH)	Not applicable	All “Reserved” area
R5F10DMFxFB, R5F10DPFxFB	98304 × 8 bits (00000H to 17FFFH)	6144 × 8 bits (FE700H to FFEFFH)	32768 × 8 bits (18000H to 1FFFFH)	All other “Reserved” area than “Specific area”
R5F10DMGxFB, R5F10DPGxFB	131072 × 8 bits (00000H to 1FFFFH)	8192 × 8 bits (FDF00H to FFEFFH)	Not applicable	All “Reserved” area
R5F10DMJxFB, R5F10TPJxFB, R5F10DPJxFB, R5F10DSJxFB	262144 × 8 bits (00000H to 3FFFFH)	16384 × 8 bits (FBF00H to FFEFFH)	Not applicable	All “Reserved” area
R5F10DSKxFB, R5F10DPKxFB	393216× 8 bits (00000H to 5FFFFH)	20480 × 8 bits (FAF00H to FFEFFH)	Not applicable	All “Reserved” area
R5F10DSLxFB, R5F10DPLxFB	524288 × 8 bits (00000H to 7FFFFH)	24576 × 8 bits (F9F00H to FFEFFH)	Not applicable	All “Reserved” area

26.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard.

The invalid memory access detection function uses the IAWEN bit.

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

Reset signal generation clears this register to 00H.

Figure 26-12. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN[Note]	Control of invalid memory access detection
0	Disables the detection of invalid memory access.
1	Enables the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is valid, and writing 0 to it after setting it to 1 is invalid.

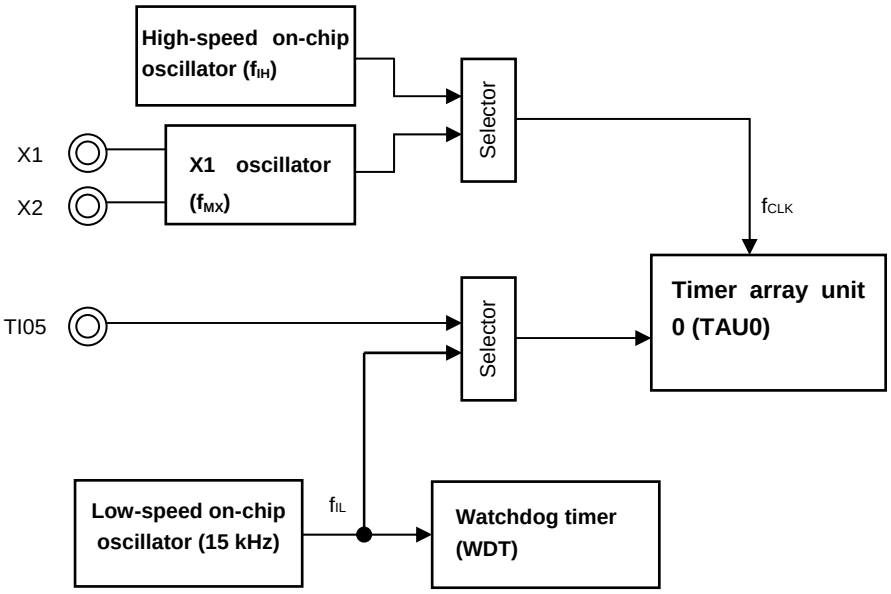
Remark When WDTON = 1 (watchdog timer operation enabled) for the option byte, the invalid memory access detection function is enabled even if IAWEN = 0.

26.3.7 Frequency Detection Function

The IEC 60730 standard requires verification of correct oscillation frequency.

With the frequency detection function, the high-speed on-chip oscillator clock or external X1 oscillator clock is compared with the low-speed on-chip oscillator clock (15 kHz), which allows detection of the clock operating at an abnormal frequency.

Figure 26-13. Configuration of Frequency Detection Function



<Operation summary>

The clock frequency is judged based on the result of pulse interval measurement carried under the following conditions.

- The high-speed on-chip oscillator clock (f_{IH}) or external X1 oscillator clock (f_{MX}) is selected as the CPU/peripheral hardware clock (f_{CLK}).
- The low-speed on-chip oscillator clock (f_{IL}: 15 kHz) is selected as the input to channel 5 of timer array unit 0 (TAU0).

If the pulse interval measurement result is abnormal, the clock frequency is determined to be abnormal. For pulse interval measurement, refer to **6.7.4, Operation as input pulse interval measurement**.

26.3.7.1 Timer input select register 0 (TISELSE)

This register is used to select the timer input of channel 5.

By selecting the internal low-speed oscillation clock for the timer input, its pulse width can be measured to determine whether the proportional relationship between the internal low-speed oscillation clock and the timer operation clock is correct.

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

Reset signal generation clears this register to 00H.

Figure 26-14. Format of Timer Input Select Else Register (TISELSE) for SAFETY FUNCTIONS

Address: FFF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TISELSE	TOTICON1	TOTICON0	0	0	0	0	TI05SEL1	TI05SEL0

TI05SEL1	TI05SEL0	TAU unit 0 channel 5 input alternate selection
0	0	Input after selected by TIS01. bit 2, 3 (TIS051-0)
0	1	Low-speed on-chip oscillator clock (FIL)
1	0	Sub system clock (FSUB)
1	1	Main external clock (FMX)

26.3.8 A/D Test Function

The IEC 60730 standard requires an A/D converter to be tested. With the A/D test function, internal 0 V, AV_{REF}, internal reference voltage (1.45 V) are A/D-converted to verify correct A/D converter operation.

Correct operation of the analog multiplexer can be verified using the following procedure.

(1) Perform A/D conversion of ANIx pin (conversion result 1).

(2) Perform A/D conversion with AV_{REFM} being selected with the ADTES register, and adjust the potential difference at both ends of A/D converter sampling capacitor to 0 V.

(3) Perform A/D conversion of ANIx pin (conversion result 2).

(4) Perform A/D conversion with AV_{REFP} being selected with the ADTES register, and adjust the potential difference at both ends of A/D converter sampling capacitor to AV_{REF}.

(5) Perform A/D conversion of ANIx pin (conversion result 3).

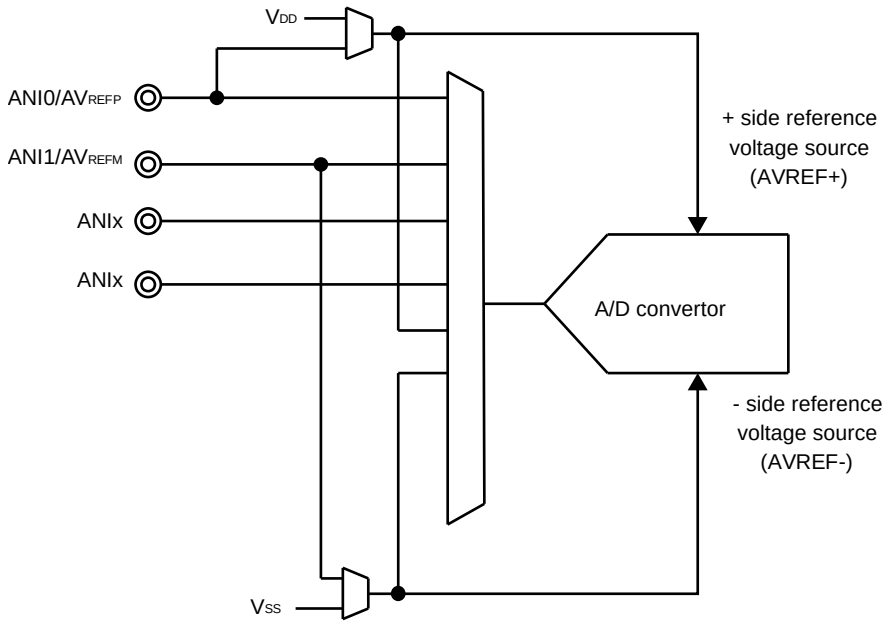
(6) Confirm that conversion results 1, 2, and 3 are identical.

With the above procedure, it can be confirmed that the analog multiplexer is selected and that there is no wire disconnection.

Remarks 1. When the variable analog voltage should be input during conversion in steps 1 through 5, a different method is necessary to check the analog multiplexer.

2. The conversion results include errors; take appropriate errors into consideration when comparing conversion results.

Figure 26-15. A/D Test Function Configuration



26.3.8.1 A/D test register (ADTES)

This register is used to select the AVREFP, AVREFM, or analog input channel (ANIxx) as the A/D conversion target, where AVREFP and AVREFM are reference voltages for the + and – sides, respectively.

When the A/D test function is used, set this register as follows.

- Select AVREFM as the A/D conversion target to measure internal 0 V.
- Select AVREFP as the A/D conversion target to measure AVREF.

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

Reset signal generation clears this register to 00H.

Figure 26-16. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

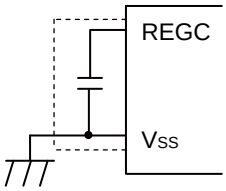
Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	ADTES2	ADTES1	ADTES0

ADTES2	ADTES1	ADTES0	A/D conversion target
0	0	0	ANIxx (This is specified using the analog input channel specification register (ADS).)
0	1	0	AVREFM
0	1	1	AVREFP
Other than the above			Setting prohibited

CHAPTER 27 REGULATOR

27.1 Regulator Overview

The RL78/D1A contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



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

The regulator output voltage, see **table 27-1**.

Table 27-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
High-speed main mode	1.8 V	In STOP mode
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
	2.1 V	Other than the above (include during OCD mode) Note

Note When it shifts to the subsystem clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

CHAPTER 28 OPTION BYTE

28.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/D1A 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.

28.1.1 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 LVD operation mode
  - Interrupt & reset mode.
  - Reset mode.
  - Interrupt mode.
- ○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

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

- ○ Setting of flash operation mode
  - HS (high speed main) mode
- ○ Setting of the frequency of the high-speed on-chip oscillator
  - Select from 4 MHz, 8 MHz, 16 MHz, 24 MHz, and 32 MHz.

Caution Set the same value as 000C2H to 020C2H when the boot swap operation is used because 000C2H is replaced by 020C2H.

28.1.2 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.

28.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 28-1. Format of User Option Byte (000C0H/020C0H)

Address: 000C0H/020C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% +1/2f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19m s)

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 the WINDOW1 and WINDOW0 bits.

Caution The watchdog timer continues its operation during 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}: Low-speed on-chip oscillator clock frequency

Figure 28-2. Format of User Option Byte (000C1H/020C1H) (1/2)

Address: 000C1H/020C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	CLKMB	LVIS1	LVIS0	LVIMDS1	LVIMDS0

CLKMB	Clock monitoring operation control
0	Operates clock monitoring.
1	Stops clock monitoring. (default)

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte Setting Value						
V_{LVDH}		V_{LVDL}	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge	Falling edge							
2.92 V	2.86 V	2.75 V	1	0	0	1	1	1	0
3.02 V	2.96 V							0	1
4.06 V	3.98 V							0	0
Other than the above			Setting prohibited						

- LVD setting (reset mode)

Detection voltage		Option byte Setting Value						
V_{LVDH}		LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge							
2.81 V	2.75 V	1	1	0	1	1	1	1
2.92 V	2.86 V			0	1	1	1	0
3.02 V	2.96 V			0	1	1	0	1
3.13 V	3.06 V			0	0	1	0	0
3.75 V	3.67 V			0	1	0	0	0
4.06 V	3.98 V			0	1	1	0	0
Other than the above		Setting prohibited						

Note Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.

Remark For LVD setting, see **25.1 Functions of Voltage Detector**.

Figure 28-2. Format of User Option Byte (000C1H/020C1H) (2/2)

Address: 000C1H/020C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	CLKMB	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte Setting Value						
V_{LVDH}		LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge							
2.81 V	2.75 V	0	1	0	1	1	1	1
2.92 V	2.86 V			0	1	1	1	0
3.02 V	2.96 V			0	1	1	0	1
3.13 V	3.06 V			0	0	1	0	0
3.75 V	3.67 V			0	1	0	0	0
4.06 V	3.98 V			0	1	1	0	0
Other than the above		Setting prohibited						

- LVD setting (LVDOFF)

Detection voltage		Option byte Setting Value						
V_{LVD}		LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
Rising edge	Falling edge							
-	-	0/1	1	1	×	×	×	×
Other than the above		Setting prohibited						

Caution External $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

Note Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.

Remarks 1. ×: don't care

2. For LVD setting, see **25.1 Functions of Voltage Detector**.

Figure 28-3. Format of Option Byte (000C2H/020C2H)

Address: 000C2H/020C2H[Note]

7	6	5	4	3	2	1	0
CMODE1	CMODE0	OPTPLL	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode
1	1	HS (high speed main) mode
Other than the above		Setting prohibited

OPTPLL	PLL hard macro multiplication selection
0	× 16 selection (× 8 from user view) (If input clock is 4/8 MHz, f_{PLL} = 32 MHz)
1	× 12 selection (× 6 from user view) (default) (If input clock is 4/8 MHz, FPLL = 24 MHz)

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
1	0	1	0	8 MHz
1	0	1	1	4 MHz
Other than the above				Setting prohibited

Note Set the same value as 000C2H to 020C2H when the boot swap operation is used because 000C2H is replaced by 020C2H.

28.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 28-4. 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.

28.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler 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.

A software description example of the option byte setting is shown below.

```
OPT   CSEG        OPT_BYTE
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          7AH     ; Select 2.75 V for VLVDL
                          ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                          ; Select the interrupt & reset mode as the LVD operation mode
                            stops clock monitoring
      DB          0C9H    ; Select the HS (high speed main) mode as the flash operation mode
                            and 16 MHz as the frequency of the high-speed on-chip oscillator
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                            data when security ID authorization fails
```

When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 020C0H to 020C3H. Describe to 020C0H to 020C3H, therefore, the same values as 000C0H to 000C3H as follows.

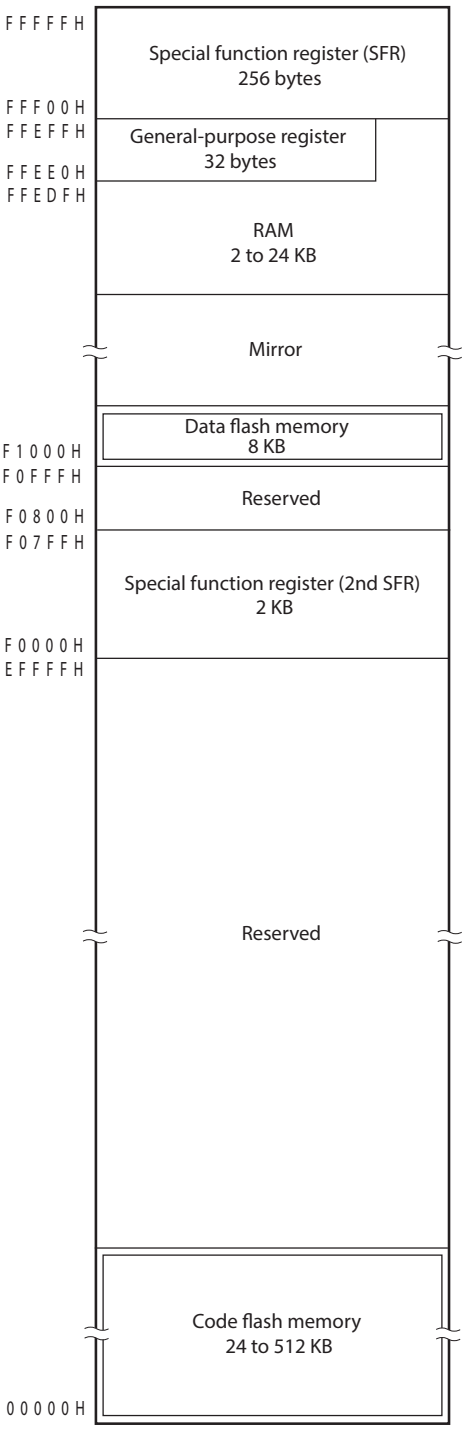
```
OPT2  CSEG   AT   020C0H
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          7AH     ; Select 2.75 V for VLVDL
                          ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                          ; Select the interrupt & reset mode as the LVD operation mode
                            stops clock monitoring
      DB          0C9H    ; Select the HS (high speed main) mode as the flash operation mode
                            and 16 MHz as the frequency of the high-speed on-chip oscillator
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                            data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 020C0H to 020C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 29 FLASH MEMORY

The RL78/D1A incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.

<R>



<R>

The following three methods for programming the flash memory are available:

- Writing to flash memory by using flash memory programmer (see **29.1**)
- Writing to flash memory by using external device (that Incorporates UART) (see **29.2**)
- Self-programming (see **29.7**)

29.1 Writing to Flash Memory by Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78/D1A.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78/D1A has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78/D1A is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 29-1. Wiring Between RL78/D1A and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.		
Signal Name		I/O	Pin Function		48-pin	64-pin	80-pin
PG-FP5, FL-PR5	E1 on-chip debugging emulator				LQFP (10x10)	LQFP (12x12)	LQFP (14x14)
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	5	5	9
SI/RxD	–	I/O	Transmit/receive signal				
SCK	–	Output	–	–	–	–	–
CLK	–	Output	–	–	–	–	–
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	6	8	12
/RESET	–	Output					
FLMD0	–	Output	Mode signal	–	–	–	–
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	12	16	20
				EV_{DD}	12	16	20
				SMV_{DD}	43	59	66, 76
GND		–	Ground	V_{SS}	11	15	19
				EV_{SS}	11	15	19
				SMV_{SS}	42	58	65, 75
				REGC Note	10	14	18
EMV_{DD}		–	Driving power for TOOL pin	EV_{DD}	12	16	20

<R>

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.	
Signal Name		I/O	Pin Function		100-pin	128-pin
PG-FP5, FL-PR5	E1 on-chip debugging emulator				LQFP (10x10)	LQFP (14x20)
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	11	114
SI/RxD	–	I/O	Transmit/receive signal			
SCK	–	Output	–	–	–	–
CLK	–	Output	–	–	–	–
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	15	118
/RESET	–	Output				
FLMD0	–	Output	Mode signal	–		–
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	24	127
				EV_{DD}	25, 33	8, 128
				SMV_{DD}	81, 91	74, 84
GND		–	Ground	V_{SS}	22	125
				EV_{SS}	23, 34	9, 126
				SMV_{SS}	80, 90	73, 83
				REGC Note	21	124
EMV_{DD}		–	Driving power for TOOL pin	EV_{DD}	25, 33	8, 128

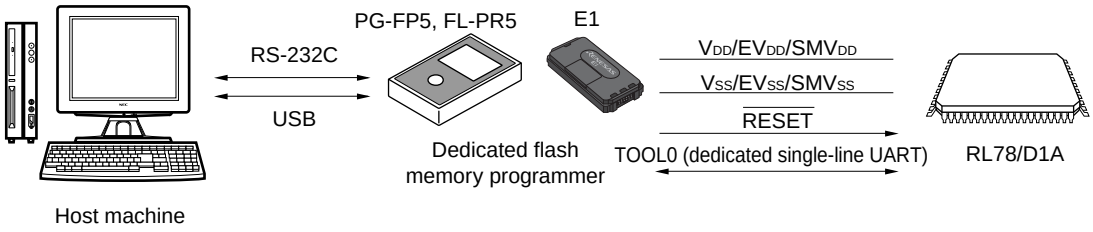
Note Connect REGC pin to ground via a capacitor (default: 0.47 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

29.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/D1A is illustrated below.

Figure 29-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

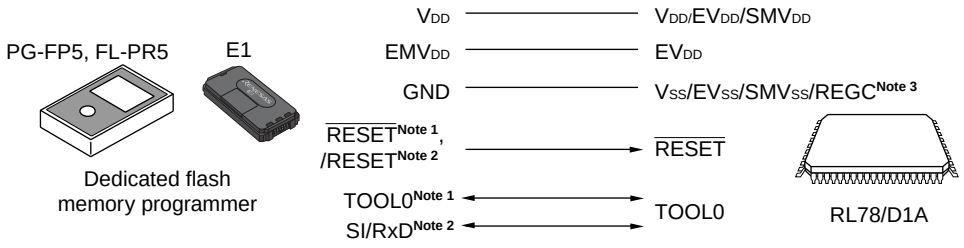
To interface between the dedicated flash memory programmer and the RL78/D1A, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

29.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78/D1A is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78/D1A.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 29-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. Connect REGC pin to ground via a capacitor (default: 0.47 μF).

The dedicated flash memory programmer generates the following signals for the RL78/D1A. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 29-2. Pin Connection

Dedicated Flash Memory Programmer				RL78/D1A	Connection
Signal Name		I/O	Pin Function	Pin Name	
PG-FP5, FL-PR5	E1 on-chip debugging emulator				
FLMD0	–	Output	Mode signal	–	×
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}, SMV_{DD}	◎
GND		–	Ground	V_{SS}, EV_{SS}, SMV_{SS}, REGC **Note**	◎
EMV_{DD}		–	Driving power for TOOL pin	EV_{DD}	◎
CLK	–	Output	Clock output	–	×
/RESET	–	Output	Reset signal	$\overline{RESET}$	◎
–	$\overline{RESET}$	Output			
–	TOOL0	I/O	Transmit/receive signal	TOOL0	◎
SI/RxD	–	I/O	Transmit/receive signal		
SCK	–	Output	Transfer clock	–	×

Note Connect REGC pin to ground via a capacitor (default: 0.47 μF).

Caution Make EV_{DD} the same potential as V_{DD}.

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

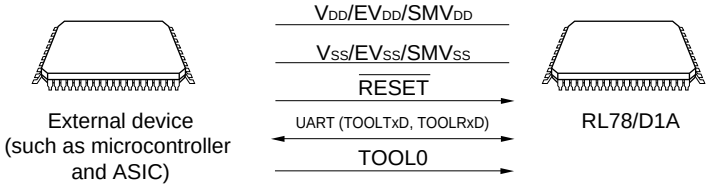
29.2 Writing to Flash Memory by Using External Device (that Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78/D1A and an external device (a microcontroller or ASIC) connected to a UART.

29.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/D1A is illustrated below.

Figure 29-3. Environment for Writing Program to Flash Memory



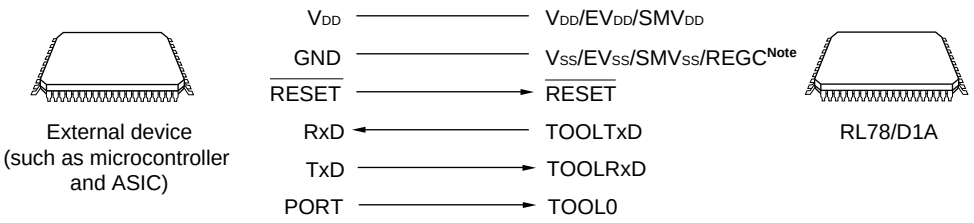
Processing to write data to or delete data from the RL78/D1A by using an external device is performed on-board. Off-board writing is not possible.

29.2.2 Communication Mode

Communication between the external device and the RL78/D1A is established by serial communication using the TOOLTxD and TOOLRxD pins via UART0 of the serial array unit of the RL78/D1A.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 29-4. Communication with External Device



Note Connect REGC pin to ground via a capacitor (0.47 μF).

Caution Make EV_{DD} the same potential as V_{DD}.

The external device generates the following signals for the RL78/D1A.

Table 29-3. Pin Connection

External Device			RL78/D1A	Connection
Signal Name	I/O	Pin Function	Pin Name	
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}, SMV_{DD}	◎
GND	–	Ground	V_{SS}, EV_{SS}, SMV_{SS}, REGC Note	◎
CLK	Output	Clock output	–	×
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$	◎
RxD	Input	Receive signal	TOOL0TxD	◎
TxD	Output	Transmit signal	TOOL0RxD	◎
PORT	Output	Mode signal	TOOL0	◎
SCK	Output	Transfer clock	–	×

Note Connect REGC pin to ground via a capacitor (0.47 μF).

Caution Make EV_{DD} the same potential as V_{DD}.

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

29.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

29.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor. When this pin is used as the port pin, use that by the following method. When used as an input pin: Input of 1 ms or more width low-level is prohibited after pin reset release. Furthermore, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

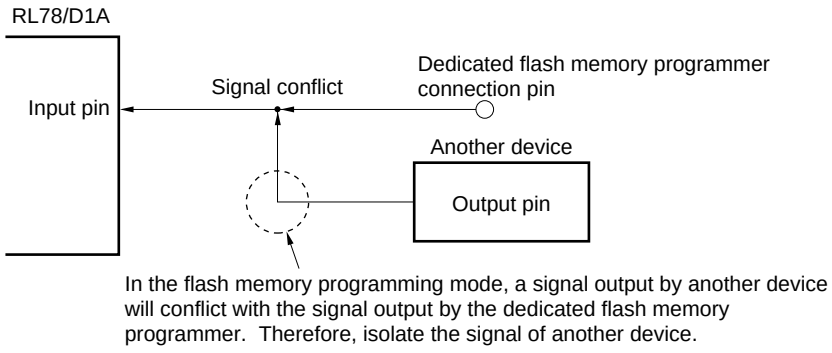
Remark The SAU and IICA pins are not used for communication between the RL78/D1A and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

29.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 29-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



29.3.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either to V_{DD}, or V_{SS}, via a resistor.

29.3.4 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

29.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

29.3.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

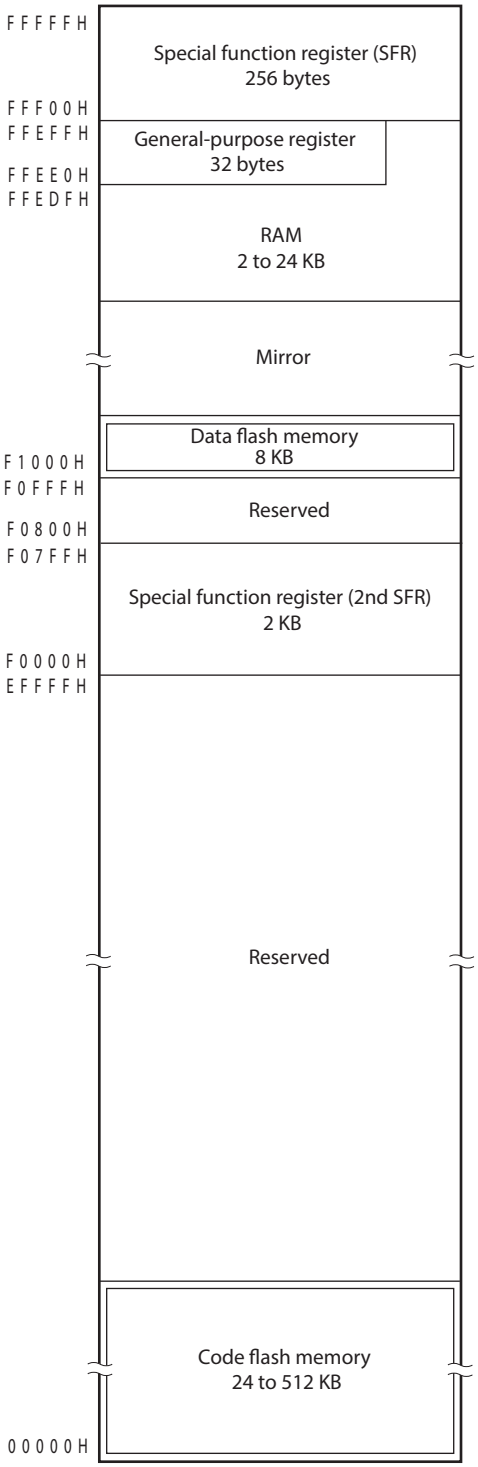
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

29.4 Data Flash

29.4.1 Data flash overview

In addition to 24 to 256 KB of code flash memory, the RL78/D1A includes 8 KB of data flash memory for storing data.

<R>



<R>

An overview of the data flash memory is provided below.

- The data flash memory can be written to by using the flash memory programmer or an external device
- Programming is performed in 8-bit units
- Blocks can be deleted in 1 KB units
- The only access by CPU instructions is byte reading (reading: four clock cycles)
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions (code fetching)
- Instructions can be executed from the code flash memory while rewriting the data flash memory (That is, dual operation is supported)
- Accessing the data flash memory is not possible while rewriting the code flash memory (such as during self-programming)
- Because the data flash memory is stopped after a reset ends, the data flash control register (DFLCTL) must be set up in order to use the data flash memory
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory
- Transition to the HALT/STOP status is not possible while rewriting the data flash memory
- Programming of data flash memory is possible while the program is being run by Renesas' library.

29.4.2 Register controlling data flash memory

(1) Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 29-6. Format of Data Flash Control Register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

29.4.3 Procedure for accessing data flash memory

The data flash memory is initially stopped after a reset ends and cannot be accessed (read or programmed). To access the memory, perform the following procedure:

<1> Write 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).
<2> Wait for the setup to finish.
The time setup takes differs for each main clock mode.
<Setup time for each main clock mode>
- HS (high-speed main) mode: 5 μs
- LS (low-speed main) mode: 720 ns

<3> After the wait, the data flash memory can be accessed.

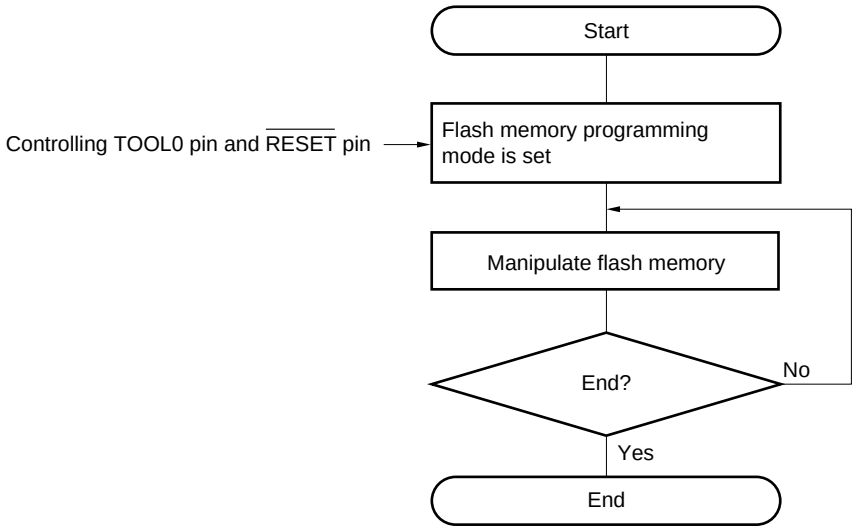
Cautions 1. Accessing the data flash memory is not possible during the setup time.
2. Before executing a STOP instruction during the setup time, temporarily clear DFLEN to 0.
3. Be sure to set the HIOSTOP bit in the CSC register to 0 when the CPU operates with the clock other than the high-speed on-chip oscillator clock.
4. The data flash should be read in either of following ways.
- **Use the flash library provided by Renesas (EEL (Pack01) version V1.13 or later).**
- **Stop the DMA transfer before reading.**

29.5 Programming Method

29.5.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 29-7. Flash Memory Manipulation Procedure



29.5.2 Flash memory programming mode

To rewrite the contents of the flash memory, set the RL78/D1A in the flash memory programming mode. To enter the mode, set as follows.

<When programming by using the dedicated flash memory programmer>

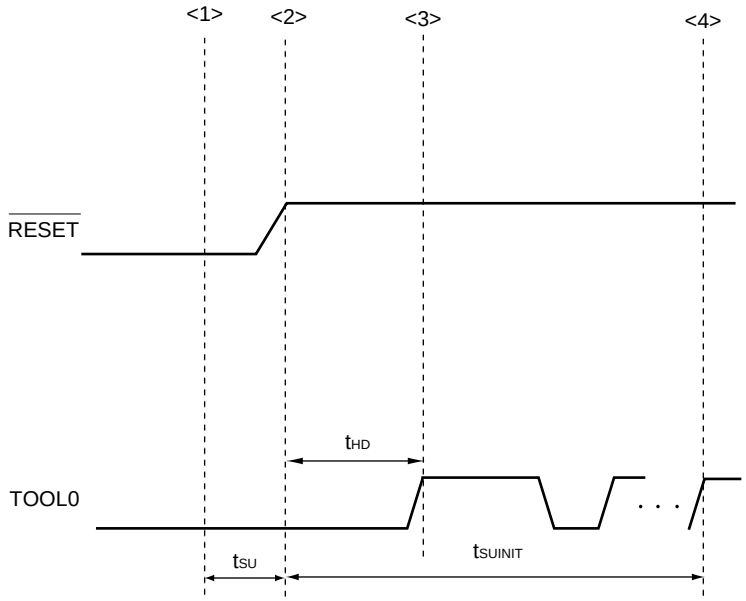
Set the TOOL0 pin to the low level, and then cancel the reset. Next, communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When programming by using an external device>

Set the TOOL0 pin to the low level, and then cancel the reset. Keep the TOOL0 pin at the low level for at least 1 ms after the reset ends, and then use UART communication to send the data “00H” from the external device. Complete UART communication within 100 ms after the reset ends.

When performing on-board writing, either switch the mode by using jumpers or perform pin processing in advance so that it will be okay if the flash memory programming mode is switched to (For details, see **29.3 Connection of Pins on Board**).

Figure 29-8. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The pins reset ends (POR and LVD reset must end before the pin reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external and internal resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends.
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end.

Table 29-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0	Flash memory programming mode

There are two flash memory programming modes for which the voltage range in which to write, erase, or verify data differs.

Table 29-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Mode	Voltages at which data can be written, erased, or verified	Writing Clock Frequency
Full speed mode **Note**	2.7 V to 5.5 V	32 MHz (MAX.)

Note This can only be specified if the CMODE0 bit is 1.

Specify the mode that corresponds to the voltage range in which to write data. When programming by using the dedicated flash memory programmer, the mode is automatically selected by the voltage setting on GUI.

Remark For details about communication commands, see **29.5.4 Communication commands**..

29.5.3 Selecting communication mode

Communication mode of the RL78/D1A as follows.

Table 29-6. Communication Modes

Communication Mode	Standard Setting **Note 1**				Pins Used
	Port	Speed **Note 2**	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0
UART0 (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOLTxD, TOOLRxD

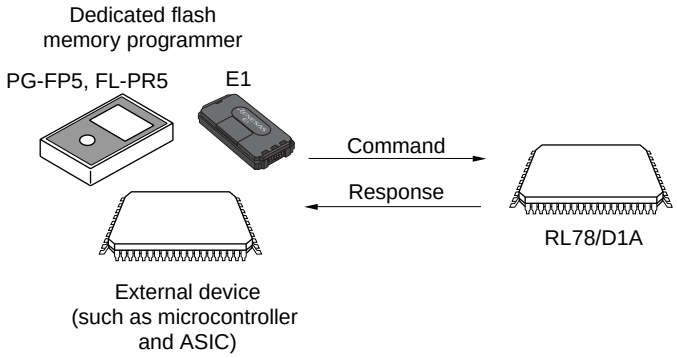
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

29.5.4 Communication commands

The RL78/D1A communicates with the dedicated flash memory programmer or external device by using commands. The signals sent from the flash memory programmer or external device to the RL78/D1A are called commands, and the signals sent from the RL78/D1A to the dedicated flash memory programmer or external device are called response.

Figure 29-9. Communication Commands



The flash memory control commands of the RL78/D1A are listed in the table below. All these commands are issued from the programmer or external device, and the RL78/D1A perform processing corresponding to the respective commands.

Table 29-7. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets the RL78/D1A information (such as the part number and flash memory configuration).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

The RL78/D1A returns a response for the command issued by the dedicated flash memory programmer or external device. The response names sent from the RL78/D1A are listed below.

Table 29-8. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

29.5.5 Description of signature data

When the “silicon signature” command is performed, the RL78/D1A information (such as the part number, flash memory configuration, and programming firmware version) can be obtained.

Table 29-9 and 29-10 show signature data list and example of signature data list.

Table 29-9. Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 0FFFFH (64 KB) → FFH, 1FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F1FFFH (4 KB) → FFH, 1FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 28-10. Example of Signature Data

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F10DPJ	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 30 = “0” 44 = “D” 50 = “P” 4A = “J” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 3FFFFH (256 KB)	3 bytes	FF FF 03
Data flash memory area last address	Data flash memory area F1000H to F2FFFH (8 KB)	3 bytes	FF 2F 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

29.6 Security Settings

The RL78/D1A supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self-programming.

- Disabling write
  Execution of the write command for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self-programming.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The block erase, write commands and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self-programming. Each security setting can be used in combination.

Table 29-11 shows the relationship between the erase and write commands when the RL78/D1A security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **29.7.2** for detail).

Table 29-11. Relationship Between Enabling Security Function and Command

(1) During on-board/off-board programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed. Note
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **29.7.2** for detail).

Table 29-12. Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Execute security release command
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution The security release command can be applied only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

(2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Execute security release command during on-board/off-board programming (cannot be disabled during self-programming)
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

29.7 Flash Memory Programming by Self-Programming

The RL78/D1A supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the RL78/D1A self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock. Be sure to set the HIOSTOP bit in the CSC register to 0 when using the self-programming function if the CPU operates with the main system clock.

2. 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.

3. Do not transit to standby mode during self-programming.

Remarks 1. For details of the self-programming function and the RL78/D1A self-programming library, refer to **RL78 Microcontroller Self Programming Library Type01 User's Manual.**

2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

Similar to when writing data by using the flash memory programmer, there are two flash memory programming modes for which the voltage range in which to write, erase, or verify data differs.

Table 28-13. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Mode	Voltages at which data can be written, erased, or verified	Writing Clock Frequency
Full speed mode Note	2.7 V to 5.5 V	32 MHz (MAX.)

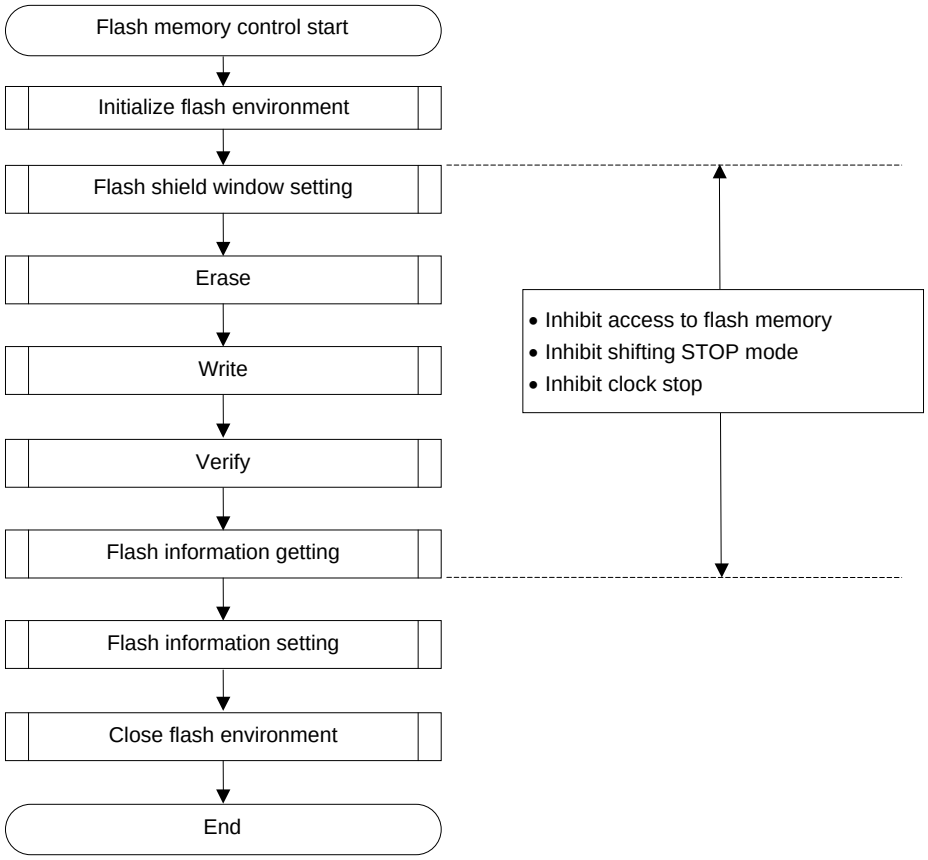
Note This can only be specified if the CMODE0 bits bit is 1.

The argument fsl_flash_voltage_u08 must be set to 00H when the FSL_Init function of the self-programming library provided by Renesas Electronics.

Remark For details of the self-programming function and the RL78/D1A self-programming library, refer to **RL78 Microcontroller Self Programming Library Type01 User's Manual.**

The following figure illustrates a flow of rewriting the flash memory by using a self-programming library.

Figure 29-10. Flow of Self Programming (Rewriting Flash Memory)



29.7.1 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.

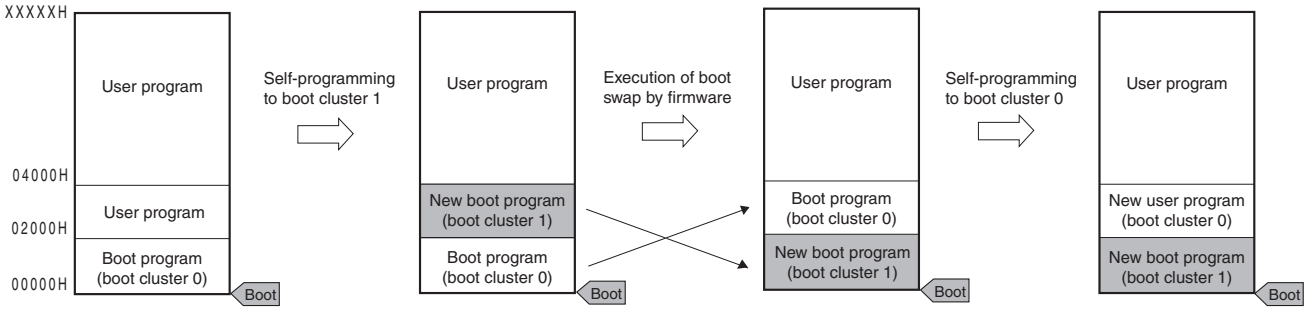
The boot swap function is used to avoid this problem.

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 RL78/D1A, 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 29-11. Boot Swap Function

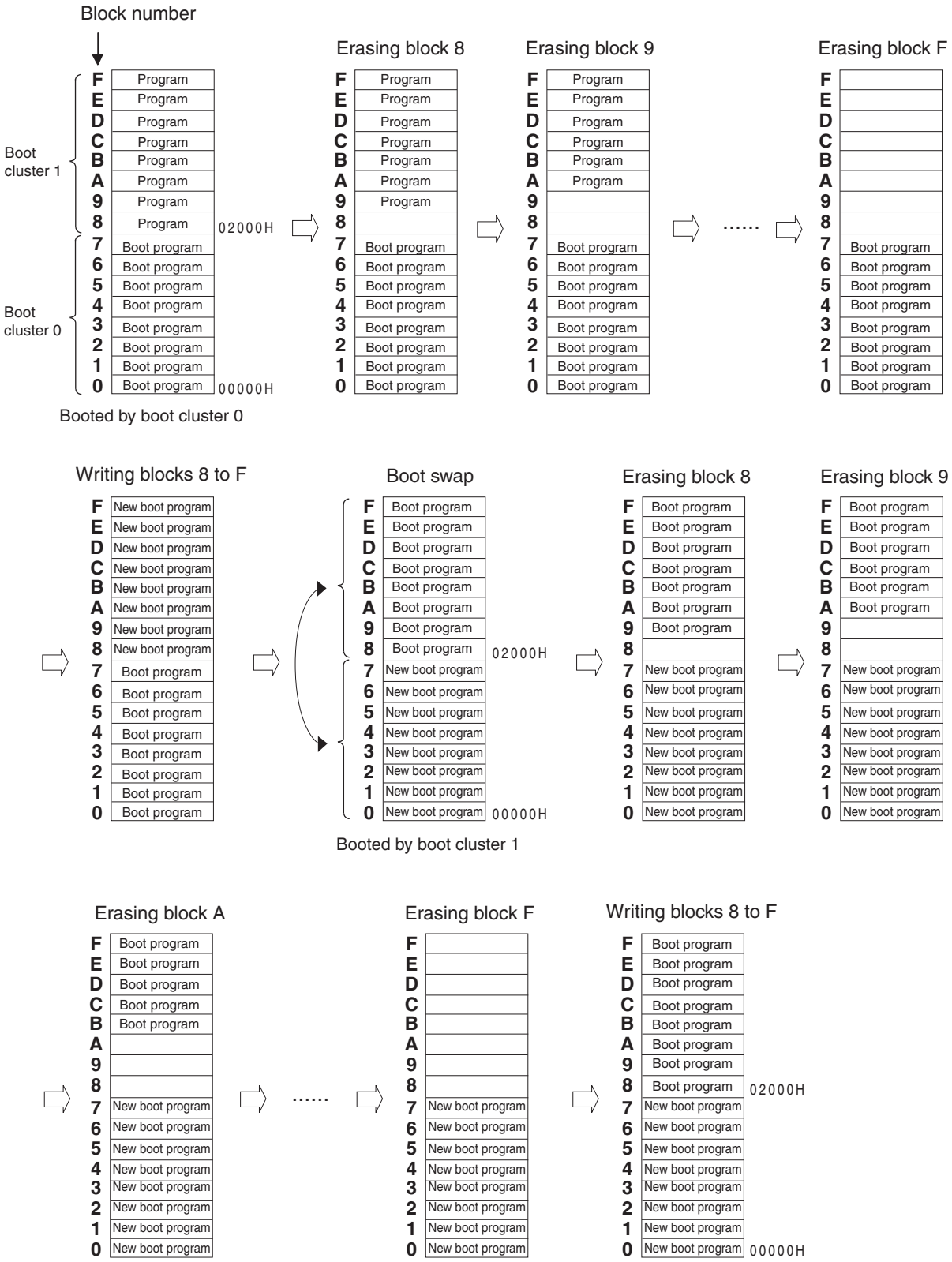


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 29-12. Example of Executing Boot Swapping



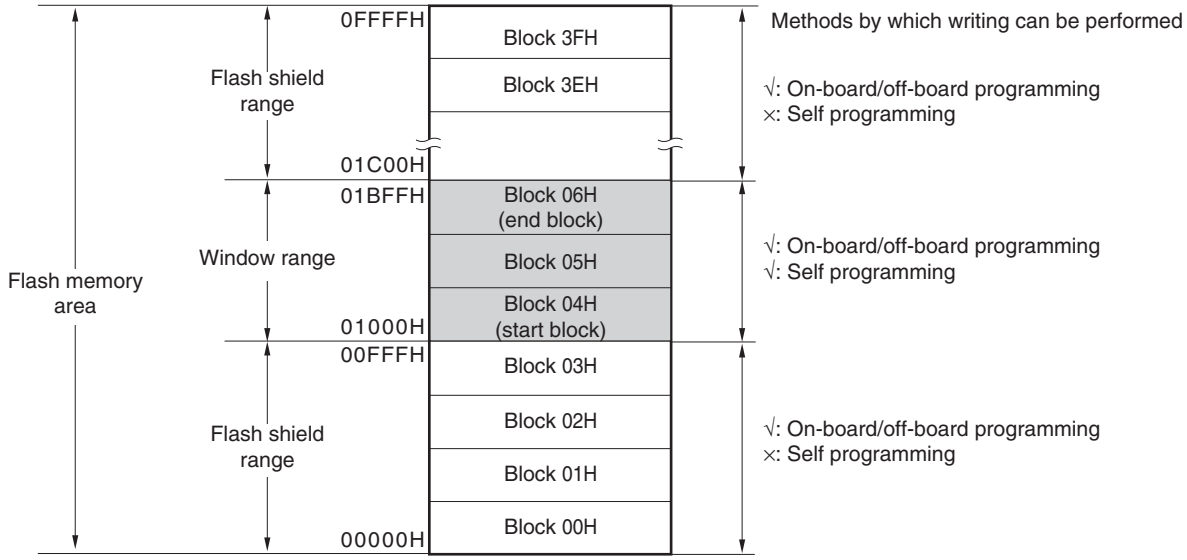
29.7.2 Flash shield window function

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 29-13. Flash Shield Window Setting Example (Start Block: 04H, End Block: 06H)



Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 29-14. 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 **29.6 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

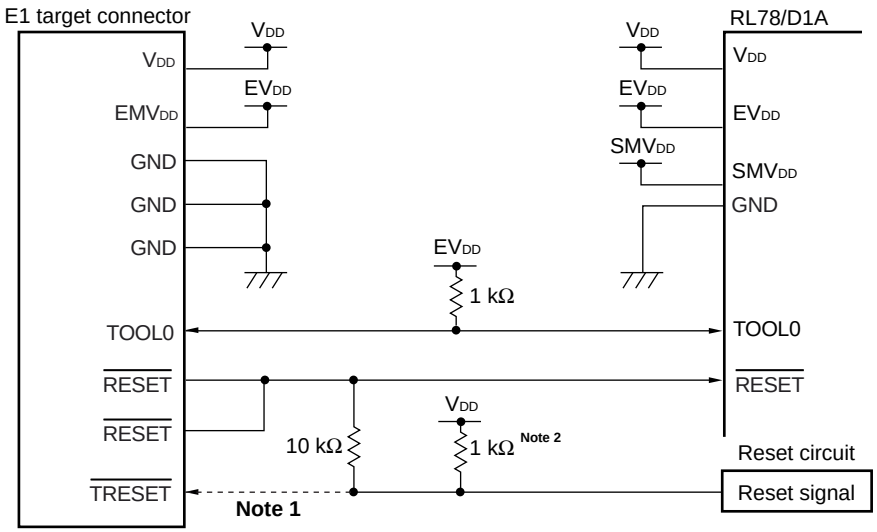
CHAPTER 30 ON-CHIP DEBUG FUNCTION

30.1 Connecting E1 On-chip Debugging Emulator to RL78/D1A

The RL78/D1A uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78/D1A 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 30-1. Connection Example of E1 On-chip Debugging Emulator and RL78/D1A



Notes 1. Connecting the dotted line is not necessary during flash programming.

2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

30.2 On-Chip Debug Security ID

The RL78/D1A has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 28 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.

Table 30-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
020C4H to 020CDH	

30.3 Securing of User Resources

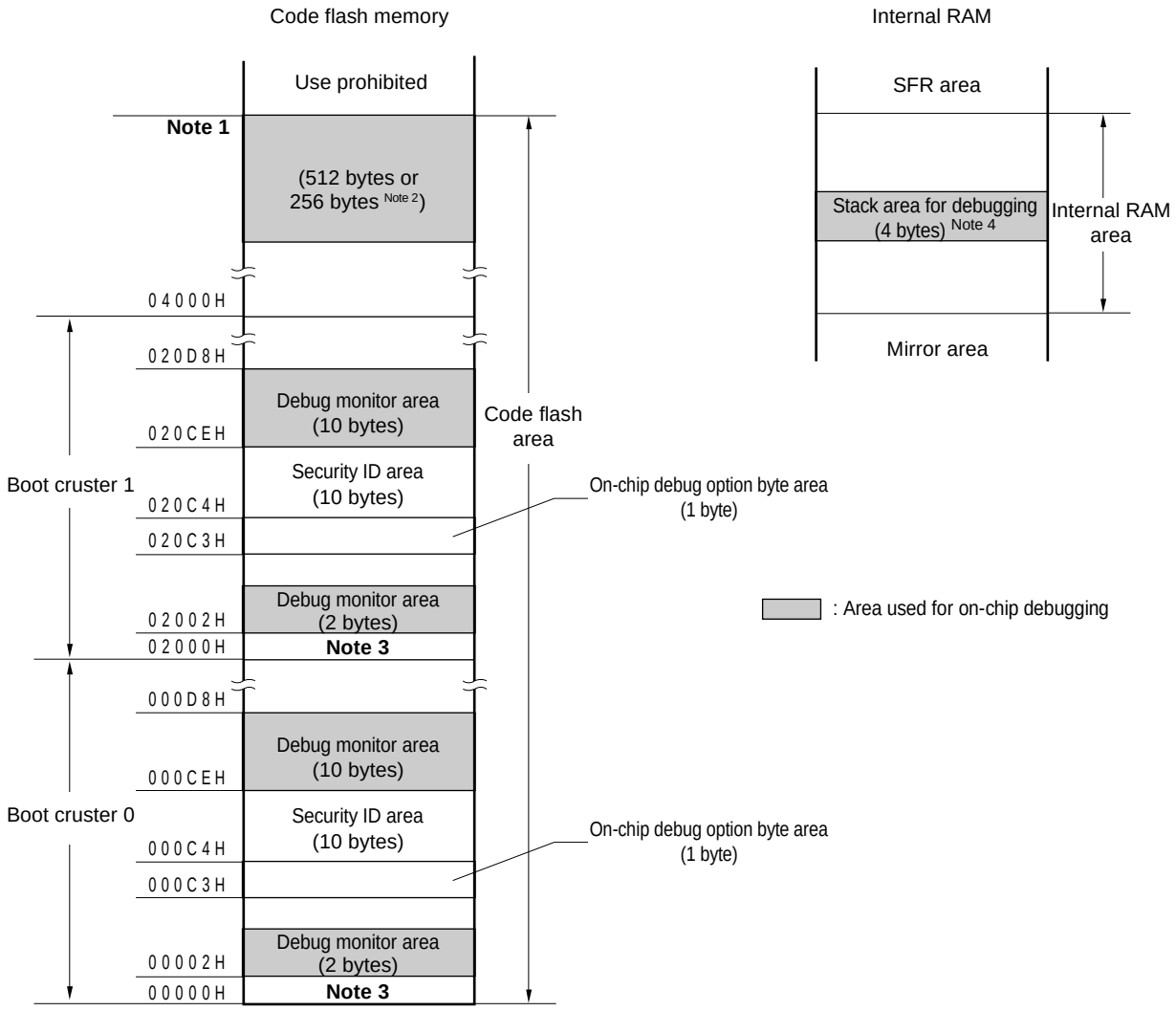
To perform communication between the RL78/D1A and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 30-2 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 30-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1**
R5F10CGB	05C00H to 05FFFH
R5F10CGC, R5F10DGC	07C00H to 07FFFH
R5F10CxD, R5F10DxD (x = G, L, M)	0BC00H to 0BFFFH
R5F10CME, R5F10DxE (x = G, L, M, P)	0FC00H to 0FFFFH
R5F10DxF (x = M, P)	17C00H to 17FFFH
R5F10DxG (x = M, P)	1FC00H to 1FFFFH
R5F10DxJ, R5F10TPJ (x = M, P)	3FC00H to 3FFFFH
<R> R5F10DPK	5FC00H to 5FFFFH
<R> R5F10DPL, R5F10DSx (x = L, K, J)	7FC00H to 7FFFFH

2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

3. In debugging, reset vector is rewritten to address allocated to a monitor program.

4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 31 BCD CORRECTION CIRCUIT

31.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

31.2 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.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 31-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

31.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

Examples are shown below.

Example 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	–

Example 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	–

Example 3: 80 + 80 = 160

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 – 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	–	–	–
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	–

CHAPTER 32 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual: Software** (R01US0015E).

Remark The shaded parts of the tables in **Table 32-5 Operation List** indicate the operation or instruction format that is newly added for the RL78 microcontrollers.

32.1 Conventions Used in Operation List

32.1.1 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 32-1. 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.

32.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 32-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∀	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

32.1.3 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 32-3. 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

32.1.4 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 32-4. 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.

32.2 Operation List

Table 32-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte	
		saddr, #byte	3	1	–	(saddr) ← byte	
		sfr, #byte	3	1	–	sfr ← byte	
		!addr16, #byte	4	1	–	(addr16) ← byte	
		A, r Note 3	1	1	–	A ← r	
		r, A Note 3	1	1	–	r ← A	
		A, saddr	2	1	–	A ← (saddr)	
		saddr, A	2	1	–	(saddr) ← A	
		A, sfr	2	1	–	A ← sfr	
		sfr, A	2	1	–	sfr ← A	
		A, !addr16	3	1	4	A ← (addr16)	
		!addr16, A	3	1	–	(addr16) ← A	
		PSW, #byte	3	3	–	PSW ← byte	× × ×
		A, PSW	2	1	–	A ← PSW	
		PSW, A	2	3	–	PSW ← A	× × ×
		ES, #byte	2	1	–	ES ← byte	
		ES, saddr	3	1	–	ES ← (saddr)	
		A, ES	2	1	–	A ← ES	
		ES, A	2	1	–	ES ← A	
		CS, #byte	3	1	–	CS ← byte	
		A, CS	2	1	–	A ← CS	
		CS, A	2	1	–	CS ← A	
		A, [DE]	1	1	4	A ← (DE)	
		[DE], A	1	1	–	(DE) ← A	
		[DE + byte], #byte	3	1	–	(DE + byte) ← byte	
		A, [DE + byte]	2	1	4	A ← (DE + byte)	
		[DE + byte], A	2	1	–	(DE + byte) ← A	
		A, [HL]	1	1	4	A ← (HL)	
		[HL], A	1	1	–	(HL) ← A	
		[HL + byte], #byte	3	1	–	(HL + byte) ← byte	

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, [HL + byte]	2	1	4	A ← (HL + byte)	
		[HL + byte], A	2	1	–	(HL + byte) ← A	
		A, [HL + B]	2	1	4	A ← (HL + B)	
		[HL + B], A	2	1	–	(HL + B) ← A	
		A, [HL + C]	2	1	4	A ← (HL + C)	
		[HL + C], A	2	1	–	(HL + C) ← A	
		word[B], #byte	4	1	–	(B + word) ← byte	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		word[C], #byte	4	1	–	(C + word) ← byte	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		word[BC], #byte	4	1	–	(BC + word) ← byte	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		[SP + byte], #byte	3	1	–	(SP + byte) ← byte	
		A, [SP + byte]	2	1	–	A ← (SP + byte)	
		[SP + byte], A	2	1	–	(SP + byte) ← A	
		B, saddr	2	1	–	B ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		X, !addr16	3	1	4	X ← (addr16)	
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte	
		A, ES:!addr16	4	2	5	A ← (ES, addr16)	
		ES:!addr16, A	4	2	–	(ES, addr16) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		ES:[DE + byte],#byte	4	2	–	((ES, DE) + byte) ← byte	
		A, ES:[DE + byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE + byte], A	3	2	–	((ES, DE) + byte) ← A	

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		ES:[HL + byte],#byte	4	2	–	((ES, HL) + byte) ← byte	
		A, ES:[HL + byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL + byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, ES:[HL + B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL + B], A	3	2	–	((ES, HL) + B) ← A	
		A, ES:[HL + C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL + C], A	3	2	–	((ES, HL) + C) ← A	
		ES:word[B], #byte	5	2	–	((ES, B) + word) ← byte	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		ES:word[C], #byte	5	2	–	((ES, C) + word) ← byte	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		ES:word[BC], #byte	5	2	–	((ES, BC) + word) ← byte	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, [DE + byte]	3	2	–	A ←→ (DE + byte)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, [HL + byte]	3	2	–	A ←→ (HL + byte)	
		A, [HL + B]	2	2	–	A ←→ (HL + B)	
		A, [HL + C]	2	2	–	A ←→ (HL + C)	

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	XCH	A, ES:!addr16	5	3	–	A ←→ (ES, addr16)	
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)	
		A, ES:[DE + byte]	4	3	–	A ←→ ((ES, DE) + byte)	
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)	
		A, ES:[HL + byte]	4	3	–	A ←→ ((ES, HL) + byte)	
		A, ES:[HL + B]	3	3	–	A ←→ ((ES, HL) + B)	
		A, ES:[HL + C]	3	3	–	A ←→ ((ES, HL) + C)	
	ONEB	A	1	1	–	A ← 01H	
		X	1	1	–	X ← 01H	
		B	1	1	–	B ← 01H	
		C	1	1	–	C ← 01H	
		saddr	2	1	–	(saddr) ← 01H	
		!addr16	3	1	–	(addr16) ← 01H	
		ES:!addr16	4	2	–	(ES, addr16) ← 01H	
	CLRB	A	1	1	–	A ← 00H	
		X	1	1	–	X ← 00H	
		B	1	1	–	B ← 00H	
		C	1	1	–	C ← 00H	
		saddr	2	1	–	(saddr) ← 00H	
		!addr16	3	1	–	(addr16) ← 00H	
		ES:!addr16	4	2	–	(ES,addr16) ← 00H	
	MOVS	[HL + byte], X	3	1	–	(HL + byte) ← X	× ×
		ES:[HL + byte], X	4	2	–	(ES, HL + byte) ← X	× ×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word	
		saddrp, #word	4	1	–	(saddrp) ← word	
		sfrp, #word	4	1	–	sfrp ← word	
		AX, saddrp	2	1	–	AX ← (saddrp)	
		saddrp, AX	2	1	–	(saddrp) ← AX	
		AX, sfrp	2	1	–	AX ← sfrp	
		sfrp, AX	2	1	–	sfrp ← AX	
		AX, rp Note 3	1	1	–	AX ← rp	
		rp, AX Note 3	1	1	–	rp ← AX	

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	MOVW	AX, !addr16	3	1	4	AX ← (addr16)	
		!addr16, AX	3	1	–	(addr16) ← AX	
		AX, [DE]	1	1	4	AX ← (DE)	
		[DE], AX	1	1	–	(DE) ← AX	
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)	
		[DE + byte], AX	2	1	–	(DE + byte) ← AX	
		AX, [HL]	1	1	4	AX ← (HL)	
		[HL], AX	1	1	–	(HL) ← AX	
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)	
		[HL + byte], AX	2	1	–	(HL + byte) ← AX	
		AX, word[B]	3	1	4	AX ← (B + word)	
		word[B], AX	3	1	–	(B + word) ← AX	
		AX, word[C]	3	1	4	AX ← (C + word)	
		word[C], AX	3	1	–	(C + word) ← AX	
		AX, word[BC]	3	1	4	AX ← (BC + word)	
		word[BC], AX	3	1	–	(BC + word) ← AX	
		AX, [SP + byte]	2	1	–	AX ← (SP + byte)	
		[SP + byte], AX	2	1	–	(SP + byte) ← AX	
		BC, saddrp	2	1	–	BC ← (saddrp)	
		BC, !addr16	3	1	4	BC ← (addr16)	
		DE, saddrp	2	1	–	DE ← (saddrp)	
		DE, !addr16	3	1	4	DE ← (addr16)	
		HL, saddrp	2	1	–	HL ← (saddrp)	
		HL, !addr16	3	1	4	HL ← (addr16)	
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)	
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX	
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)	
		ES:[DE], AX	2	2	–	(ES, DE) ← AX	
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)	
		ES:[DE + byte], AX	3	2	–	((ES, DE) + byte) ← AX	
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)	
		ES:[HL], AX	2	2	–	(ES, HL) ← AX	

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. When the internal RAM area, 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. Except rp = AX

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C)	×	×	×

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B) – CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	r ← r ∧ A	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	1	–	A ← A ∨ byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	1	–	A ← A ∨ r	×
		r, A	2	1	–	r ← r ∨ A	×
		A, saddr	2	1	–	A ← A ∨ (saddr)	×
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×
		A, [HL]	1	1	4	A ← A ∨ (HL)	×
		A, [HL + byte]	2	1	4	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	1	4	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	1	4	A ← A ∨ (HL + C)	×
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×
		A, ES:[HL + byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×
		A, ES:[HL + B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×
		A, ES:[HL + C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×
	XOR	A, #byte	2	1	–	A ← A ⩛ byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr) ⩛ byte	×
		A, r Note 3	2	1	–	A ← A ⩛ r	×
		r, A	2	1	–	r ← r ⩛ A	×
		A, saddr	2	1	–	A ← A ⩛ (saddr)	×
		A, !addr16	3	1	4	A ← A ⩛ (addr16)	×
		A, [HL]	1	1	4	A ← A ⩛ (HL)	×
		A, [HL + byte]	2	1	4	A ← A ⩛ (HL + byte)	×
		A, [HL + B]	2	1	4	A ← A ⩛ (HL + B)	×
		A, [HL + C]	2	1	4	A ← A ⩛ (HL + C)	×
		A, ES:!addr16	4	2	5	A ← A ⩛ (ES:addr16)	×
		A, ES:[HL]	2	2	5	A ← A ⩛ (ES:HL)	×
		A, ES:[HL + byte]	3	2	5	A ← A ⩛ ((ES:HL) + byte)	×
		A, ES:[HL + B]	3	2	5	A ← A ⩛ ((ES:HL) + B)	×
		A, ES:[HL + C]	3	2	5	A ← A ⩛ ((ES:HL) + C)	×

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A – (HL + C)	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
	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	(ES:addr16) – 00H	×	×	×
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	×	×	×

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX + HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX + (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A × X			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r + 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) + 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) + 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp + 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) + 1			
		!addr16	3	2	–	(addr16) ← (addr16) + 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	–	rp ← rp – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. 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 32-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Rotate	ROR	A, 1	2	1	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1	×
	ROL	A, 1	2	1	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1	×
	RORC	A, 1	2	1	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1	×
	ROLC	A, 1	2	1	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1	×
	ROLWC	AX,1	2	1	–	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1	×
		BC,1	2	1	–	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 1	×
Bit manipulate	MOV1	CY, saddr.bit	3	1	–	CY ← (saddr).bit	×
		CY, sfr.bit	3	1	–	CY ← sfr.bit	×
		CY, A.bit	2	1	–	CY ← A.bit	×
		CY, PSW.bit	3	1	–	CY ← PSW.bit	×
		CY,[HL].bit	2	1	4	CY ← (HL).bit	×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY	
		sfr.bit, CY	3	2	–	sfr.bit ← CY	
		A.bit, CY	2	1	–	A.bit ← CY	
		PSW.bit, CY	3	4	–	PSW.bit ← CY	× ×
		[HL].bit, CY	2	2	–	(HL).bit ← CY	
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit	×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY	
	AND1	CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit	×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	1	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit	×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit	×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit	×
	OR1	CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit	×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	1	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit	×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, saddr.bit	3	1	–	CY ← CY ⩛ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⩛ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ⩛ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⩛ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⩛ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⩛ (ES, HL).bit			×
	SET1	saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		A.bit	2	1	–	A.bit ← 1			
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		A.bit	2	1	–	A.bit ← 0			
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{CY}$			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP – 4	
		$!addr20	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← PC + 3 + jdisp16, SP ← SP – 4	
		!addr16	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP – 4	
		!!addr20	4	3	–	(SP – 2) ← (PC + 4)S, (SP – 3) ← (PC + 4)H, (SP – 4) ← (PC + 4)L, PC ← addr20, SP ← SP – 4	
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L , PCS ← 0000, PCH ← (0000, addr5 + 1), PCL ← (0000, addr5), SP ← SP – 4	
	BRK	–	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0	
	RET	–	1	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), SP ← SP + 4	
	RETI	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R R R
	RETB	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R R R

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 32-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP – 2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2)← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area, 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. 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 32-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0	
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0	
	BTCLR	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit	
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
		ES:[HL].bit, $addr20	4	4/6[Note 3]	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit	
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1	
	SKNC	–	2	1	–	Next instruction skip if CY = 0	
	SKZ	–	2	1	–	Next instruction skip if Z = 1	
	SKNZ	–	2	1	–	Next instruction skip if Z = 0	
	SKH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 0	
	SKNH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 1	
CPU control	SEL	RBn	2	1	–	RBS[1:0] ← n	
	NOP	–	1	1	–	No Operation	
	EI	–	3	4	–	IE ← 1(Enable Interrupt)	
	DI	–	3	4	–	IE ← 0(Disable Interrupt)	
	HALT	–	2	3	–	Set HALT Mode	
	STOP	–	2	3	–	Set STOP Mode	

Notes 1. When the internal RAM area, 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. 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 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT)

Cautions 1. These specifications show target values, which may change after device evaluation.

2. The RL78/D1A 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.

Definition of Pin Groups

Definition of pin groups described in this chapter is shown in the following table.

<R>

Pin groups		Pin names				
		48-pin products	64-pin products	80-pin products	100-pin products	128-pin products
Pin group 1	Group 1R	P40	P40, P70, P71	P40, P70, P71	P40, P70, P71, P130 to P135, P140	P40 to P44, P70, P71, P100 to P103, P130 to P135, P140
	Group 1L	P00 to P01, P10 to P14, P30 to P33, P54 to P57, P60, P61, P72 to P75	P00 to P05, P07, P10 to P15, P17, P30 to P33, P54 to P57, P60, P61, P72 to P75	P00 to P07, P10 to P17, P30 to P37, P54 to P57,P60, P61, P65, P66, P72 to P75	P00 to P07, P10 to P17, P30 to P37, P50 to P57, P72 to P75, P136	P00 to P07, P10 to P17, P30 to P37, P45 to P47, P50 to P57, P72 to P75, P104 to P107, P110 to P117, P125 to P127 P136
	Group 1C	-	-	-	P60 to P66	P60 to P66
Pin group 2 (ANI pins)		P20 to P23, P27	P20 to P23, P27	P20 to P27	P20 to P27, P150	P20 to P27, P150 to P152
Pin group 3 (SMC pins)	Group 3A	P80 to P83	P80 to P83	P80 to P83	P80 to P83	P80 to P83
	Group 3B	-	P84 to P87	P84 to P87	P84 to P87	P84 to P87
	Group 3C	P90 to P94	P90 to P94	P90 to P93	P90 to P93	P90 to P93
	Group 3D	-	-	P94 to P97	P94 to P97	P94 to P97
	Group 3E	P90 to P94	P84 to P87, P90 to P94	-	-	-
<R> Pin group 4 (System pins)		P121 to P122, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137

Definition of Product Groups

Definition of product groups described in this chapter is shown in the following table.

<R>

Product groups	Product names				
	48-pin products	64-pin products	80-pin products	100-pin products	128-pin products
Product Group A	R5F10CGBJFB R5F10CGCJFB R5F10CGDJFB R5F10DGCJFB R5F10DGDJFB R5F10DGEJFB	R5F10CLDJFB R5F10DLDJFB R5F10DLEJFB	R5F10CMDJFB R5F10CMEJFB R5F10DMDJFB R5F10DMEJFB R5F10DMFJFB R5F10DMGJFB R5F10DMJJFB	R5F10DPEJFB R5F10DPFJFB R5F10DPGJFB R5F10DPJJFB R5F10TPJJFB	-
Product Group B	-	-	-	R5F10DPKJFB R5F10DPLJFB	R5F10DSJJFB R5F10DSKJFB R5F10DSLJFB

33.1 Absolute Maximum Ratings

T_A = +25 °C

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}	V_{DD}	-0.5 to +6.5	V
	EV_{DD0} EV_{DD1}	$EV_{DD0} = EV_{DD1}$	-0.5 to +6.5 and -0.5 to V_{DD} + 0.3	V
	SMV_{DD0} SMV_{DD1}	$SMV_{DD0} = SMV_{DD1}$	-0.5 to +6.5 and -0.5 to V_{DD} + 0.3	V
	V_{SS}	V_{SS}	-0.5 to +0.3	V
	EV_{SS0} EV_{SS1}	$EV_{SS0} = EV_{SS1}$	-0.5 to +0.3	V
	SMV_{SS0} SMV_{SS1}	$SMV_{SS0} = SMV_{SS1}$	-0.5 to +0.3	V
Supply voltage up/down ramp	V_{DDRAMP}		≤50	V/ms
REGC pin input voltage	V_{IREGC}	REGC	-0.3 to +2.8 and -0.3 to V_{DD} + 0.3 Note1	V
Input voltage	V_{I1}	Pin group 1	-0.3 to +6.5 and -0.3 to EV_{DD0}(EV_{DD1}) + 0.3	V
	V_{I2}	Pin group 2	-0.3 to +6.5 and -0.3 to V_{DD} + 0.3	V
	V_{I3}	Pin group 3	-0.3 to +6.5 and -0.3 to SMV_{DD0}(SMV_{DD1}) + 0.3	V
	V_{I4}	Pin group 4	-0.3 to +6.5 and -0.3 to V_{DD} + 0.3	V

(Continue to next page)

T_A = +25 °C

Parameter	Symbols	Conditions			Ratings	Unit
Output voltage	V_{O1}	Pin group 1			-0.3 to EV_{DD0} (EV_{DD1}) + 0.3	V
	V_{O2}	Pin group 2			-0.3 to V_{DD} + 0.3	V
	V_{O3}	Pin group 3			-0.3 to SMV_{DD0} (SMV_{DD1}) + 0.3	V
	V_{COM}	COM0 to COM3			-0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH1}	Per pin	Pin group 1		-20	mA
		Total	Pin group 1		-150	mA
				Pin group 1L	-60	mA
				Pin group 1R	-55	mA
				Pin group 1C	-40	mA
	I_{OH2}	Per pin	Pin group 2		-0.5	mA
		Total			-2.0	mA
	I_{OH3}	Per pin	Pin group 3		-58	mA
		Total	Pin group 3	48-pin, 64-pin	-270	mA
<R>				80-pin, 100-pin, 128-pin	-480	mA
				Pin group 3A	-120	mA
				Pin group 3B	-120	mA
				Pin group 3C	-120	mA
				Pin group 3D	-120	mA
				Pin group 3E	-150	mA
	I_{OHCOM}	Per pin	COM0 to COM3		-0.5	mA
		Total	COM0 to COM3		-1.0	mA
Output current, low	I_{OL1}	Per pin	Pin group 1		20	mA
		Total	Pin group 1		150	mA
				Pin group 1L	60	mA
				Pin group 1R	50	mA
				Pin group 1C	40	mA
	I_{OL2}	Per pin	Pin group 2		1.0	mA
		Total			5.0	mA
	I_{OL3}	Per pin	Pin group 3		58	mA
		Total	Pin group 3	48-pin, 64-pin	270	mA
<R>				80-pin, 100-pin, 128-pin	480	mA
				Pin group 3A	120	mA
				Pin group 3B	120	mA
				Pin group 3C	120	mA
				Pin group 3D	120	mA
				Pin group 3E	150	mA
	I_{OLCOM}	Per pin	COM0 to COM3		0.5	mA
		Total	COM0 to COM3		1.0	mA
Operating ambient temperature	T_A	for normal operation mode			-40 to +85	°C
		for code flash programming			-40 to +85	°C
		for data flash programming			-40 to +85	°C
Storage temperature	T_{stg}				-65 to +150	°C

Note 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.

<R> 33.2 Power consumption characteristics

33.2.1 Product group A

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, run mode	I_{DD1} Note 1	High speed MAIN RUN Note 2, 3, 4	f_{CLK} = 32 MHz	f_{HOCO} = 32 MHz	5.2	22	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	4.2	18	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	3.8	16	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	2.1	11	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	1.6	9	mA
				f_X = 4 MHz			
		SUB RUN Note 2, 3, 5	f_{CLK} = f_{XT} = 32.768 kHz		6	300	μA

Notes 1. The common condition for I_{DD1}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The program is running in the code flash.

2. The typical value is that when Ta=+25deg.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. But the A/D converter, RTC, LCD circuit and stepper motor circuit are stopped, and the data flash and the code flash are stated to read mode. The 16-bit wakeup timer operates with f_{LOCO}.

4. Either f_X or f_{HOCO} which is selected for f_{CLK} is operated. The other is stopped.

5. f_X and f_{HOCO} are stopped.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, halt mode	I_{DD2} Note 1	High speed MAIN HALT Note 3, 4, 5	f_{CLK} = 32 MHz 2.7 V≤V_{DD}	f_{HOCO} = 32 MHz	1.0	8.1	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	0.8	6.9	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	0.7	6.0	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	0.4	4.3	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	0.35	3.6	mA
				f_X = 4 MHz			
		SUB HALT Note 3, 4, 6	f_{CLK} = f_{XT} = 32.768 kHz	RTC is stopped	1.0	130	μA
				RTC is operated by f_{XT} = 32.768KHz	1.2		
Supply current, stop mode	I_{DD3} Note 2	STOP Note 3, 4	RTC and f_{XT} are stopped		0.4	60	μA
			RTC is operated by f_{XT} = 32.768KHz		0.8		

Notes 1. The common condition for I_{DD2}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The HALT instruction is executed by the program in the code flash.
- The specification shows the stable current during HALT mode.

2. The common condition for I_{DD3}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The STOP instruction is executed by the program in the code flash during MAIN RUN operation.
- The spec shows the stable current during STOP mode.

3. The typical value is that when Ta=+25deg.C and V_{DD}=5.0 V. Peripheral devices and the data flash are stopped.

4. The maximum value is that when all peripheral devices are operating. But the A/D converter, LCD circuit, stepper motor circuit, the data flash and the code flash are stopped. The 16-bit wakeup timer operates with f_{LOCO}.

5. Either f_X or f_{HOCO} which is selected for f_{CLK} is operated. The other is stopped.

6. f_X and f_{HOCO} are stopped.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Typ.	Max.	Unit
WDT operating current Note 1	I_{WDT}			0.25	1.0	μA
ADC operating current Note 2	I_{ADC}	Normal mode	V_{DD} = 5.0 V	1.3	1.7	mA
		Low voltage mode	V_{DD} = 3.0 V	0.5	0.7	mA
LCD operating Current Note 3	I_{LCD}	f_{LCD} = f_{SUB}, LCD clock = 512 Hz	V_{DD} = 5.0 V	100	140	μA
			V_{DD} = 3.0 V	90	130	μA
ZPD operating current Note 4	I_{ZPD}	One ZPD operated	V_{DD} = 5.0 V	150	600	μA
			V_{DD} = 3.0 V	100	500	μA
		Four ZPDs operated	V_{DD} = 5.0 V	500	2000	μA
			V_{DD} = 3.0 V	400	1600	μA

Notes
1. Current flowing only to the watchdog timer. The maximum specification of I_{DD1}, I_{DD2} and I_{DD3} include I_{WDT}.
2. Current flowing only to the A/D converter.The current value of the RL78/D1A is the sum of I_{DD} and I_{ADC} when the A/D converter operates.
3. Current flowing only to the LCD controller/driver circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{LCD} when the LCD controller/driver circuit operates.
4. Current flowing only to the ZPD circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{ZPD} when the ZPD circuit operates.

33.2.2 Product group B

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, run mode	I_{DD1} Note 1	High speed MAIN RUN Note 2, 3, 4, 5	f_{CLK} = 32 MHz	f_{HOCO} = 32 MHz	5.7	24	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	4.7	20	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	4.3	17.5	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	2.4	12	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	1.8	9.5	mA
				f_X = 4 MHz			
		SUB RUN Note 2, 3, 6	f_{CLK} = f_{XT} = 32.768 kHz		7	360	μA

Notes 1. The common condition for I_{DD1}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The program is running in the code flash.

2. The typical value is that when Ta=+25deg.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. But the A/D converter, RTC, LCD circuit and stepper motor circuit are stopped, and the data flash and the code flash are stated to read mode. The 16-bit wakeup timer operates with f_{LOCO}.

4. Either f_X or f_{HOCO} which is selected for f_{CLK} is operated. The other is stopped.

5. At 128-pin products, the value of I_{DD1} does not include the LCDB (P11x, P46-7) pin toggle current.
I_{DD1} condition of LCDB macro is Fclk=32MHz, mod8 mode, data rate=8MHz, 4cycle, 16bit write/read.

6. f_X and f_{HOCO} are stopped.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, halt mode	I_{DD2} Note 1	High speed MAIN HALT Note 3, 4, 5	f_{CLK} = 32 MHz 2.7 V≤V_{DD}	f_{HOCO} = 32 MHz	1.0	8.9	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	0.8	7.5	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	0.7	6.5	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	0.4	4.5	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	0.35	3.8	mA
				f_X = 4 MHz			
		SUB HALT Note 3, 4, 6	f_{CLK} = f_{XT} = 32.768 kHz	RTC is stopped	1.0	140	μA
				RTC is operated by f_{XT} = 32.768KHz	1.2		
Supply current, stop mode	I_{DD3} Note 2	STOP Note 3, 4	RTC and f_{XT} are stopped		0.4	70	μA
			RTC is operated by f_{XT} = 32.768KHz		0.8		

Notes 1. The common condition for I_{DD2}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The HALT instruction is executed by the program in the code flash.
- The specification shows the stable current during HALT mode.

2. The common condition for I_{DD3}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The STOP instruction is executed by the program in the code flash during MAIN RUN operation.
- The spec shows the stable current during STOP mode.

3. The typical value is that when Ta=+25deg.C and V_{DD}=5.0 V. Peripheral devices and the data flash are stopped.

4. The maximum value is that when all peripheral devices are operating. But the A/D converter, LCD circuit, stepper motor circuit, the data flash and the code flash are stopped. The 16-bit wakeup timer operates with f_{LOCO}.

5. Either f_X or f_{HOCO} which is selected for f_{CLK} is operated. The other is stopped.

6. f_X and f_{HOCO} are stopped.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Typ.	Max.	Unit
WDT operating current Note 1	I_{WDT}			0.25	1.0	μA
ADC operating current Note 2	I_{ADC}	Normal mode	V_{DD} = 5.0 V	1.3	1.7	mA
		Low voltage mode	V_{DD} = 3.0 V	0.5	0.7	mA
LCD operating Current Note 3	I_{LCD}	f_{LCD} = f_{SUB}, LCD clock = 512 Hz	V_{DD} = 5.0 V	100	140	μA
			V_{DD} = 3.0 V	90	130	μA
ZPD operating current Note 4	I_{ZPD}	One ZPD operated	V_{DD} = 5.0 V	150	600	μA
			V_{DD} = 3.0 V	100	500	μA
		Four ZPDs operated	V_{DD} = 5.0 V	500	2000	μA
			V_{DD} = 3.0 V	400	1600	μA

Notes
1. Current flowing only to the watchdog timer. The maximum specification of I_{DD1}, I_{DD2} and I_{DD3} include I_{WDT}.
2. Current flowing only to the A/D converter.The current value of the RL78/D1A is the sum of I_{DD} and I_{ADC} when the A/D converter operates.
3. Current flowing only to the LCD controller/driver circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{LCD} when the LCD controller/driver circuit operates.
4. Current flowing only to the ZPD circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{ZPD} when the ZPD circuit operates.

33.3 Oscillator characteristics

33.3.1 Main(X1) oscillator characteristics

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Main(X1) clock oscillation frequency	f_X	Ceramic resonator	1.0		20.0	MHz
		Crystal resonator	1.0		20.0	MHz

Remark Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

33.3.2 High speed on chip oscillator characteristics

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
HOCO oscillation frequency	f_{HOCO}	4 MHz mode	3.92	4.00	4.08	MHz
		8 MHz mode	7.84	8.00	8.16	MHz
		16 MHz mode	15.68	16.00	16.32	MHz
		24 MHz mode	23.52	24.00	24.48	MHz
		32 MHz mode	31.36	32.00	32.64	MHz

Remark Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

33.3.3 Low speed on chip oscillator characteristics

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
LOCO oscillation frequency	f_{LOCO}		12.75	15.0	17.25	kHz

33.3.4 Sub(XT1) oscillator characteristics

T_A = -40 to +85 °C, 2.7 V≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Sub(XT1) clock oscillation frequency	f_{XT}	Possible to oscillate	29.0	32.768	35.0	kHz

Remark Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

33.4 DC characteristics

33.4.1 Pin group 1

T_A = -40 to +85 °C, 4.0 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V (1/2)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Output current, high[Note 1]	I_{OH1}	Per pin				-5.0	mA
		I_{OH2}	Per pin, P73 or P135 (SG port)				-13.0	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note 2])	Group 1L			-40.0	mA
<R>				Group 1R			-40.0	mA
<R>				Group 1C (128-pin, 100-pin)			-30.0	mA
<R>			for 128-pin, 100-pin				-110.0	mA
			for 80-pin, 64-pin, 48-pin				-60.0	mA
	Output current, low	I_{OL1}	Per pin				8.5	mA
		I_{OL2}	Per pin, P73 or P135 (SG ports)				13.0	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note 3])	Group 1L			40.0	mA
<R>				Group 1R			35.0	mA
<R>				Group 1C (128-pin, 100-pin)			40.0	mA
<R>			for 128-pin, 100-pin				115.0	mA
			for 80-pin, 64-pin, 48-pin				60.0	mA

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

<R> **2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -30.0 mA

Total output current of pins = (-30.0 × 0.7)/(80 × 0.01) ≈ -26.2 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.

<R> **3.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 40.0 mA

Total output current of pins = (40.0 × 0.7)/(80 × 0.01) = 35.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

T_A = -40 to +85 °C, 4.0 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V (2/2)

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Input voltage, high Note 2	V_{IH1}	Schmitt3 mode	0.8EV_{DD}		EV_{DD}	V
	V_{IH2}	Schmitt1 mode Note 3	0.65EV_{DD}		EV_{DD}	V
Input voltage, low Note 2	V_{IL1}	Schmitt3 mode	0		0.5EV_{DD}	V
	V_{IL2}	Schmitt1 mode Note 3	0		0.35EV_{DD}	V
Input hysteresis width Note 2, 4	V_{IHYS1}	Schmitt3 mode	0.1	0.19	0.29	V
	V_{IHYS2}	Schmitt1 mode Note 3	0.15	0.59	0.84	V
Output voltage, high Note 1	V_{OH1}	I_{OH} = -5.0 mA	EV_{DD}-1.0		EV_{DD}	V
		I_{OH} = -3.0 mA up to 6 pins	EV_{DD}-0.5		EV_{DD}	V
	V_{OH2}	I_{OH} = -13.0 mA, P73 or P135 (SG port)	EV_{DD}-0.7		EV_{DD}	V
Output voltage, low	V_{OL1}	I_{OL} = 8.5 mA	0		0.7	V
		I_{OL} = 3.0 mA up to 6 pins	0		0.5	V
	V_{OL2}	I_{OL} = 13.0 mA, P73 or P135 (SG port)	0		0.7	V
Input leakage current, high	I_{LIH1}	V_I = EV_{DD}			1	μA
Input leakage current, low	I_{LIL1}	V_I = EV_{SS}			-1	μA
On chip pull-up resistance Note 5	R_U	V_I = EV_{SS}	10	20	100	kΩ
On chip pull-down resistance Note 6	R_D	V_I = EV_{DD}	100			kΩ

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

2. Except P130 because it is output only port.

<R> **3.** P01, P10, P11, P17, P31, P40, P50 to P52, P55 to P57, P61, P63, P70, P110 to P117, P135 only.

4. This value is defined by evaluation result.

5. Except P130 and P137. Pull-up resistance is connected by software when pin is set to input mode.

6. LCD segment shared pins only. Pull-down resistance is connected during reset.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

T_A = -40 to +85 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ 4.0 V, EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, VSS = EV_{SS0} = EV_{SS} = 0 V (1/2)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Output current, high[Note 1]	I_{OH1}	Per pin				-1.0	mA
		I_{OH2}	Per pin, P73 or P135 (SG port)				-7.5	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note 2])	Group 1L			-15.0	mA
<R>				Group 1R			-30.0	mA
<R>				Group 1C (128-pin, 100-pin)			-7.0	mA
<R>				for 128-pin, 100-pin			-52.0	mA
				for 80-pin, 64-pin, 48-pin			-33.0	mA
	Output current, low	I_{OL1}	Per pin				1.5	mA
		I_{OL2}	Per pin, P73 or P135 (SG ports)				7.0	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note 3])	Group 1L			18.0	mA
<R>				Group 1R			30.0	mA
<R>				Group 1C (128-pin, 100-pin)			10.0	mA
<R>				for 128-pin, 100-pin			58.0	mA
				for 80-pin, 64-pin, 48-pin			35.0	mA

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

<R> **2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -7.0 mA

Total output current of pins = (-7.0 × 0.7)/(80 × 0.01) ≈ -6.1 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.

<R> **3.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

T_A = -40 to +85 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ 4.0 V, EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, VSS = EV_{SS0} = EV_{SS} = 0 V (2/2)

	Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
	Input voltage, high Note 2	V_{IH1}	Schmitt3 mode	0.8EV_{DD}		EV_{DD}	V
		V_{IH2}	Schmitt1 mode Note 3	0.7EV_{DD}		EV_{DD}	V
	Input voltage, low Note 2	V_{IL1}	Schmitt3 mode	0		0.4EV_{DD}	V
		V_{IL2}	Schmitt1 mode Note 3	0		0.3EV_{DD}	V
	Input hysterisis width Note 2, 4	V_{IHYS1}	Schmitt3 mode	0.05		0.21	V
		V_{IHYS2}	Schmitt1 mode Note 3	0.08		0.53	V
	Output voltage, high Note 1	V_{OH1}	I_{OH} = -1.0 mA	EV_{DD}-0.5		EV_{DD}	V
		V_{OH2}	I_{OH} = -7.5 mA, P73 or P135 (SG port)	EV_{DD}-0.7		EV_{DD}	V
	Output voltage, low	V_{OL1}	I_{OL} = 1.5 mA	0		0.5	V
		V_{OL2}	I_{OL}= 7.0 mA, P73 or P135 (SG port)	0		0.7	V
<R>	Input leakage current, high	I_{LIH1}	V_I = EV_{DD}			1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = EV_{SS}			-1	μA
<R>	On chip pull-up resistance Note 5	R_U	V_I = EV_{SS}	10	20	100	kΩ
<R>	On chip pull-down resistance Note 6	R_D	V_I = EV_{DD}	100			kΩ

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

2. Except P130 because it is output only port.

<R> **3.** P01, P10, P11, P17, P31, P40, P50 to P52, P55 to P57, P61, P63, P70, P110 to P117, P135 only.

4. This value is defined by evaluation result.

5. Except P130 and P137. Pull-up resistance is connected by software when pin is set to input mode.

6. LCD segment shared pins only. Pull-down resistance is connected during reset.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

33.4.2 Pin group 2 (ANI pins)

T_A = -40 to +85 °C, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

	Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
	Output current, high	I_{OH1}	Per pin			-0.1	mA
		$I_{OHTOTAL}$	Total			-0.8	mA
	Output current, low	I_{OL1}	Per pin			0.4	mA
		$I_{OLTOTAL}$	Total			3.2	mA
	Input voltage, high	V_{IH1}		0.8V_{DD}		V_{DD}	V
	Input voltage, low	V_{IL1}		0		0.5V_{DD}	V
	Input hysteresis width Note	V_{IHYS1}		0.1	0.19	0.29	V
	Output voltage, high	V_{OH1}	I_{OH} = -0.1 mA	V_{DD}-0.5		V_{DD}	V
	Output voltage, low	V_{OL1}	I_{OL} = 0.4 mA	0		0.4	V
<R>	Input leakage current, high	I_{LIH1}	V_I = V_{DD}			1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = V_{SS}			-1	μA

Note This specification is guaranteed by design. It is not tested when shipment.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 4.0 V, V_{SS} = 0 V

	Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
	Output current, high	I_{OH1}	Per pin			-0.1	mA
		$I_{OHTOTAL}$	Total			-0.8	mA
	Output current, low	I_{OL1}	Per pin			0.4	mA
		$I_{OLTOTAL}$	Total			3.2	mA
	Input voltage, high	V_{IH1}		0.8V_{DD}		V_{DD}	V
	Input voltage, low	V_{IL1}		0		0.4V_{DD}	V
	Input hysteresis width Note	V_{IHYS1}		0.05		0.21	V
	Output voltage, high	V_{OH1}	I_{OH} = -0.1 mA	V_{DD}-0.5		V_{DD}	V
	Output voltage, low	V_{OL1}	I_{OL} = 0.4 mA	0		0.4	V
<R>	Input leakage current, high	I_{LIH1}	V_I = V_{DD}			1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = V_{SS}			-1	μA

Note This specification is guaranteed by design. It is not tested when shipment.

33.4.3 Pin group 3 (SMC pins)

T_A = -40 to +85 °C, 4.0 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (1/3)

	Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
	Output current, high	I_{OH1}	Per pin		T_A = -40 °C			-52	mA
					T_A = +25 °C			-39	mA
					T_A = +85 °C			-32	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A	T_A = -40 °C			-118	mA
<R>					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
				Group 3B	T_A = -40 °C			-118	mA
					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
<R>				Group 3C 128-pin, 100-pin 80-pin	T_A = -40 °C			-118	mA
					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
				Group 3C 64-pin 48-pin	T_A = -40 °C			-118	mA
					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
<R>				Group 3D (128-pin, 100-pin, 80-pin)	T_A = -40 °C			-118	mA
					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
				Group 3E (64-pin, 48-pin)	T_A = -40 °C			-148	mA
					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -118.0 mA

Total output current of pins = (-118.0 × 0.7)/(80 × 0.01) ≈ -103.2 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.

T_A = -40 to +85 °C, 4.0 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (2/3)

	Parameter	Symbols	Conditions				Min.	Typ.	Max.	Unit
	Output current, low	I_{OL1}	Per pin			T_A = -40 °C			52	mA
						T_A = +25 °C			39	mA
						T_A = +85 °C			32	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A		T_A = -40 °C			118	mA
<R>						T_A = +25 °C			118	mA
						T_A = +85 °C			96	mA
				Group 3B		T_A = -40 °C			118	mA
						T_A = +25 °C			118	mA
						T_A = +85 °C			96	mA
<R>				Group 3C	128-pin, 100-pin 80-pin	T_A = -40 °C			118	mA
						T_A = +25 °C			118	mA
						T_A = +85 °C			96	mA
					64-pin 48-pin	T_A = -40 °C			118	mA
						T_A = +25 °C			118	mA
						T_A = +85 °C			96	mA
<R>				Group 3D (128-pin, 100-pin, 80-pin)		T_A = -40 °C			118	mA
						T_A = +25 °C			118	mA
						T_A = +85 °C			96	mA
				Group 3E (64-pin, 48-pin)		T_A = -40 °C			148	mA
						T_A = +25 °C			118	mA
						T_A = +85 °C			96	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 118.0 mA
Total output current of pins = (118.0 × 0.7)/(80 × 0.01) ≈ 103.2 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.

T_A = -40 to +85 °C, 4.0 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (3/3)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Input voltage, high	V_{IH1}			0.8SMV_{DD}		SMV_{DD}	V
	Input voltage, low	V_{IL1}			0		0.5SMV_{DD}	V
	Input hysteresis width Note 1	V_{IHYS1}			0.1	0.19	0.29	V
	Output voltage, high	V_{OH1}	T_A = -40 °C	I_{OH} = -52 mA	SMV_{DD} - 0.5		SMV_{DD}	V
			T_A = +25 °C	I_{OH} = -39 mA				
			T_A = +85 °C	I_{OH} = -32 mA				
	Output voltage, low	V_{OL1}	T_A = -40 °C	I_{OL} = 52 mA	0		0.5	V
			T_A = +25 °C	I_{OL} = 39 mA				
			T_A = +85 °C	I_{OL} = 32 mA				
	Output voltage deviation Note 2	V_{DEV}			0		50	mV
<R>	Input leakage current, high	I_{LIH1}	V_I = SMV_{DD}				1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = SMV_{SS}				-1	μA
<R>	On chip pull-up resistance Note 3	R_U	V_I = SMV_{SS}		10	20	100	kΩ
<R>	On chip pull-down resistance Note 4	R_D	V_I = SMV_{DD}		100			kΩ

Notes 1. This specification is guaranteed by design. It is not tested when shipment.

2. Output voltage deviation defines the difference of the outputs levels of the same stepper motor.
V_{DEV} = max ($|V_{OHx}-V_{OHy}|$, $|V_{OLx}-V_{OLy}|$) @I_{OHx} = I_{OHy}, I_{OLx} = I_{OLy}.
X and y denote any combination of two pins of the following pin groups: (P80-P83, P84-P87, P90-P93, P94-P97)

3. Pull-up resistance is connected by software when pin is set to input mode.

4. LCD segment shared pins only. Pull-down resistance is connected during reset.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 4.0 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (1/3)

	Parameter	Symbols	Conditions				Min.	Typ.	Max.	Unit
	Output current, high	I_{OH1}	Per pin			T_A = -40 °C			-30	mA
						T_A = +25 °C			-25	mA
						T_A = +85 °C			-23	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A		T_A = -40 °C			-90	mA
<R>						T_A = +25 °C			-75	mA
						T_A = +85 °C			-69	mA
				Group 3B		T_A = -40 °C			-90	mA
						T_A = +25 °C			-75	mA
						T_A = +85 °C			-69	mA
<R>				Group 3C	128-pin, 100-pin 80-pin	T_A = -40 °C			-90	mA
						T_A = +25 °C			-75	mA
						T_A = +85 °C			-69	mA
					64-pin 48-pin	T_A = -40 °C			-90	mA
						T_A = +25 °C			-75	mA
						T_A = +85 °C			-69	mA
<R>				Group 3D (128-pin, 100-pin, 80-pin)		T_A = -40 °C			-90	mA
						T_A = +25 °C			-75	mA
						T_A = +85 °C			-69	mA
				Group 3E (64-pin, 48-pin)		T_A = -40 °C			-90	mA
						T_A = +25 °C			-75	mA
						T_A = +85 °C			-69	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -75.0 mA

Total output current of pins = (-75.0 × 0.7)/(80 × 0.01) ≈ -65.6 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.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 4.0 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (2/3)

	Parameter	Symbols	Conditions				Min.	Typ.	Max.	Unit
	Output current, low	I_{OL1}	Per pin			T_A = -40 °C			30	mA
						T_A = +25 °C			23	mA
						T_A = +85 °C			20	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A		T_A = -40 °C			90	mA
<R>						T_A = +25 °C			69	mA
						T_A = +85 °C			60	mA
				Group 3B		T_A = -40 °C			90	mA
						T_A = +25 °C			69	mA
						T_A = +85 °C			60	mA
<R>				Group 3C	128-pin, 100-pin 80-pin	T_A = -40 °C			90	mA
						T_A = +25 °C			69	mA
						T_A = +85 °C			60	mA
					64-pin 48-pin	T_A = -40 °C			90	mA
						T_A = +25 °C			69	mA
						T_A = +85 °C			60	mA
<R>				Group 3D (128-pin, 100-pin, 80-pin)		T_A = -40 °C			90	mA
						T_A = +25 °C			69	mA
						T_A = +85 °C			60	mA
				Group 3E (64-pin, 48-pin)		T_A = -40 °C			90	mA
						T_A = +25 °C			69	mA
						T_A = +85 °C			60	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 69.0 mA
Total output current of pins = (69.0 × 0.7)/(80 × 0.01) ≈ 60.3 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.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 4.0 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (3/3)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Input voltage, high	V_{IH1}			0.8SMVDD		SMV_{DD}	V
	Input voltage, low	V_{IL1}			0		0.4SMV_{DD}	V
	Input hysteresis width Note 1	V_{IHYS1}			0.05		0.21	V
	Output voltage, high	V_{OH1}	T_A = -40 °C	I_{OH} = -30 mA	SMVDD - 0.5		SMV_{DD}	V
			T_A = +25 °C	I_{OH} = -25 mA				
			T_A = +85 °C	I_{OH} = -23 mA				
	Output voltage, low	V_{OL1}	T_A = -40 °C	I_{OL} = 30 mA	0		0.5	V
			T_A = +25 °C	I_{OL} = 23 mA				
			T_A = +85 °C	I_{OL} = 20 mA				
	Output voltage deviation Note 2	V_{DEV}			0		50	mV
<R>	Input leakage current, high	I_{LIH1}	V_I = SMV_{DD}				1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = SMV_{SS}				-1	μA
<R>	On chip pull-up resistance Note 3	R_U	V_I = SMV_{SS}		10	20	100	kΩ
<R>	On chip pull-down resistance Note 4	R_D	V_I = SMV_{DD}		100			kΩ

Notes 1. This speccification is guaranteed by design. It is not tested when shipment.

2. Output voltage deviation defines the difference of the outputs levels of the same stepper motor.
V_{DEV} = max ($|V_{OHx}-V_{OHy}|$, $|V_{OLx}-V_{OLy}|$) @I_{OHx} = I_{OHy}, I_{OLx} = I_{OLy}.
X and y denote any combination of two pins of the following pin groups: (P80-P83, P84-P87, P90-P93, P94-P97)

3. Pull-up resistance is connected by software when pin is set to input mode.

4. LCD segment shared pins only. Pull-down resistance is connected during reset.

33.4.4 Pin group 4 (OSC, reset and P137 pins)

T_A = -40 to +85 °C, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0.8V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$		0.65V_{DD}		V_{DD}	V
	V_{IH3}	P137		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0		0.2V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$		0		0.35V_{DD}	V
	V_{IL3}	P137		0		0.5V_{DD}	V
Input hysteresis width Note 2	V_{IHYS1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0.1	0.7		V
<R>	V_{IHYS2}	$\overline{\text{RESET}}$		0.15	0.59	0.84	V
Input leakage current, high	I_{LIH1}	P121, P122, P123, P124 $V_I = V_{DD}$	Port			1	μA
			EXCLK Note 1			1	μA
			OSC			10	μA
	I_{LIH2}	$\overline{\text{RESET}}$, $V_I = V_{DD}$				1	μA
	I_{LIH3}	P137, $V_I = V_{DD}$				1	μA
Input leakage current, low	I_{LIL1}	P121, P122, P123, P124 $V_I = V_{SS}$	Port			-1	μA
			EXCLK Note 1			-1	μA
			OSC			-10	μA
	I_{LIL2}	$\overline{\text{RESET}}$, $V_I = V_{SS}$				-1	μA
	I_{LIL3}	P137, $V_I = V_{SS}$				-1	μA

Notes 1. P122(EXCLK) only.

2. This speccification is guaranteed by design. It is not tested when shipment.

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 4.0 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P121,P122,P123,P124 (Port or EXCLK Note 1)		0.8V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$		0.7V_{DD}		V_{DD}	V
	V_{IH3}	P137		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0		0.2V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$		0		0.3V_{DD}	V
	V_{IL3}	P137		0		0.4V_{DD}	V
Input hysteresis width Note 2	V_{IHYS1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0.08			V
<R>	V_{IHYS2}	$\overline{\text{RESET}}$		0.08			V
Input leakage current, high	I_{LIH1}	P121, P122, P123, P124 V_I = V_{DD}	Port			1	μA
			EXCLK Note 1			1	μA
			OSC			10	μA
	I_{LIH2}	$\overline{\text{RESET}}$, V_I = V_{DD}				1	μA
	I_{LIH3}	P137, V_I = V_{DD}				1	μA
Input leakage current, low	I_{LIL1}	P121, P122, P123, P124 V_I = V_{SS}	Port			-1	μA
			EXCLK Note 1			-1	μA
			OSC			-10	μA
	I_{LIL2}	$\overline{\text{RESET}}$, V_I = V_{SS}				-1	μA
	I_{LIL3}	P137, V_I = V_{SS}				-1	μA

Notes 1. P122(EXCLK) only.

2. This speccification is guaranteed by design. It is not tested when shipment.

33.5 AC characteristics

33.5.1 Basic operation

T_A = -40 to +85 °C,

<R> 2.7 V ≤ = EV_{DD0} = EV_{DD1} ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock operation (Including PLL)	High speed main run		0.03125 Note 1		1	μs
		Sub system clock operation	SDIV = 0		1/f_{XT} (Typ. 30.5)			μs
<R> External main system clock frequency	f_{EX}	Square wave input to EXCLK			2		20	MHz
External main system clock (square wave) input high/low level width	t_{EXH} t_{EXL}	Square wave input to EXCLK			24			ns
Timer input high/low level width	t_{TIH} t_{TIL}	TI00 to TI07, TI10 to TI17, TI20 to TI27			1/f_{MCK}+10 Note 2			ns
<R> Port output frequency	f_{GPO}	P80 to P87, P90 to P97	C = 30 pF	4.0 V ≤ SMV_{DD}			2	MHz
<R>		P20 to P27, P150 to P152	C = 30 pF	4.0 V ≤ V_{DD}			2	MHz
<R>		P73,P135	C = 30 pF	4.0 V ≤ EV_{DD}			8	MHz
<R>		Other than the above	C = 30 pF	4.0 V ≤ EV_{DD}			16	MHz
<R>				2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
External interrupt input high/low level width Note 3	t_{INIH} t_{INIL}	INTP0 to INTP5, INTPLR0, INTPLR1			1			μs
$\overline{RESET}$ input low level width Note 3	t_{RSL}	$\overline{RESET}$			10			μs
Analog noise filter rejection pulse width Note 4	t_{WRJ}	INTP0 to INTP5, INTPLR0, INTPLR1, ADTRG			30	50	1000	ns
ADTRG input high level width	t_{ATH}	Without noise filter	AWC = 0		1/f_{CLK}+10			ns
			AWC = 1		10			ns
		With noise filter	AWC = 0		1/f_{CLK}+10 or t_{WRJ} (**Note 5**)			ns
			AWC = 1		t_{WRJ} (**Note 5**)			ns

Notes 1. Value is in case of f_{CLK} is 32.0 MHz. It is also allowed to exceed frequency up to +3%.

2. f_{MCK} shows the frequency value of operation clock for TAU. Usually, f_{MCK} is defined by MHz but this speccification is defined by ns. It is not defined by μs, so please be careful.

3. Pulses longer than this value will pass the input filter.

4. Pulses shorter than this value do not pass the input filters.

5. If the value of "1/f_{CLK}+10 [ns]" is less than t_{WRJ}, please use t_{WRJ} value instead of "1/f_{CLK} +10 [ns]".

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 33-1. AC Timing Test Points

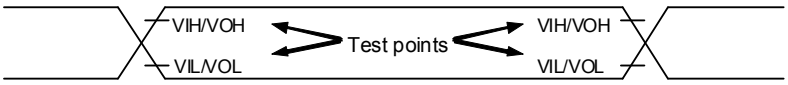


Figure 33-2. External Main System Clock Timing

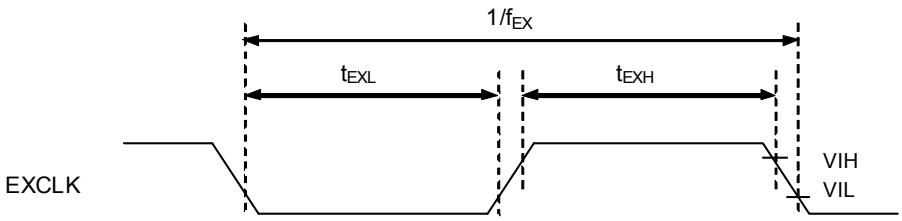


Figure 33-3. TI Timing

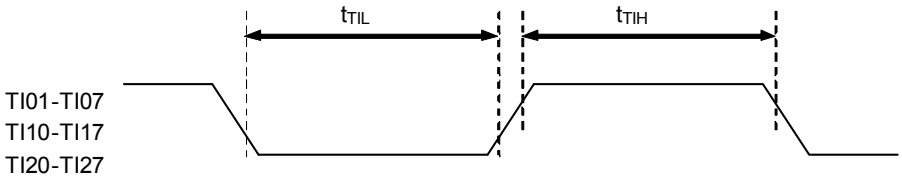


Figure 33-4. Interrupt Request Input Timing

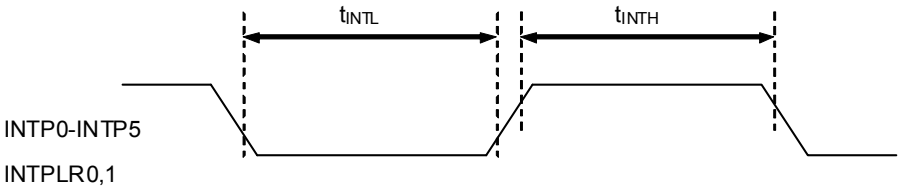
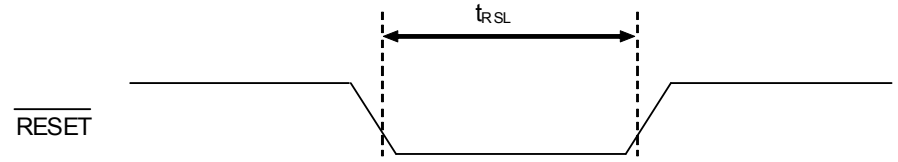


Figure 33-5. $\overline{\text{RESET}}$ Input Timing



33.5.2 Stepper motor controller/driver

T_A = -40 to +85 °C,

<R> 2.7 V ≤ SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Meter Controller/Driver input frequency	f_{MC} Note 1					32	MHz
PWM output rise time	t_R	10%-90% Note 2	4.0 V ≤ SMV_{DD} ≤ 5.5 V	15	60	100	ns
<R>			2.7 V ≤ SMV_{DD} ≤ 4.0 V	20		500	ns
PWM output fall time	t_F	10%-90% Note 2	4.0 V ≤ SMV_{DD} ≤ 5.5 V	15	60	100	ns
<R>			2.7 V ≤ SMV_{DD} ≤ 4.0 V	20		500	ns
Peak Cross Current Note 3	I_{CROSS}					50	ns
Output Pulse Width Note 4	t_{MO}	4.0 V≤SMV_{DD}≤5.5 V		250			ns
		2.7 V≤SMV_{DD}≤5.5 V		5000			ns
Output Pulse Length Deviation Note 5	t_{SMDEV}	4.0 V≤SMV_{DD}≤5.5 V		-65	-12	+10	ns
		2.7 V≤SMV_{DD}≤5.5 V		-100		+400	ns
Symmetry performance Note 6	ΔHSPmn	I_{OH} = -32 mA ΔHSPmn = | V_{OH}[(SMmn)max - (SMmn)min] |	2.7 V ≤ SMV_{DD} ≤ 5.5 V			50	mV
	ΔHSPmn	I_{OL} = 32 mA ΔHSPmn = | V_{OL}[(SMmn)max - (SMmn)min] |	4.0 V ≤ SMV_{DD} ≤ 5.5 V			50	mV
			2.7 V ≤ SMV_{DD} ≤ 5.5 V			100	mV

Notes 1. Source clock of the free-running counter.

2. t_R,t_F is not tested in production, specified by design.

3. The slew rate control generates a cross current in the output stage to control the energy of the external inductive load. The cross current flows only during the output transition time t_R, t_F. It flows in addtion to the output current. The cross current is not tested,but derived from simulation.

4. The output buffer can not generate high or low pulses shorter than this time, because of its slew rate control system. This value is not tested,but derived from simulation.

5. The slew rate control function causes a deviation of output pulse time compared to the ideal selected output pulse setting. This value is not tested,but derived from simulation.

6. Indicates the dispersion of 16 PWM output voltages.(4 buffers' output voltage differences in the state of I_{OH}(I_{OL}) at the same time.) Not tested in production, specified by design.

Remark m = 1 to 4, n = 1 to 4

<R> 33.5.3 Sound generator

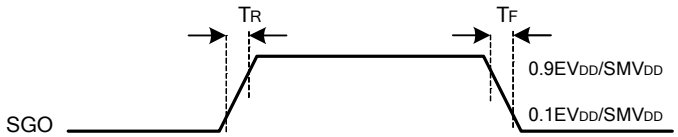
T_A = -40 to +85 °C,

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Sound generator input frequency	f_{SG}					32	MHz
SGO output rise time	f_R	C = 100 pF	P73, P135			200	ns
			P93			500	
SGO output fall time	f_F	C = 100 pF	P73, P135			200	ns
			P93			500	

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Sound Generator Output Timing



33.5.4 Serial interface: CSI operation

<R> <Master mode>

T_A = -40 to +85 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	Min.		Max.	Unit
SCK cycle time	t_{KCY1}	4.0≤V_{DD}	125	t_{KCY1}≥4/f_{CLK} Note		ns
		V_{DD}<4.0 V	250			ns
SCK high/low level width	t_{KH1}	4.0≤V_{DD}	t_{KCY1}/2-12			ns
	t_{KL1}	V_{DD}<4.0 V	t_{KCY1}/2-18			ns
SI set up time	t_{SIK1}	4.0≤V_{DD}	44			ns
		V_{DD}<4.0 V	55			ns
SI hold time	t_{KSI1}		19			ns
SO output delay time	t_{KSO1}	C = 30 pF			25	ns

Note When CSI transfer is operated by DMA, it is necessary to concider DMA response time to decide cycle time.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

<R> <Slave mode>

T_A = -40 to +85 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions		Min.	Max.	Unit
SCK cycle time	t_{KCY2}	4.0≤EV_{DD}	20 MHz<f_{MCK}	8/f_{MCK} Note		ns
			f_{MCK}≤20 MHz	6/f_{MCK} Note		ns
		EV_{DD}<4.0 V	16 MHz<f_{MCK}	8/f_{MCK} Note		ns
			f_{MCK}≤16 MHz	6/f_{MCK} Note		ns
SCK high/low level width	t_{KH2} t_{KL2}			t_{KCY2}/2-8		ns
SI set up time	t_{SIK2}	2.7≤EV_{DD}		1/f_{MCK}+20		ns
SI hold time	t_{KSI2}			1/f_{MCK}+31		ns
SO output delay time	t_{KSO2}	C = 30pF	4.0≤EV_{DD}		2/f_{MCK}+44	ns
			EV_{DD}<4.0 V		2/f_{MCK}+57	ns

Note When CSI transfer is operated by DMA, it is necessary to concider DMA response time to decide cycle time.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 33-6. CSI mode connection diagram

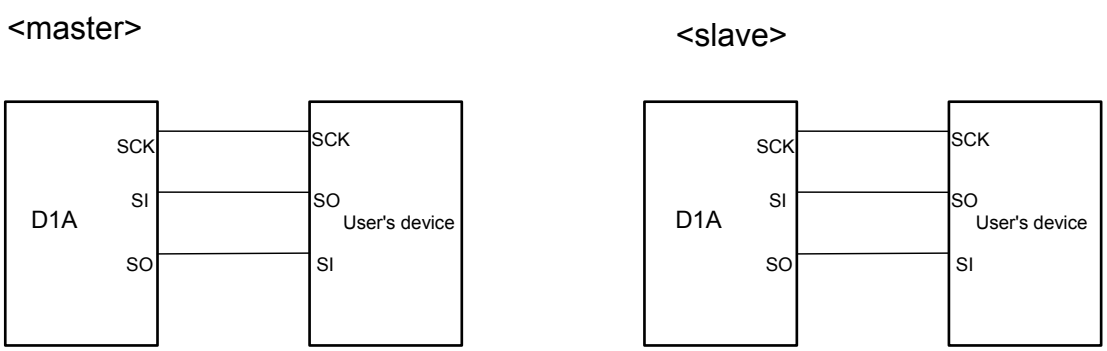


Figure 33-7. CSI mode serial transfer timing (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)

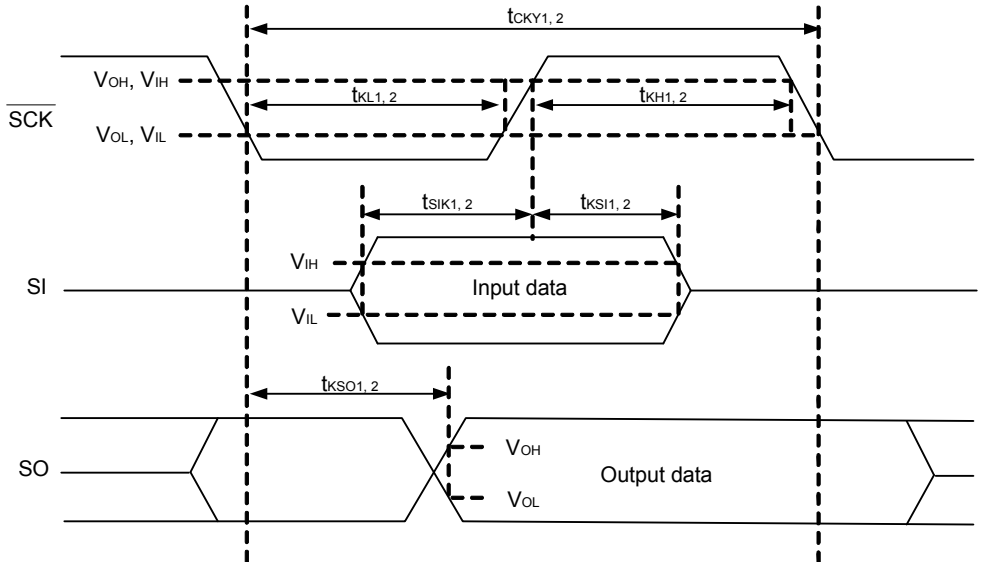
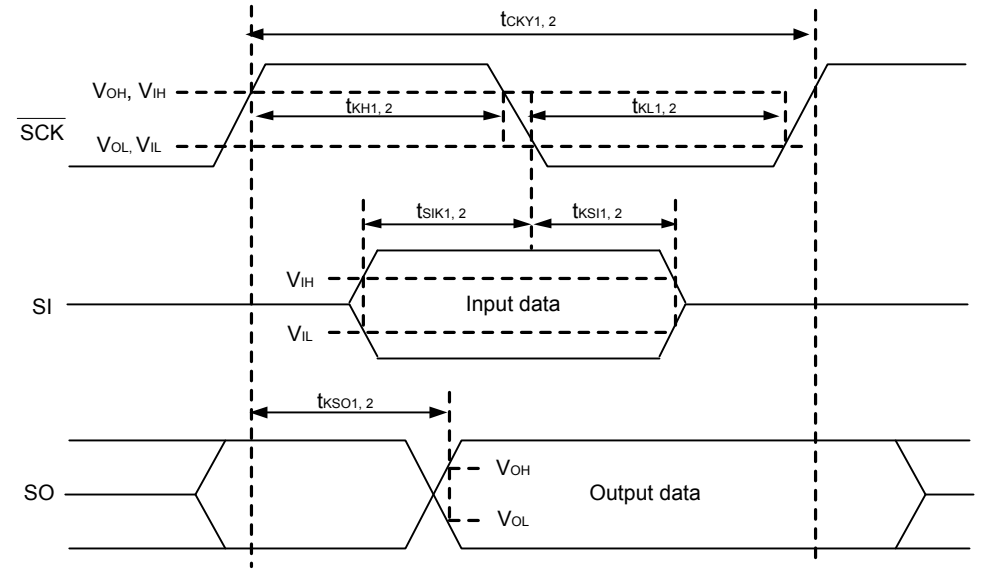


Figure 33-8. CSI mode serial transfer timing (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



<R> 33.5.5 Serial interface: UART operation (128-pin only)

T_A = -40 to +85 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} = ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Item	Symbol	Conditions	Min.	Max.	Unit
Transfer rate	T			$f_{MCK}/6$ Note	bps
		Theoretical value of the maximum transfer rate f_{CLK} = 32 MHz, f_{MCK} = f_{CLK}		5.3	Mbps

Note When CSI transfer is operated by DMA, it is necessary to concider DMA response time to decide cycle time.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 33-9 UART connection diagram

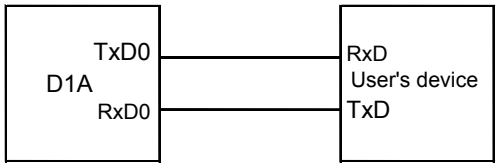
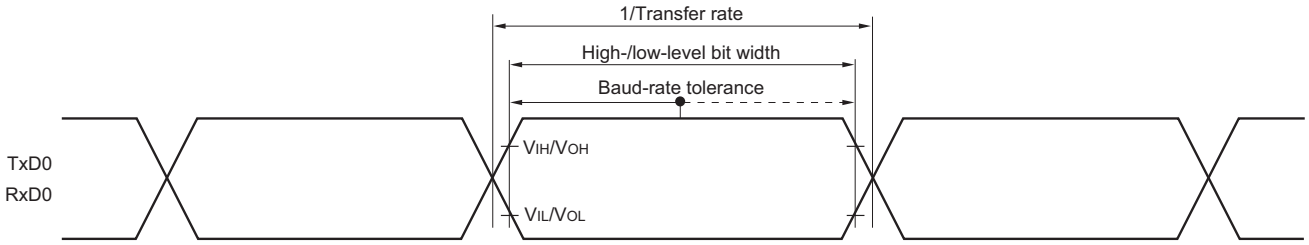


Figure 33-10. UART mode bit width (reference)



33.5.6 Serial interface: simplified I^2C operation

T_A = -40 to +85 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} = ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

	Items	Symbols	Conditions	Min.	Max.	Unit
	SCL clock frequency	f_{SCL}	Rb = 3 kΩ, Cb = 100 pF		400	kHz
	Hold time during SCL = ”L”	t_{LOW}	Rb = 3 kΩ, Cb = 100 pF	1150		ns
	Hold time during SCL = ”H”	t_{HIGH}	Rb = 3kΩ, Cb = 100 pF	1150		ns
<R>	Data set up time (reception)	$t_{SU;DAT}$	Rb = 3 kΩ, Cb = 100 pF	1/f_{MCK}+85		ns
<R>	Data hold time (transmission)	$t_{HD;DAT}$	Rb = 3 kΩ, Cb = 100 pF	0	305	ns

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 33-11 simplified I^2C connection diagram

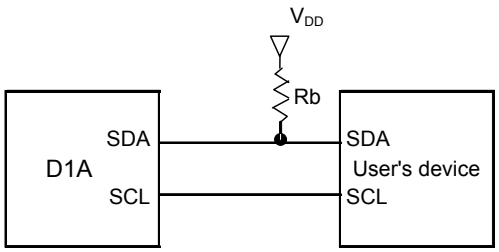
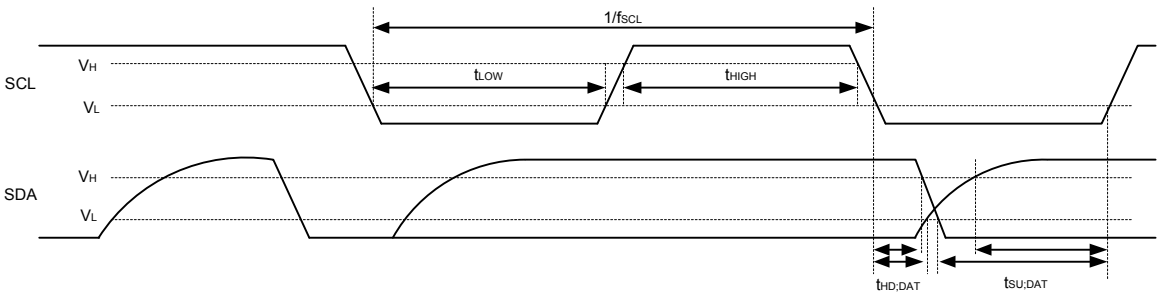


Figure 33-12. Simplified I^2C mode serial transfer timing



33.5.7 Serial interface: LIN-UART(UARTF) operation

T_A = -40 to +85 °C

<R> 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Item	Symbols	Conditions	Min.	Max.	Unit
Transfer rate	T			1.0	Mbps

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

<R> 33.5.8 Serial interface: CAN operation

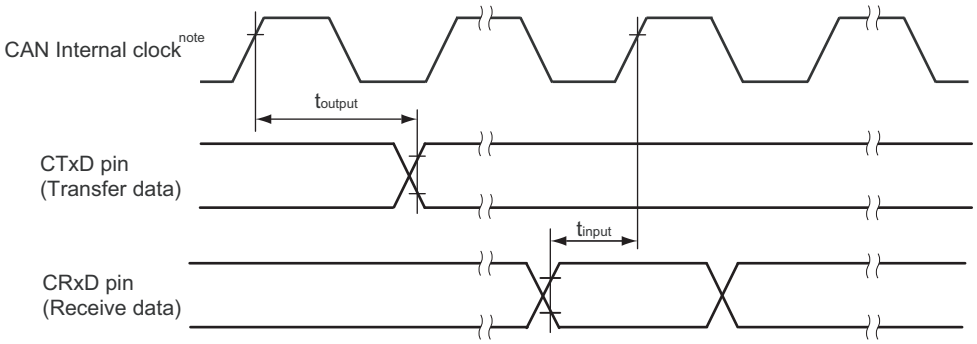
T_A = -40 to +85 °C,

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	Min.	Max.	Unit
Transfer rate	T			1.0	Mbps
Internal delay time	t_{NODE}			100	ns
CRxD minimum pulse width for wake up	t_{CRXW}	Necessary width to detect wakeup signal	200		ns

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 33-13. Internal delay time of CAN



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

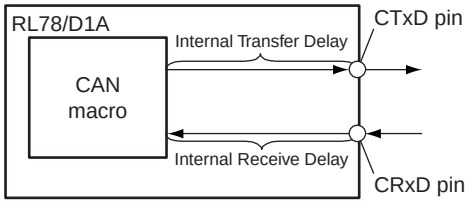


Image figure of internal delay

<R> 33.6 LCD Bus Interface characteristics (128-pin products only)

T_A = -40 to +85 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} =EV_{SS1} =0 V

Items	Symbols	Conditions	MIN.	MAX.	Unit
Transfer frequency	F			8	MHz
CycleTime	t_{CYC}		LBCYC x T		ns
Control low pulse width	t_{CL}		(LBWST + 1)T - 1		ns
Enable active pulse width	t_{ELH}		(LBWST + 1)T - 5		ns
Control setup time	t_{RWS}		$0.5T_S$ - 8		ns
Control hold time	t_{RWH}		0.5T - 3		ns
Data output setup time	t_{DOS}		$0.5T_S$ - 6	$0.5T_S$ + 17	ns
Data output hold time	t_{DOH}		{LBCYC - (LBWST + 1.5)} T - 27		ns
Data input setup time	t_{DIS}		50		ns
Data input hold time	t_{DIH}		0		ns
Output disable time	t_{OD}		0.5T - 14		ns

Remarks 1. T = (1/f_{CLK}) × n (n: LCD Bus Interface clock n divider setting)

2. T_S = (1/f_{CLK}) × N (N: LCD Bus Interface no clock divider N = 1, n divider N = n - 1)

3. F = 1/t_{CYC}

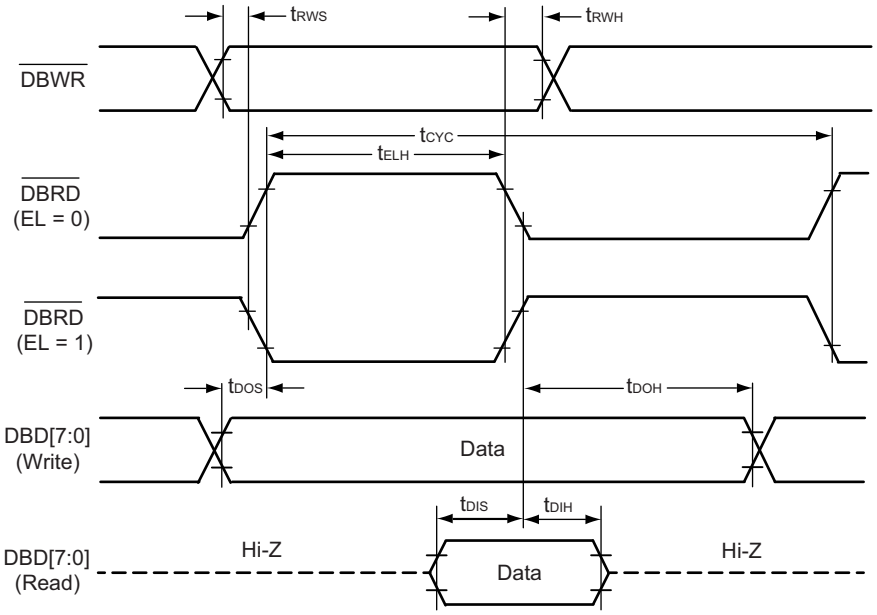
4. When EV_{DDx} = SMV_{DD}x ≤ V_{DD}, LCD controller/driver related registers must be initial value (LCDON = 0, SCOC = 0, MDSET1 – 0 = 00, LCDPFx = 0).

5. The above table shows the timing of LCD bus interface with Schmitt1 input characteristic.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 33-14. LCD Bus Interface AC timing (1/2)

(a) LCDB mod68 mode timing



(b) mode68 turnaround timing

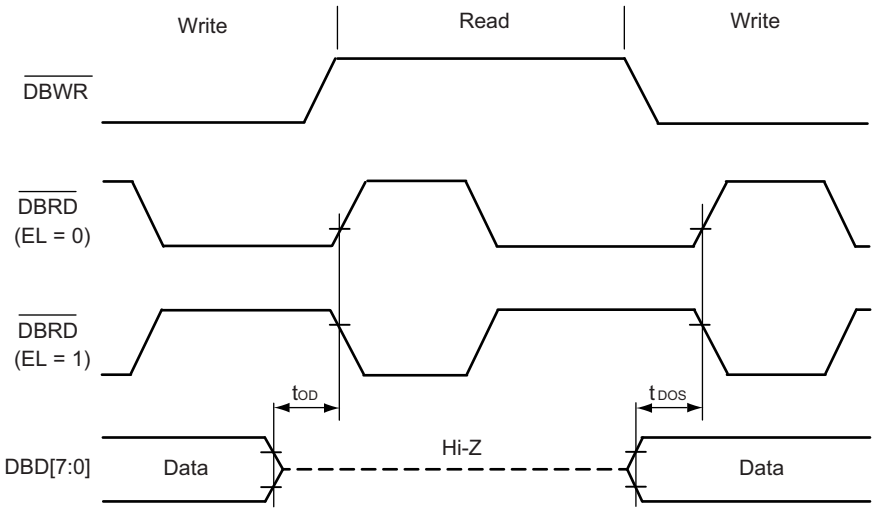
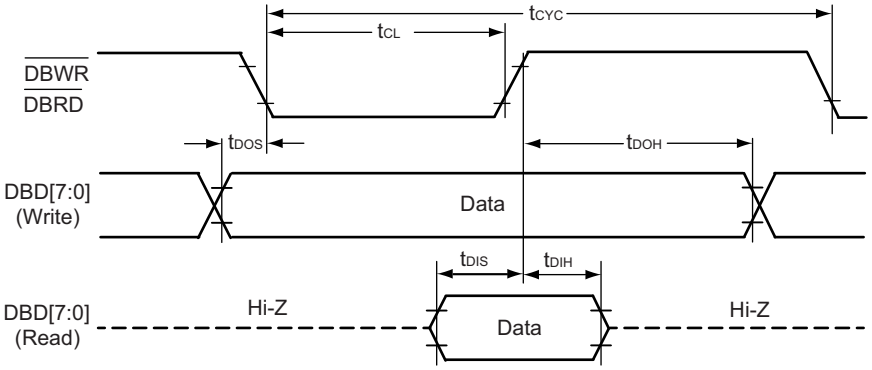
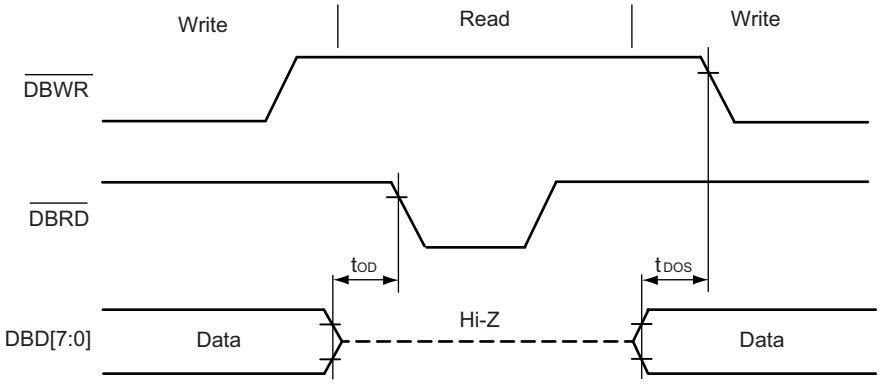


Figure 33-14. LCD Bus Interface AC timing (2/2)

(c) LCDB mod80 mode timing



(d) Mode80 turnaround timing



33.6 LCD characteristics

<R> T_A = -40 to +85 °C, 3.2 V ≤ EV_{DD0} = EV_{DD1} = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
LCD division resistance Note 1	R_{LCD}				3	kΩ
LCD Segment output voltage (unloaded)	V_{ODS}	Io = ±1 μA	V_{LCDn}-0.05	V_{LCDn} Note 2	V_{LCDn}+0.05	V
LCD Common output voltage (unloaded)	V_{ODC}	Io = ±1 μA	V_{LCDn}-0.05	V_{LCDn}	V_{LCDn}+0.05	V
<R> LCD Segment Output Voltage (loaded)	V_{ODSL0}	Io = ±10 μA, all segment pins at same time	V_{LCD0}-0.6	V_{LCD0}	V_{LCD0}+0.6	V
<R>	V_{ODSL1}	Io = ±10 μA, all segment pins at same time	V_{LCD1}-0.6	V_{LCD1}	V_{LCD1}+0.6	V
<R>	V_{ODSL2}	Io = ±10 μA, all segment pins at same time	V_{LCD2}-0.6	V_{LCD2}	V_{LCD2}+0.6	V
<R>	V_{ODSL3}	Io = ±10 μA, all segment pins at same time	V_{LCD3}-0.6	V_{LCD3}	V_{LCD3}+0.6	V
LCD Common Output Voltage (loaded)	V_{ODCL0}	Io = ±40 μA, single pin	V_{LCD0}-0.2	V_{LCD0}	V_{LCD0}+0.2	V
	V_{ODCL1}	Io = ±40 μA, single pin	V_{LCD1}-0.2	V_{LCD1}	V_{LCD1}+0.2	V
	V_{ODCL2}	Io = ±40 μA, single pin	V_{LCD2}-0.2	V_{LCD2}	V_{LCD2}+0.2	V
	V_{ODCL3}	Io = ±40 μA, single pin	V_{LCD3}-0.2	V_{LCD3}	V_{LCD3}+0.2	V
LCD split voltage drive capability Note 1	V_{LC0}	Io = ±530 μA	V_{LCD0}-0.1	V_{LCD0}	V_{LCD0}+0.1	V
	V_{LC1}	Io = ±530 μA	V_{LCD1}-0.1	V_{LCD1}	V_{LCD1}+0.1	V
	V_{LC2}	Io = ±530 μA	V_{LCD2}-0.1	V_{LCD2}	V_{LCD2}+0.1	V
	V_{LC3}	Io = ±530 μA	V_{LCD3}-0.1	V_{LCD3}	V_{LCD3}+0.1	V
LCD output resistance (COM) Note 3	R_{ODC}				8	kΩ
LCD output resistance (SEG) Note 3	R_{ODS}				8	kΩ

Notes 1. Only internal connection. The vaule is design specification.

2. V_{LCDn} (n= 0..3) represents one of the four possible voltage levels at the LCD pins. See table below for reference.

V_{LCDn}	no step-down transforming	step-down transforming
V_{LCD0}	V_{DD}	3/5 V_{DD}
V_{LCD1}	2/3 V_{DD}	2/5 V_{DD}
V_{LCD2}	1/3 V_{DD}	1/5 V_{DD}
V_{LCD3}	V_{SS}	V_{SS}

3. RODC is internal equivalent weight resistance from COM pin + COM IOBUF resistance.
RODS is internal equivalent weight resistance from SEG pin +SEG IOBUF resistance.

T_A = -40 to +85 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
LCD division resistance Note 1	R_{LCD}				3	kΩ
LCD Segment output voltage (unloaded)	V_{ODS}	Io = ±1 μA	V_{LCDn}-0.05	V_{LCDn} Note 2	V_{LCDn}+0.05	V
LCD Common output voltage (unloaded)	V_{ODC}	Io = ±1 μA	V_{LCDn}-0.05	V_{LCDn}	V_{LCDn}+0.05	V
LCD Segment Output Voltage (loaded)	V_{ODSL0}	Io = ±5 μA, all segment pins at same time	V_{LCD0}-0.6	V_{LCD0}	V_{LCD0}+0.6	V
	V_{ODSL1}	Io = ±5 μA, all segment pins at same time	V_{LCD1}-0.6	V_{LCD1}	V_{LCD1}+0.6	V
	V_{ODSL2}	Io = ±5 μA, all segment pins at same time	V_{LCD2}-0.6	V_{LCD2}	V_{LCD2}+0.6	V
	V_{ODSL3}	Io = ±5 μA, all segment pins at same time	V_{LCD3}-0.6	V_{LCD3}	V_{LCD3}+0.6	V
LCD Common Output Voltage (loaded)	V_{ODCL0}	Io = ±25 μA, single pin	V_{LCD0}-0.2	V_{LCD0}	V_{LCD0}+0.2	V
	V_{ODCL1}	Io = ±25 μA, single pin	V_{LCD1}-0.2	V_{LCD1}	V_{LCD1}+0.2	V
	V_{ODCL2}	Io = ±25 μA, single pin	V_{LCD2}-0.2	V_{LCD2}	V_{LCD2}+0.2	V
	V_{ODCL3}	Io = ±25 μA, single pin	V_{LCD3}-0.2	V_{LCD3}	V_{LCD3}+0.2	V
LCD split voltage drive capability	V_{LC0}	Io = ±530 μA	V_{LCD0}-0.1	V_{LCD0}	V_{LCD0}+0.1	V
	V_{LC1}	Io = ±530 μA	V_{LCD1}-0.1	V_{LCD1}	V_{LCD1}+0.1	V
	V_{LC2}	Io = ±530 μA	V_{LCD2}-0.1	V_{LCD2}	V_{LCD2}+0.1	V
	V_{LC3}	Io = ±530 μA	V_{LCD3}-0.1	V_{LCD3}	V_{LCD3}+0.1	V
LCD output resistance (COM) Note 3	R_{ODC}				10	kΩ
LCD output resistance (SEG) Note 3	R_{ODS}				10	kΩ

Notes 1. V_{LCDn} (n= 0..3) represents one of the four possible voltage levels at the LCD pins. See table below for reference.

V_{LCDn}	no step-down transforming
V_{LCD0}	V_{DD}
V_{LCD1}	2/3 V_{DD}
V_{LCD2}	1/3 V_{DD}
V_{LCD3}	V_{SS}

2. Only internal connection. The vaule is design specification.

3. RODC is internal equivalent weight resistance from COM pin + COM IOBUF resistance.
RODS is internal equivalent weight resistance from SEG pin +SEG IOBUF resistance.

33.7 Analog characteristics

33.7.1 A/D converter characteristics

<R>

T_A = -40 to +85 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} =0 V
Reference voltage (+) = AV_{REFP}, Reference voltage(–) = AV_{REFM}

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	Bit
Overall error Note 1,2	AINL	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V			1.2	±3.5	LSB
Conversion time	Tconv	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V	3.6 V≤V_{DD}<5.5 V	2.125		39	μs
			2.7 V≤V_{DD}<5.5 V	3.1875		39	μs
<R> Zero-scale error Note 1, 2	E_{ZS}	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				0.25	%FSR
<R> Full-scale error Note 1, 2	E_{FS}	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				0.25	%FSR
Integral non-linearity error Note 1	ILE	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				±2.5	LSB
Differential non-linearityerror Note 1	DLE	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				±1.5	LSB
REF voltage(+)	AV_{REFP}			2.7		V_{DD}	V
REF voltage(-)	AV_{REFM}			V_{SS}			V
Analog input voltage	V_{AIN}			AV_{REFM}		AV_{REFP}	V
REF supply Current	I_{REF}	AV_{REFP} = 3 V			38	60	μA
		AV_{REFP} = 5 V			63	100	μA

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Minimum V_{DD}-0.5 V is allowed for AV_{REFP} to keep characteristic values

Remark When reference voltage(+) is not AV_{REFP} pin or reference voltage(-) is not AV_{REFM} pin, the accuracy will become worse.
Renesas recommends to use A/D converter with AV_{REFP} and AV_{REFM} though other reference can be functionally selected.

33.7.2 ZPD characteristics

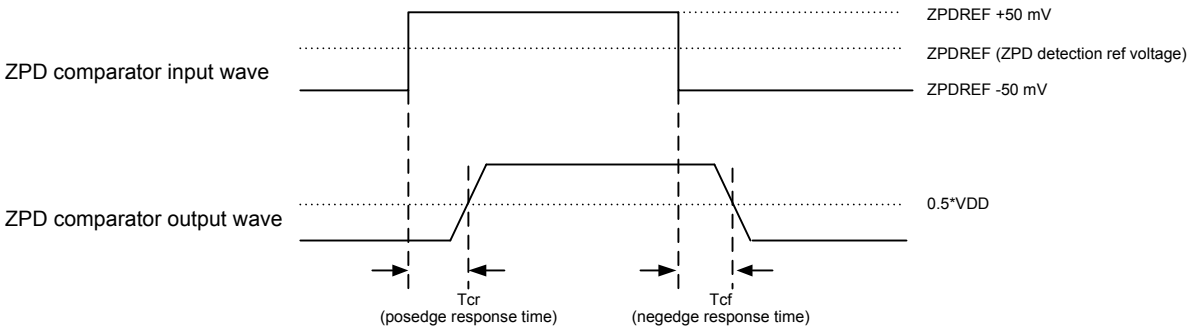
T_A = -40 to +85 °C,

<R> 2.7 V ≤ EV_{DD0} = EV_{DD1} = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Threshhold voltage	V_{ZPD}	0 Point detection voltage set = 000		6/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 001		10/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 010		14/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 011		18/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 100		22/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 101		9/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 110		11/200*SMV_{DD}±40 mv			V
Detection delay	T_{ZPDD}	100 mV Step, 50 mV Overdrive (refer to the below figure)	SMV_{DD} = 4.75 V to 5.25 V			100	ns
			SMV_{DD} = 2.7 V to 5.5 V			100	ns
Operation Stabilization wait time	T_{ZPDW}	Ref voltage Stabilization +ZPD comparator Stabilization				1+5 = 6	μs

Figure 33-15. ZPD timing

<R>



33.7.3 POR characteristics

T_A = -40 to +85 °C

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.45	1.51	1.57	V
	V_{PDR}		1.44	1.5	1.56	V
Detection delay	T_{PD}				300	μs
Minimum pulse width	T_{PW}	Necessary width of internal voltage drop down below V_{PDR}	300			μs

Caution LVD reset or external $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

33.7.4 LVD characteristics

T_A = -40 to +85 °C, $V_{PDR} \leq EV_{DD0} = EV_{DD1} \leq V_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V

	Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
	RESET and INTMODE	V_{LVI5}	VPOC0,1,2 = 0,1,1 Power down Reset Voltage: 2.7 V		2.70	2.75	2.81	V
		V_{LVI4}	LVIS0,1 = 1,0 (+0.1 V)	Power on Reset Release Voltage	2.86	2.92	2.97	V
				Power down Interrupt Voltage	2.80	2.86	2.91	V
		V_{LVI3}	LVIS0,1 = 0,1 (+0.2 V)	Power on Reset Release Voltage	2.96	3.02	3.08	V
				Power down Interrupt Voltage	2.90	2.96	3.02	V
		V_{LVI0}	LVIS0,1 = 0,0 (+1.2 V)	Power on Reset Release Voltage	3.98	4.06	4.14	V
				Power down Interrupt Voltage	3.90	3.98	4.06	V
<R>	Detection delay time	T_{LD}					300	μs
<R>	Minimum pulse width	T_{LW}	Necessary width of V_{DD} drop down below selected V_{LVIx} (x = 0, 3 to 5)		300			μs

Caution LVD reset or external $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

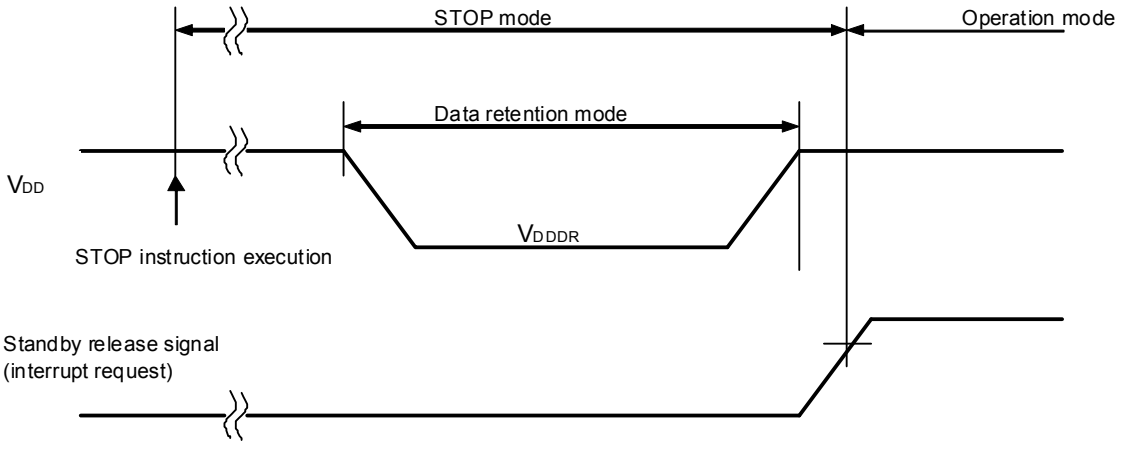
33.8 RAM Data Retention Characteristics

T_A = -40 to +85 °C

	Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
<R>	Data retention supply voltage	V_{DDDR}		1.44 Note1		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained until a POR reset is effected, but data is not retained when a POR reset is effected.

Figure 33-16. STOP Mode Data Retention timming



33.9 Capacitance Connected to REGC

T_A = -40 to +85 °C

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
Capacitance	C_{REG}		0.47		1.0	μF

33.10 Flash programming characteristics

<R>

T_A = -40 to +85 °C, 2.7 ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}	Programming current			12.2	mA
System Clcok frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	2		32	MHz
Number of Code Flash rewrites Notes 1, 2, 3	Cerwr	Retained for 20 years	1000			Times
Number of Data Flash rewrites Notes 1, 2, 3	Cerwr	Retained for 20 years	10000			Times

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite.
The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT)

Cautions 1. These specifications show target values, which may change after device evaluation.

2. The RL78/D1A 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.

Definition of Pin Groups

Definition of pin groups described in this chapter is shown in the following table.

<R>

Pin groups		Pin names				
		48-pin products	64-pin products	80-pin products	100-pin products	128-pin products
Pin group 1	Group 1R	P40	P40, P70, P71	P40, P70, P71	P40, P70, P71, P130 to P135, P140	P40 to P44, P70, P71, P100 to P103, P130 to P135, P140
	Group 1L	P00 to P01, P10 to P14, P30 to P33, P54 to P57, P60, P61, P72 to P75	P00 to P05, P07, P10 to P15, P17, P30 to P33, P54 to P57, P60, P61, P72 to P75	P00 to P07, P10 to P17, P30 to P37, P54 to P57,P60, P61, P65, P66, P72 to P75	P00 to P07, P10 to P17, P30 to P37, P50 to P57, P72 to P75, P136	P00 to P07, P10 to P17, P30 to P37, P45 to P47, P50 to P57, P72 to P75, P104 to P107, P110 to P117, P125 to P127 P136
	Group 1C	-	-	-	P60 to P66	P60 to P66
Pin group 2 (ANI pins)		P20 to P23, P27	P20 to P23, P27	P20 to P27	P20 to P27, P150	P20 to P27, P150 to P152
Pin group 3 (SMC pins)	Group 3A	P80 to P83	P80 to P83	P80 to P83	P80 to P83	P80 to P83
	Group 3B	-	P84 to P87	P84 to P87	P84 to P87	P84 to P87
	Group 3C	P90 to P94	P90 to P94	P90 to P93	P90 to P93	P90 to P93
	Group 3D	-	-	P94 to P97	P94 to P97	P94 to P97
	Group 3E	P90 to P94	P84 to P87, P90 to P94	-	-	-
Pin group 4 (System pins)		P121 to P122, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137	P121 to P124, $\overline{\text{RESET}}$, P137

Definition of Product Groups

Definition of product groups described in this chapter is shown in the following table.

<R>

Product groups	Product names				
	48-pin products	64-pin products	80-pin products	100-pin products	128-pin products
Product Group A	R5F10CGBLFB R5F10CGCLFB R5F10CGDLFB R5F10DGCLFB R5F10DGDLFB R5F10DGELFB	R5F10CLDLFB R5F10DLDLFB R5F10DLELFB	R5F10CMDLFB R5F10CMELFB R5F10DMDLFB R5F10DMELFB R5F10DMFLFB R5F10DMGLFB R5F10DMJLFB	R5F10DPELFB R5F10DPFLFB R5F10DPGLFB R5F10DPJLFB R5F10TPJLFB	-
Product Group B	-	-	-	R5F10DPKLFB R5F10DPLLFB	R5F10DSJLFB R5F10DSKLFB R5F10DSLLFB

34.1 Absolute Maximum Ratings

T_A = +25 °C

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}	V_{DD}	-0.5 to +6.5	V
	EV_{DD0} EV_{DD1}	$EV_{DD0} = EV_{DD1}$	-0.5 to +6.5 and -0.5 to V_{DD} + 0.3	V
	SMV_{DD0} SMV_{DD1}	$SMV_{DD0} = SMV_{DD1}$	-0.5 to +6.5 and -0.5 to V_{DD} + 0.3	V
	V_{SS}	V_{SS}	-0.5 to +0.3	V
	EV_{SS0} EV_{SS1}	$EV_{SS0} = EV_{SS1}$	-0.5 to +0.3	V
	SMV_{SS0} SMV_{SS1}	$SMV_{SS0} = SMV_{SS1}$	-0.5 to +0.3	V
Supply voltage up/down ramp	V_{DDRAMP}		≤50	V/ms
REGC pin input voltage	V_{IREGC}	REGC	-0.3 to +2.8 and -0.3 to V_{DD} + 0.3 Note1	V
Input voltage	V_{I1}	Pin group 1	-0.3 to +6.5 and -0.3 to EV_{DD0}(EV_{DD1}) + 0.3	V
	V_{I2}	Pin group 2	-0.3 to +6.5 and -0.3 to V_{DD} + 0.3	V
	V_{I3}	Pin group 3	-0.3 to +6.5 and -0.3 to SMV_{DD0}(SMV_{DD1}) + 0.3	V
	V_{I4}	Pin group 4	-0.3 to +6.5 and -0.3 to V_{DD} + 0.3	V

(Continue to next page)

T_A = +25 °C

Parameter	Symbols	Conditions			Ratings	Unit
Output voltage	V_{O1}	Pin group 1			-0.3 to EV_{DD0} (EV_{DD1}) + 0.3	V
	V_{O2}	Pin group 2			-0.3 to V_{DD} + 0.3	V
	V_{O3}	Pin group 3			-0.3 to SMV_{DD0} (SMV_{DD1}) + 0.3	V
	V_{COM}	COM0 to COM3			-0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH1}	Per pin	Pin group 1		-20	mA
		Total	Pin group 1		-150	mA
				Pin group 1L	-60	mA
				Pin group 1R	-55	mA
				Pin group 1C	-40	mA
	I_{OH2}	Per pin	Pin group 2		-0.5	mA
		Total			-2.0	mA
	I_{OH3}	Per pin	Pin group 3		-58	mA
		Total	Pin group 3	48-pin, 64-pin	-270	mA
<R>				80-pin, 100-pin, 128-pin	-480	mA
				Pin group 3A	-120	mA
				Pin group 3B	-120	mA
				Pin group 3C	-120	mA
				Pin group 3D	-120	mA
				Pin group 3E	-150	mA
	I_{OHCOM}	Per pin	COM0 to COM3		-0.5	mA
		Total	COM0 to COM3		-1.0	mA
Output current, low	I_{OL1}	Per pin	Pin group 1		20	mA
		Total	Pin group 1		150	mA
				Pin group 1L	60	mA
				Pin group 1R	50	mA
				Pin group 1C	40	mA
	I_{OL2}	Per pin	Pin group 2		1.0	mA
		Total			5.0	mA
	I_{OL3}	Per pin	Pin group 3		58	mA
		Total	Pin group 3	48-pin, 64-pin	270	mA
<R>				80-pin, 100-pin, 128-pin	480	mA
				Pin group 3A	120	mA
				Pin group 3B	120	mA
				Pin group 3C	120	mA
				Pin group 3D	120	mA
				Pin group 3E	150	mA
	I_{OLCOM}	Per pin	COM0 to COM3		0.5	mA
		Total	COM0 to COM3		1.0	mA
Operating ambient temperature	T_A	for normal operation mode			-40 to +105	°C
		for code flash programming			-40 to +105	°C
		for data flash programming			-40 to +105	°C
Storage temperature	T_{stg}				-65 to +150	°C

Note 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.

<R> 34.2 Power consumption characteristics

34.2.1 Product group A

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, run mode	I_{DD1} Note 1	High speed MAIN RUN Note 2, 3, 4	f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	4.2	18	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	3.8	16	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	2.1	11	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	1.6	9	mA
				f_X = 4 MHz			
		SUB RUN Note 2, 3, 5	f_{CLK} = f_{XT} = 32.768 kHz		6	500	μA

Notes 1. The common condition for I_{DD1}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The program is running in the code flash.

2. The typical value is that when Ta=+25deg.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. But the A/D converter, RTC, LCD circuit and stepper motor circuit are stopped, and the data flash and the code flash are stated to read mode. The 16-bit wakeup timer operates with f_{LOCO}.

4. In case of I_{DD1} of "f_X = 4/8 MHz with PLL", the high speed on-chip oscillator (HOCO) is stopped.

5. f_X and f_{HOCO} are stopped.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, halt mode	I_{DD2} Note 1	High speed MAIN HALT Note 3, 4, 5	f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	0.8	6.9	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	0.7	6.0	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	0.4	4.3	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	0.35	3.6	mA
				f_X = 4 MHz			
		SUB HALT Note 3, 4, 6	f_{CLK} = f_{XT} = 32.768 kHz	RTC is stopped	1.0	190	μA
				RTC is operated by f_{XT} = 32.768 kHz	1.2		
Supply current, stop mode	I_{DD3} Note 2	STOP Note 3, 4	RTC and f_{XT} are stopped		0.4	120	μA
			RTC is operated by f_{XT} = 32.768 kHz		0.8		

Notes 1. The common condition for I_{DD2}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The HALT instruction is executed by the program in the code flash.
- The specification shows the stable current during HALT mode.

2. The common condition for I_{DD3}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The STOP instruction is executed by the program in the code flash during MAIN RUN operation.
- The spec shows the stable current during STOP mode.

3. The typical value is that when Ta=+25deg.C and V_{DD}=5.0 V. Peripheral devices and the data flash are stopped.

4. The maximum value is that when all peripheral devices are operating. But the A/D converter, LCD circuit, stepper motor circuit, the data flash and the code flash are stopped. The 16-bit wakeup timer operates with f_{LOCO}.

5. Either f_X or f_{HOCO} which is selected for f_{CLK} is operated. The other is stopped.

6. f_X and f_{HOCO} are stopped.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Typ.	Max.	Unit
WDT operating current Note 1	I_{WDT}			0.25	2.0	μA
ADC operating current Note 2	I_{ADC}	Normal mode	V_{DD} = 5.0 V	1.3	1.7	mA
		Low voltage mode	V_{DD} = 3.0 V	0.5	0.7	mA
LCD operating Current Note 3	I_{LCD}	f_{LCD} = f_{SUB}, LCD clock = 512 Hz	V_{DD} = 5.0 V	100	210	μA
			V_{DD} = 3.0 V	90	200	μA
ZPD operating current Note 4	I_{ZPD}	One ZPD operated	V_{DD} = 5.0 V	150	600	μA
			V_{DD} = 3.0 V	100	500	μA
		Four ZPDs operated	V_{DD} = 5.0 V	500	2000	μA
			V_{DD} = 3.0 V	400	1600	μA

Notes
1. Current flowing only to the watchdog timer. The maximum specification of I_{DD1}, I_{DD2} and I_{DD3} include I_{WDT}.
2. Current flowing only to the A/D converter.The current value of the RL78/D1A is the sum of I_{DD} and I_{ADC} when the A/D converter operates.
3. Current flowing only to the LCD controller/driver circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{LCD} when the LCD controller/driver circuit operates.
4. Current flowing only to the ZPD circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{ZPD} when the ZPD circuit operates.

34.2.2 Product group B

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, run mode	I_{DD1} Note 1	High speed MAIN RUN Note 2, 3, 4, 5	f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	4.7	20	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	4.3	17.5	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	2.4	12	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	1.8	9.5	mA
				f_X = 4 MHz			
		SUB RUN Note 2, 3, 6	f_{CLK} = f_{XT} = 32.768 kHz		7	560	μA

Notes 1. The common condition for I_{DD1}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The program is running in the code flash.

2. The typical value is that when Ta=+25deg.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. But the A/D converter, RTC, LCD circuit and stepper motor circuit are stopped, and the data flash and the code flash are stated to read mode. The 16-bit wakeup timer operates with f_{LOCO}.

4. In case of I_{DD1} of "f_X = 4/8 MHz with PLL", the high speed on-chip oscillator (HOCO) is stopped.

5. At 128-pin products, the value of I_{DD1} does not include the LCDB (P11x, P46-7) pin toggle current.
I_{DD1} condition of LCDB macro is Fclk=24MHz, mod8 mode, data rate=6MHz, 4cycle, 16bit write/read.

6. f_X and f_{HOCO} are stopped.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions			Typ.	Max.	Unit
Supply current, halt mode	I_{DD2} Note 1	High speed MAIN HALT Note 3, 4, 5	f_{CLK} = 24 MHz	f_{HOCO} = 24 MHz	0.8	7.5	mA
				f_{HOCO} = 4 MHz with PLL			
				f_X = 4 MHz with PLL			
				f_X = 8 MHz with PLL			
			f_{CLK} = 20 MHz	f_X = 20 MHz	0.7	6.5	mA
			f_{CLK} = 8 MHz	f_{HOCO} = 8 MHz	0.4	4.5	mA
				f_X = 8 MHz			
			f_{CLK} = 4 MHz	f_{HOCO} = 4 MHz	0.35	3.8	mA
				f_X = 4 MHz			
		SUB HALT Note 3, 4, 6	f_{CLK} = f_{XT} = 32.768 kHz	RTC is stopped	1.0	200	μA
				RTC is operated by f_{XT} = 32.768 kHz	1.2		
Supply current, stop mode	I_{DD3} Note 2	STOP Note 3, 4	RTC and f_{XT} are stopped		0.4	130	μA
			RTC is operated by f_{XT} = 32.768 kHz		0.8		

Notes 1. The common condition for I_{DD2}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The HALT instruction is executed by the program in the code flash.
- The specification shows the stable current during HALT mode.

2. The common condition for I_{DD3}:
- I_{DD} includes the total current flowing into whole power pins, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
- The STOP instruction is executed by the program in the code flash during MAIN RUN operation.
- The spec shows the stable current during STOP mode.

3. The typical value is that when Ta=+25deg.C and V_{DD}=5.0 V. Peripheral devices and the data flash are stopped.

4. The maximum value is that when all peripheral devices are operating. But the A/D converter, LCD circuit, stepper motor circuit, the data flash and the code flash are stopped. The 16-bit wakeup timer operates with f_{LOCO}.

5. Either f_X or f_{HOCO} which is selected for f_{CLK} is operated. The other is stopped.

6. f_X and f_{HOCO} are stopped.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Typ.	Max.	Unit
WDT operating current Note 1	I_{WDT}			0.25	1.0	μA
ADC operating current Note 2	I_{ADC}	Normal mode	V_{DD} = 5.0 V	1.3	1.7	mA
		Low voltage mode	V_{DD} = 3.0 V	0.5	0.7	mA
LCD operating Current Note 3	I_{LCD}	f_{LCD} = f_{SUB}, LCD clock = 512 Hz	V_{DD} = 5.0 V	100	140	μA
			V_{DD} = 3.0 V	90	130	μA
ZPD operating current Note 4	I_{ZPD}	One ZPD operated	V_{DD} = 5.0 V	150	600	μA
			V_{DD} = 3.0 V	100	500	μA
		Four ZPDs operated	V_{DD} = 5.0 V	500	2000	μA
			V_{DD} = 3.0 V	400	1600	μA

Notes
1. Current flowing only to the watchdog timer. The maximum specification of I_{DD1}, I_{DD2} and I_{DD3} include I_{WDT}.
2. Current flowing only to the A/D converter.The current value of the RL78/D1A is the sum of I_{DD} and I_{ADC} when the A/D converter operates.
3. Current flowing only to the LCD controller/driver circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{LCD} when the LCD controller/driver circuit operates.
4. Current flowing only to the ZPD circuit. The current value of the RL78/D1A is the sum of I_{DD} and I_{ZPD} when the ZPD circuit operates.

34.3 Oscillator characteristics

34.3.1 Main(X1) oscillator characteristics

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Main(X1) clock oscillation frequency	f_X	Ceramic resonator	1.0		20.0	MHz
		Crystal resonator	1.0		20.0	MHz

Remark Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

34.3.2 High speed on chip oscillator characteristics

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
HOCO oscillation frequency	f_{HOCO}	4 MHz mode	3.88	4.00	4.12	MHz
		8 MHz mode	7.76	8.00	8.24	MHz
		16 MHz mode	15.52	16.00	16.48	MHz
		24 MHz mode	23.28	24.00	24.72	MHz

Remark Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

34.3.3 Low speed on chip oscillator characteristics

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
LOCO oscillation frequency	f_{LOCO}		12.75	15.0	17.25	kHz

34.3.4 Sub(XT1) oscillator characteristics

T_A = -40 to +105 °C, 2.7 V≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Sub(XT1) clock oscillation frequency	f_{XT}	Possible to oscillate	29.0	32.768	35.0	kHz

Remark Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

34.4 DC characteristics

34.4.1 Pin group 1

T_A = -40 to +105 °C, 4.0 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V (1/2)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Output current, high[Note 1]	I_{OH1}	Per pin				-5.0	mA
		I_{OH2}	Per pin, P73 or P135 (SG port)				-13.0	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note 2])	Group 1L			-40.0	mA
<R>				Group 1R			-40.0	mA
<R>				Group 1C (128-pin, 100-pin)			-30.0	mA
<R>			for 128-pin, 100-pin				-110.0	mA
			for 80-pin, 64-pin, 48-pin				-60.0	mA
	Output current, low	I_{OL1}	Per pin				8.5	mA
		I_{OL2}	Per pin, P73 or P135 (SG ports)				13.0	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note 3])	Group 1L			40.0	mA
<R>				Group 1R			35.0	mA
<R>				Group 1C (128-pin, 100-pin)			40.0	mA
<R>			for 128-pin, 100-pin				115.0	mA
			for 80-pin, 64-pin, 48-pin				60.0	mA

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

<R> **2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -30.0 mA

Total output current of pins = (-30.0 × 0.7)/(80 × 0.01) ≈ -26.2 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.

<R> **3.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 40.0 mA

Total output current of pins = (40.0 × 0.7)/(80 × 0.01) = 35.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

T_A = -40 to +105 °C, 4.0 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V (2/2)

	Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
	Input voltage, high Note 2	V_{IH1}	Schmitt3 mode	0.8EV_{DD}		EV_{DD}	V
		V_{IH2}	Schmitt1 mode Note 3	0.65EV_{DD}		EV_{DD}	V
	Input voltage, low Note 2	V_{IL1}	Schmitt3 mode	0		0.5EV_{DD}	V
		V_{IL2}	Schmitt1 mode Note 3	0		0.35EV_{DD}	V
	Input hysteresis width Note 2, 4	V_{IHYS1}	Schmitt3 mode	0.1	0.19	0.29	V
		V_{IHYS2}	Schmitt1 mode Note 3	0.15	0.59	0.84	V
	Output voltage, high Note 1	V_{OH1}	I_{OH} = -5.0 mA	EV_{DD}-1.0		EV_{DD}	V
			I_{OH} = -3.0 mA up to 6 pins	EV_{DD}-0.5		EV_{DD}	V
		V_{OH2}	I_{OH} = -13.0 mA, P73 or P135 (SG port)	EV_{DD}-0.7		EV_{DD}	V
	Output voltage, low	V_{OL1}	I_{OL} = 8.5 mA	0		0.7	V
			I_{OL} = 3.0 mA up to 6 pins	0		0.5	V
		V_{OL2}	I_{OL} = 13.0 mA, P73 or P135 (SG port)	0		0.7	V
<R>	Input leakage current, high	I_{LIH1}	V_I = EV_{DD}			1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = EV_{SS}			-1	μA
<R>	On chip pull-up resistance Note 5	R_U	V_I = EV_{SS}	10	20	100	kΩ
<R>	On chip pull-down resistance Note 6	R_D	V_I = EV_{DD}	100			kΩ

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

2. Except P130 because it is output only port.

<R> **3.** P01, P10, P11, P17, P31, P40, P50 to P52, P55 to P57, P61, P63, P70, P110 to P117, P135 only.

4. This value is defined by evaluation result.

5. Except P130 and P137. Pull-up resistance is connected by software when pin is set to input mode.

6. LCD segment shared pins only. Pull-down resistance is connected during reset.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

T_A = -40 to +105 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ 4.0 V, EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, VSS = EV_{SS0} = EV_{SS} = 0 V (1/2)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Output current, high[Note 1]	I_{OH1}	Per pin				-1.0	mA
		I_{OH2}	Per pin, P73 or P135 (SG port)				-7.5	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note 2])	Group 1L			-15.0	mA
				Group 1R			-30.0	mA
<R>				Group 1C (128-pin, 100-pin)			-7.0	mA
<R>				for 128-pin, 100-pin			-52.0	mA
				for 80-pin, 64-pin, 48-pin			-33.0	mA
<R>	Output current, low	I_{OL1}	Per pin				1.5	mA
		I_{OL2}	Per pin, P73 or P135 (SG ports)				7.0	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note 3])	Group 1L			18.0	mA
				Group 1R			30.0	mA
<R>				Group 1C (128-pin, 100-pin)			10.0	mA
<R>				for 128-pin, 100-pin			58.0	mA
				for 80-pin, 64-pin, 48-pin			35.0	mA

<R> **Notes 1.** When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

2. These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated <R> from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -7.0 mA

Total output current of pins = (-7.0 × 0.7)/(80 × 0.01) ≈ -6.1 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.

<R> **3.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

T_A = -40 to +105 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ 4.0 V, EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, VSS = EV_{SS0} = EV_{SS} = 0 V (2/2)

	Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
	Input voltage, high Note 2	V_{IH1}	Schmitt3 mode	0.8EV_{DD}		EV_{DD}	V
		V_{IH2}	Schmitt1 mode Note 3	0.7EV_{DD}		EV_{DD}	V
	Input voltage, low Note 2	V_{IL1}	Schmitt3 mode	0		0.4EV_{DD}	V
		V_{IL2}	Schmitt1 mode Note 3	0		0.3EV_{DD}	V
	Input hysterisis width Note 2, 4	V_{IHYS1}	Schmitt3 mode	0.05		0.21	V
		V_{IHYS2}	Schmitt1 mode Note 3	0.08		0.53	V
	Output voltage, high Note 1	V_{OH1}	I_{OH} = -1.0 mA	EV_{DD}-0.5		EV_{DD}	V
		V_{OH2}	I_{OH} = -7.5 mA, P73 or P135 (SG port)	EV_{DD}-0.7		EV_{DD}	V
	Output voltage, low	V_{OL1}	I_{OL} = 1.5 mA	0		0.5	V
		V_{OL2}	I_{OL}= 7.0 mA, P73 or P135 (SG port)	0		0.7	V
<R>	Input leakage current, high	I_{LIH1}	V_I = EV_{DD}			1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = EV_{SS}			-1	μA
<R>	On chip pull-up resistance Note 5	R_U	V_I = EV_{SS}	10	20	100	kΩ
<R>	On chip pull-down resistance Note 6	R_D	V_I = EV_{DD}	100			kΩ

Notes 1. When P60 or P61 is set to Nch open drain mode, it does not drive high level output.

2. Except P130 because it is output only port.

<R> **3.** P01, P10, P11, P17, P31, P40, P50 to P52, P55 to P57, P61, P63, P70, P110 to P117, P135 only.

4. This value is defined by evaluation result.

5. Except P130 and P137. Pull-up resistance is connected by software when pin is set to input mode.

6. LCD segment shared pins only. Pull-down resistance is connected during reset.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

34.4.2 Pin group 2 (ANI pins)

T_A = -40 to +105 °C, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Output current, high	I_{OH1}	Per pin			-0.1	mA
	$I_{OHTOTAL}$	Total			-0.8	mA
Output current, low	I_{OL1}	Per pin			0.4	mA
	$I_{OLTOTAL}$	Total			3.2	mA
Input voltage, high	V_{IH1}		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}		0		0.5V_{DD}	V
Input hysteresis width Note	V_{IHYS1}		0.1	0.19	0.29	V
Output voltage, high	V_{OH1}	I_{OH} = -0.1 mA	V_{DD}-0.5		V_{DD}	V
Output voltage, low	V_{OL1}	I_{OL} = 0.4 mA	0		0.4	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}			1	μA
Input leakage current, low	I_{LIL1}	V_I = V_{SS}			-1	μA

Note This specification is guaranteed by design. It is not tested when shipment.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 4.0 V, V_{SS} = 0 V

Parameter	Symbols	Conditions	Min.	Typ.	Max.	Unit
Output current, high	I_{OH1}	Per pin			-0.1	mA
	$I_{OHTOTAL}$	Total			-0.8	mA
Output current, low	I_{OL1}	Per pin			0.4	mA
	$I_{OLTOTAL}$	Total			3.2	mA
Input voltage, high	V_{IH1}		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}		0		0.4V_{DD}	V
Input hysteresis width Note	V_{IHYS1}		0.05		0.21	V
Output voltage, high	V_{OH1}	I_{OH} = -0.1 mA	V_{DD}-0.5		V_{DD}	V
Output voltage, low	V_{OL1}	I_{OL} = 0.4 mA	0		0.4	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}			1	μA
Input leakage current, low	I_{LIL1}	V_I = V_{SS}			-1	μA

Note This specification is guaranteed by design. It is not tested when shipment.

34.4.3 Pin group 3 (SMC pins)

T_A = -40 to +105 °C, 4.0 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (1/3)

	Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
	Output current, high	I_{OH1}	Per pin		T_A = -40 °C			-52	mA
					T_A = +25 °C			-39	mA
					T_A = +85 °C			-32	mA
					T_A = +105 °C			-32	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A	T_A = -40 °C			-118	mA
<R>					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
					T_A = +105 °C			-96	mA
				Group 3B	T_A = -40 °C			-118	mA
<R>					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
					T_A = +105 °C			-96	mA
				Group 3C	128-pin, 100-pin 80-pin	T_A = -40 °C			-118	mA
						T_A = +25 °C			-118	mA
						T_A = +85 °C			-96	mA
						T_A = +105 °C			-96	mA
					64-pin 48-pin	T_A = -40 °C			-118	mA
						T_A = +25 °C			-118	mA
						T_A = +85 °C			-96	mA
						T_A = +105 °C			-96	mA
				Group 3D (128-pin, 100-pin, 80-pin)		T_A = -40 °C			-118	mA
<R>						T_A = +25 °C			-118	mA
						T_A = +85 °C			-96	mA
						T_A = +105 °C			-96	mA
				Group 3E (64-pin, 48-pin)		T_A = -40 °C			-148	mA
						T_A = +25 °C			-118	mA
						T_A = +85 °C			-96	mA
						T_A = +105 °C			-96	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -118.0 mA

Total output current of pins = (-118.0 × 0.7)/(80 × 0.01) ≈ -103.2 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.

T_A = -40 to +105 °C, 4.0 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (2/3)

	Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
	Output current, low	I_{OL1}	Per pin		T_A = -40 °C			52	mA
					T_A = +25 °C			39	mA
					T_A = +85 °C			32	mA
					T_A = +105 °C			32	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A	T_A = -40 °C			118	mA
					T_A = +25 °C			118	mA
					T_A = +85 °C			96	mA
					T_A = +105 °C			96	mA
				Group 3B	T_A = -40 °C			118	mA
					T_A = +25 °C			118	mA
					T_A = +85 °C			96	mA
					T_A = +105 °C			96	mA
<R>				Group 3C 128-pin, 100-pin 80-pin	T_A = -40 °C			118	mA
					T_A = +25 °C			118	mA
					T_A = +85 °C			96	mA
					T_A = +105 °C			96	mA
				Group 3C 64-pin 48-pin	T_A = -40 °C			118	mA
					T_A = +25 °C			118	mA
					T_A = +85 °C			96	mA
					T_A = +105 °C			96	mA
<R>				Group 3D (128-pin, 100-pin, 80-pin)	T_A = -40 °C			118	mA
					T_A = +25 °C			118	mA
					T_A = +85 °C			96	mA
					T_A = +105 °C			96	mA
				Group 3E (64-pin, 48-pin)	T_A = -40 °C			-148	mA
					T_A = +25 °C			-118	mA
					T_A = +85 °C			-96	mA
					T_A = +105 °C			96	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 118.0 mA
Total output current of pins = (118.0 × 0.7)/(80 × 0.01) ≈ 103.2 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.

T_A = -40 to +105 °C, 4.0 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (3/3)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Input voltage, high	V_{IH1}			0.8SMV_{DD}		SMV_{DD}	V
	Input voltage, low	V_{IL1}			0		0.5SMV_{DD}	V
	Input hysteresis width Note 1	V_{IHYS1}			0.1	0.19	0.29	V
	Output voltage, high	V_{OH1}	T_A = -40 °C	I_{OH} = -52 mA	SMV_{DD} - 0.5		SMV_{DD}	V
			T_A = +25 °C	I_{OH} = -39 mA				
			T_A = +85 °C	I_{OH} = -32 mA				
			T_A = +105 °C	I_{OH} = -32 mA				
	Output voltage, low	V_{OL1}	T_A = -40 °C	I_{OL} = 52 mA	0		0.5	V
			T_A = +25 °C	I_{OL} = 39 mA				
			T_A = +85 °C	I_{OL} = 32 mA				
			T_A = +105 °C	I_{OL} = 32 mA				
	Output voltage deviation Note 2	V_{DEV}			0		50	mV
<R>	Input leakage current, high	I_{LIH1}	V_I = SMV_{DD}				1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = SMV_{SS}				-1	μA
<R>	On chip pull-up resistance Note 3	R_U	V_I = SMV_{SS}		10	20	100	kΩ
<R>	On chip pull-down resistance Note 4	R_D	V_I = SMV_{DD}		100			kΩ

Notes 1. This specification is guaranteed by design. It is not tested when shipment.

2. Output voltage deviation defines the difference of the outputs levels of the same stepper motor.
V_{DEV} = max ($|V_{OHx}-V_{OHy}|$, $|V_{OLx}-V_{OLy}|$) @I_{OHx} = I_{OHy}, I_{OLx} = I_{OLy}.
X and y denote any combination of two pins of the following pin groups: (P80-P83, P84-P87, P90-P93, P94-P97)

3. Pull-up resistance is connected by software when pin is set to input mode.

4. LCD segment shared pins only. Pull-down resistance is connected during reset.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 4.0 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (1/3)

	Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
	Output current, high	I_{OH1}	Per pin		T_A = -40 °C			-30	mA
					T_A = +25 °C			-25	mA
					T_A = +85 °C			-23	mA
					T_A = +105 °C			-22	mA
		$I_{OHTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A	T_A = -40 °C			-90	mA
<R>					T_A = +25 °C			-75	mA
					T_A = +85 °C			-69	mA
					T_A = +105 °C			-66	mA
				Group 3B	T_A = -40 °C			-90	mA
					T_A = +25 °C			-75	mA
					T_A = +85 °C			-69	mA
					T_A = +105 °C			-66	mA
<R>				Group 3C 128-pin, 100-pin 80-pin	T_A = -40 °C			-90	mA
					T_A = +25 °C			-75	mA
					T_A = +85 °C			-69	mA
					T_A = +105 °C			-66	mA
				Group 3C 64-pin 48-pin	T_A = -40 °C			-90	mA
					T_A = +25 °C			-75	mA
					T_A = +85 °C			-69	mA
					T_A = +105 °C			-66	mA
				Group 3D (128-pin, 100-pin, 80-pin)	T_A = -40 °C			-90	mA
<R>					T_A = +25 °C			-75	mA
					T_A = +85 °C			-69	mA
					T_A = +105 °C			-66	mA
				Group 3E (64-pin, 48-pin)	T_A = -40 °C			-90	mA
					T_A = +25 °C			-75	mA
					T_A = +85 °C			-69	mA
					T_A = +105 °C			-66	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = -75.0 mA

Total output current of pins = (-75.0 × 0.7)/(80 × 0.01) ≈ -65.6 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.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 4.0 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (2/3)

	Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
	Output current, low	I_{OL1}	Per pin		T_A = -40 °C			30	mA
					T_A = +25 °C			23	mA
					T_A = +85 °C			20	mA
					T_A = +105 °C			17	mA
		$I_{OLTOTAL}$	Total (for duty factors ≤ 70%[Note])	Group 3A	T_A = -40 °C			90	mA
					T_A = +25 °C			69	mA
					T_A = +85 °C			60	mA
					T_A = +105 °C			51	mA
				Group 3B	T_A = -40 °C			90	mA
					T_A = +25 °C			69	mA
					T_A = +85 °C			60	mA
					T_A = +105 °C			51	mA
<R>				Group 3C (128-pin, 100-pin 80-pin)	T_A = -40 °C			90	mA
					T_A = +25 °C			69	mA
					T_A = +85 °C			60	mA
					T_A = +105 °C			51	mA
				Group 3C (64-pin 48-pin)	T_A = -40 °C			90	mA
					T_A = +25 °C			69	mA
					T_A = +85 °C			60	mA
					T_A = +105 °C			51	mA
<R>				Group 3D (128-pin, 100-pin, 80-pin)	T_A = -40 °C			90	mA
					T_A = +25 °C			69	mA
					T_A = +85 °C			60	mA
					T_A = +105 °C			51	mA
				Group 3E (64-pin, 48-pin)	T_A = -40 °C			90	mA
					T_A = +25 °C			69	mA
					T_A = +85 °C			60	mA
					T_A = +105 °C			51	mA

<R> **Note** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OL} \times 0.7$)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 69.0 mA
Total output current of pins = (69.0 × 0.7)/(80 × 0.01) ≈ 60.3 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.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 4.0 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V (3/3)

	Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
	Input voltage, high	V_{IH1}			0.8SMV_{DD}		SMV_{DD}	V
	Input voltage, low	V_{IL1}			0		0.4SMV_{DD}	V
	Input hysteresis width Note 1	V_{IHYS1}			0.05		0.21	V
	Output voltage, high	V_{OH1}	T_A = -40 °C	I_{OH} = -30 mA	SMVDD -0.5		SMV_{DD}	V
			T_A = +25 °C	I_{OH} = -25 mA				
			T_A = +85 °C	I_{OH} = -23 mA				
			T_A = +105 °C	I_{OH} = -22 mA				
	Output voltage, low	V_{OL1}	T_A = -40 °C	I_{OL} = 30 mA	0		0.5	V
			T_A = +25 °C	I_{OL} = 23 mA				
			T_A = +85 °C	I_{OL} = 20 mA				
			T_A = +105 °C	I_{OL} = 17 mA				
	Output voltage deviation Note 2	V_{DEV}			0		50	mV
<R>	Input leakage current, high	I_{LIH1}	V_I = SMV_{DD}				1	μA
<R>	Input leakage current, low	I_{LIL1}	V_I = SMV_{SS}				-1	μA
<R>	On chip pull-up resistance Note 3	R_U	V_I = SMV_{SS}		10	20	100	kΩ
<R>	On chip pull-down resistance Note 4	R_D	V_I = SMV_{DD}		100			kΩ

Notes 1. This specification is guaranteed by design. It is not tested when shipment.

2. Output voltage deviation defines the difference of the outputs levels of the same stepper motor.
V_{DEV} = max ($|V_{OHx}-V_{OHy}|$, $|V_{OLx}-V_{OLy}|$) @I_{OHx} = I_{OHy}, I_{OLx} = I_{OLy}.
X and y denote any combination of two pins of the following pin groups: (P80-P83, P84-P87, P90-P93, P94-P97)

3. Pull-up resistance is connected by software when pin is set to input mode.

4. LCD segment shared pins only. Pull-down resistance is connected during reset.

34.4.4 Pin group 4 (OSC, reset and P137 pins)

T_A = -40 to +105 °C, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0.8V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$		0.65V_{DD}		V_{DD}	V
	V_{IH3}	P137		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0		0.2V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$		0		0.35V_{DD}	V
	V_{IL3}	P137		0		0.5V_{DD}	V
Input hysteresis width Note 2	V_{IHYS1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0.1	0.7		V
	V_{IHYS2}	$\overline{\text{RESET}}$		0.15	0.59	0.84	V
<R> Input leakage current, high	I_{LIH1}	P121, P122, P123, P124 $V_I = V_{DD}$	Port			1	μA
			EXCLK Note 1			1	μA
			OSC			10	μA
	I_{LIH2}	$\overline{\text{RESET}}$, $V_I = V_{DD}$				1	μA
	I_{LIH3}	P137, $V_I = V_{DD}$				1	μA
Input leakage current, low	I_{LIL1}	P121, P122, P123, P124 $V_I = V_{SS}$	Port			-1	μA
			EXCLK Note 1			-1	μA
			OSC			-10	μA
	I_{LIL2}	$\overline{\text{RESET}}$, $V_I = V_{SS}$				-1	μA
	I_{LIL3}	P137, $V_I = V_{SS}$				-1	μA

Notes 1. P122(EXCLK) only.

2. This specification is guaranteed by design. It is not tested when shipment.

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 4.0 V, V_{SS} = 0 V

Parameter	Symbols	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P121,P122,P123,P124 (Port or EXCLK Note 1)		0.8V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$		0.7V_{DD}		V_{DD}	V
	V_{IH3}	P137		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0		0.2V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$		0		0.3V_{DD}	V
	V_{IL3}	P137		0		0.4V_{DD}	V
Input hysteresis width Note 2	V_{IHYS1}	P121, P122, P123, P124 (Port or EXCLK Note 1)		0.08			V
<R>	V_{IHYS2}	$\overline{\text{RESET}}$		0.08		0.53	V
Input leakage current, high	I_{LIH1}	P121, P122, P123, P124 V_I = V_{DD}	Port			1	μA
			EXCLK Note 1			1	μA
			OSC			10	μA
	I_{LIH2}	$\overline{\text{RESET}}$, V_I = V_{DD}				1	μA
	I_{LIH3}	P137, V_I = V_{DD}				1	μA
Input leakage current, low	I_{LIL1}	P121, P122, P123, P124 V_I = V_{SS}	Port			-1	μA
			EXCLK Note 1			-1	μA
			OSC			-10	μA
	I_{LIL2}	$\overline{\text{RESET}}$, V_I = V_{SS}				-1	μA
	I_{LIL3}	P137, V_I = V_{SS}				-1	μA

Notes 1. P122(EXCLK) only.

2. This specification is guaranteed by design. It is not tested when shipment.

34.5 AC characteristics

34.5.1 Basic operation

T_A = -40 to +105 °C,

<R> 2.7 V ≤ = EV_{DD0} = EV_{DD1} ≤ V_{DD} = SMV_{DD0} = SMV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Parameter	Symbols	Conditions			Min.	Typ.	Max.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock operation (Including PLL)	High speed main run		0.04166 Note 1		1	μs
		Sub system clock operation	SDIV = 0		1/f_{XT} (Typ. 30.5)			μs
External main system clock frequency	f_{EX}	Square wave input to EXCLK			2		20	MHz
External main system clock (square wave) input high/low level width	t_{EXH} t_{EXL}	Square wave input to EXCLK			24			ns
Timer input high/low level width	t_{TIH} t_{TIL}	TI00 to TI07, TI10 to TI17, TI20 to TI27			1/f_{MCK}+10 Note 2			ns
<R> Port output frequency	f_{GPO}	P80 to P87, P90 to P97	C = 30 pF	4.0 V ≤ SMV_{DD}			2	MHz
<R>		P20 to P27, P150 to P152	C = 30 pF	4.0 V ≤ V_{DD}			2	MHz
<R>		P73,P135	C = 30 pF	4.0 V ≤ EV_{DD}			8	MHz
<R>		Other than the above	C = 30 pF	4.0 V ≤ EV_{DD}			16	MHz
<R>				2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
External interrupt input high/low level width Note 3	t_{INIH} t_{INIL}	INTP0 to INTP5, INTPLR0, INTPLR1			1			μs
$\overline{RESET}$ input low level width Note 3	t_{RSL}	$\overline{RESET}$			10			μs
Analog noise filter rejection pulse width Note 4	t_{WRJ}	INTP0 to INTP5, INTPLR0, INTPLR1, ADTRG			30	50	1000	ns
ADTRG input high level width	t_{ATH}	Without noise filter		AWC=0	1/f_{CLK}+10			ns
				AWC=1	10			ns
		With noise filter		AWC=0	1/f_{CLK}+10 or t_{WRJ} (**Note 5**)			ns
				AWC=1	t_{WRJ} (**Note 5**)			ns

Notes 1. Value is in case of f_{CLK} is 24.0 MHz. It is also allowed to exceed frequency up to +3%.

2. f_{MCK} shows the frequency value of operation clock for TAU. Usually, f_{MCK} is defined by MHz but this specification is defined by ns. It is not defined by μs, so please be careful.

3. Pulses longer than this value will pass the input filter.

4. Pulses shorter than this value do not pass the input filters.

5. If the value of "1/f_{CLK}+10 [ns]" is less than tWRJ, please use t_{WRJ} value instead of "1/f_{CLK} +10 [ns]".

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 34-1. AC Timing Test Points

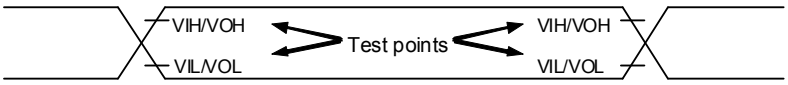


Figure 34-2. External Main System Clock Timing

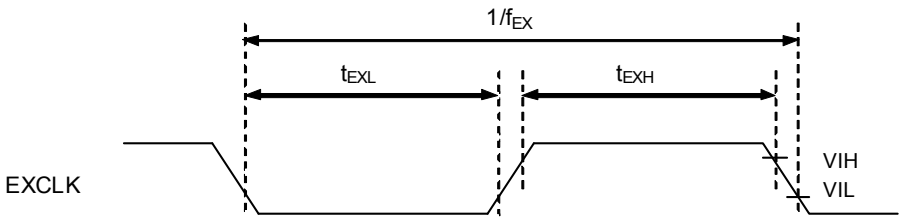


Figure 34-3. TI Timing

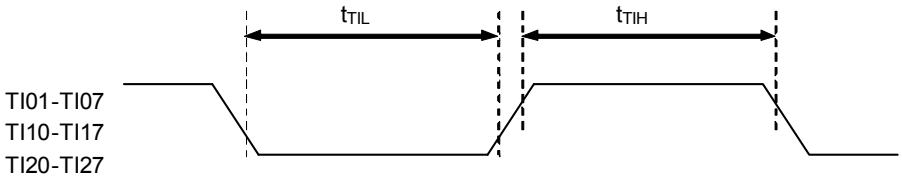


Figure 34-4. Interrupt Request Input Timing

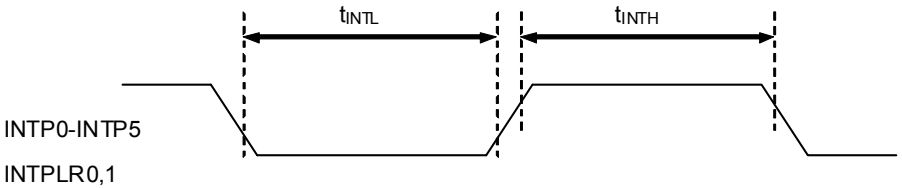
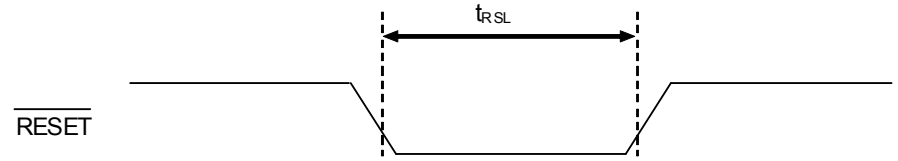


Figure 34-5. $\overline{\text{RESET}}$ Input Timing



34.5.2 Stepper motor controller/driver

T_A = -40 to +105 °C,

<R> 2.7 V ≤ = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Meter Controller/Driver input frequency	f_{MC} Note 1					24	MHz
PWM output rise time	t_R	10%-90% Note 2	4.0 V ≤ SMV_{DD} ≤ 5.5 V	15	60	100	ns
<R>			2.7 V ≤ SMV_{DD} ≤ 4.0 V	20		500	ns
PWM output fall time	t_F	10%-90% Note 2	4.0 V ≤ SMV_{DD} 5.5 V	15	60	100	ns
<R>			2.7 V ≤ SMV_{DD} ≤ 4.0 V	20		500	ns
Peak Cross Current Note 3	I_{CROSS}					50	ns
Output Pulse Width Note 4	t_{MO}	4.0 V ≤ SMV_{DD} ≤ 5.5 V		250			ns
		2.7 V ≤ SMV_{DD} ≤ 5.5 V		5000			ns
Output Pulse Length Deviation Note 5	t_{SMDEV}	4.0 V ≤ SMV_{DD} ≤ 5.5 V		-65	-12	+10	ns
		2.7 V ≤ SMV_{DD} ≤ 5.5 V		-100		+400	ns
Symmetry performance Note 6	ΔHSPmn	IOH = -32 mA ΔHSPmn = | VOH[(SMmn)max - (SMmn)min] |	2.7 V ≤ SMV_{DD} ≤ 5.5 V			50	mV
	ΔHSPmn	IOL = 32 mA ΔHSPmn = | VOL[(SMmn)max - (SMmn)min] |	4.0 V ≤ SMV_{DD} ≤ 5.5 V			50	mV
			2.7 V ≤ SMV_{DD} ≤ 5.5 V			100	mV

Notes 1. Source clock of the free-running counter.

2. t_R,t_F is not tested in production, specified by design.

3. The slew rate control generates a cross current in the output stage to control the energy of the external inductive load. The cross current flows only during the output transition time t_R, t_F. It flows in addtion to the output current. The cross current is not tested,but derived from simulation.

4. The output buffer can not generate high or low pulses shorter than this time, because of its slew rate control system. This value is not tested,but derived from simulation.

5. The slew rate control function causes a deviation of output pulse time compared to the ideal selected output pulse setting. This value is not tested,but derived from simulation.

6. Indicates the dispersion of 16 PWM output voltages.(4 buffers' output voltage differences in the state of I_{OH}(I_{OL}) at the same time.) Not tested in production, specified by design.

Remark m = 1 to 4, n = 1 to 4

<R> 34.5.3 Sound generator

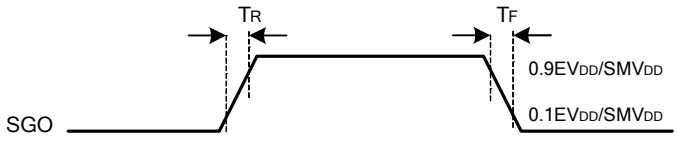
T_A = -40 to +105 °C,

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Sound generator input frequency	f_{SG}					24	MHz
SGO output rise time	f_R	C = 100 pF	P73, P135			200	ns
			P93			500	
SGO output fall time	f_F	C = 100 pF	P73, P135			200	ns
			P93			500	

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Sound Generator Output Timing



34.5.4 Serial interface: CSI operation

<R> <Master mode>

T_A = -40 to +105 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	Min.		Max.	Unit
SCK cycle time	t_{KCY1}	4.0≤V_{DD}	167	t_{KCY1}≥4/f_{CLK} Note		ns
		V_{DD}<4.0 V	250			ns
SCK high/low level width	t_{KH1}	4.0≤V_{DD}	t_{KCY1}/2-12			ns
	t_{KL1}	V_{DD}<4.0 V	t_{KCY1}/2-18			ns
SI set up time	t_{SIK1}	4.0≤V_{DD}	44			ns
		V_{DD}<4.0 V	55			Ns
SI hold time	t_{KSI1}		19			ns
SO output delay time	t_{KSO1}	C = 30 pF			25	ns

Note When CSI transfer is operated by DMA, it is necessary to concider DMA response time to decide cycle time.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

<R> <Slave mode>

T_A = -40 to +105 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions		Min.	Max.	Unit
SCK cycle time	t_{KCY2}	4.0≤EV_{DD}	20 MHz<f_{MCK}	8/f_{MCK}		ns
			f_{MCK}≤20 MHz	6/f_{MCK}		ns
		EV_{DD}<4.0 V	16 MHz<f_{MCK}	8/f_{MCK}		ns
			f_{MCK}≤16 MHz	6/f_{MCK}		ns
SCK high/low level width	t_{KH2}			t_{KCY2}/2		ns
	t_{KL2}					
SI set up time	t_{SIK2}	2.7≤EV_{DD}		1/f_{MCK}+40		ns
SI hold time	t_{KSI2}			1/f_{MCK}+62		ns
SO output delay time	t_{KSO2}	C = 30pF	4.0≤EV_{DD}		2/f_{MCK}+44	ns
			EV_{DD}<4.0 V		2/f_{MCK}+57	ns

Note When CSI transfer is operated by DMA, it is necessary to concider DMA response time to decide cycle time.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 34-6. CSI mode connection diagram

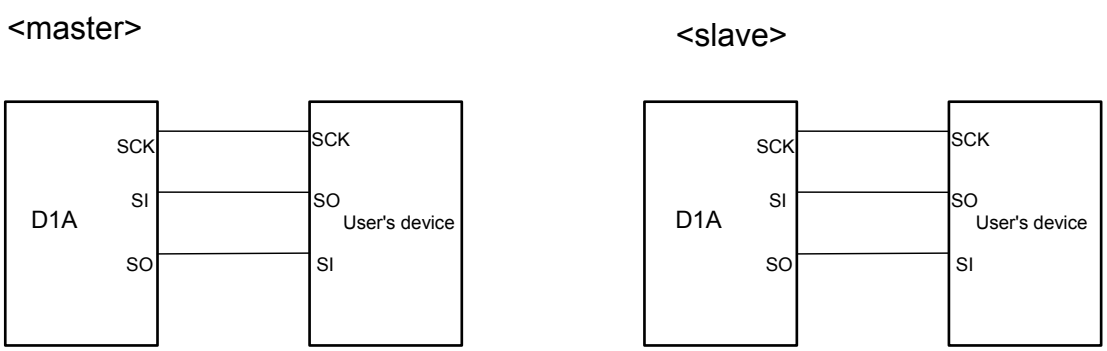


Figure 34-7. CSI mode serial transfer timing (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)

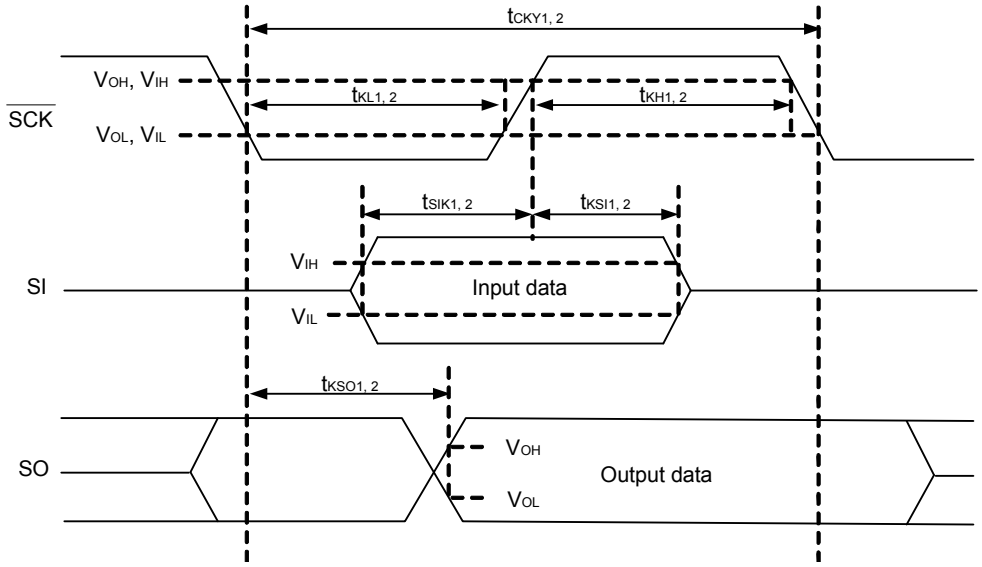
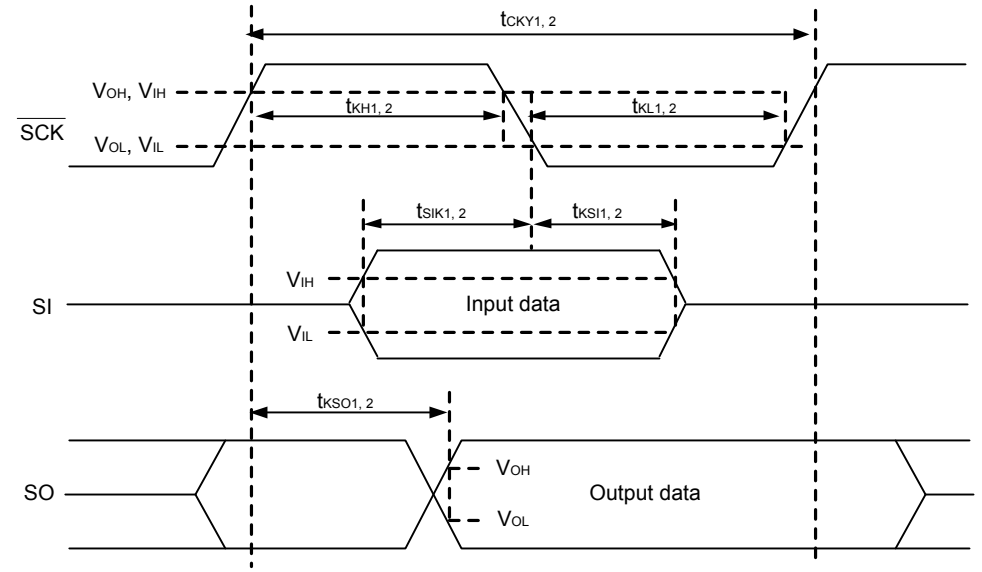


Figure 34-8. CSI mode serial transfer timing (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



34.5.5 Serial interface: UART operation (128-pin only)

T_A = -40 to +105 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} = ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Item	Symbol	Conditions	Min.	Max.	Unit
Transfer rate	T			$f_{MCK}/12$ Note	bps
		Theoretical value of the maximum transfer rate f_{CLK} = 24 MHz, f_{MCK} = f_{CLK}		2	Mbps

Note When CSI transfer is operated by DMA, it is necessary to concider DMA response time to decide cycle time.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 34-9 UART connection diagram

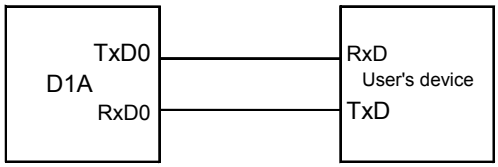
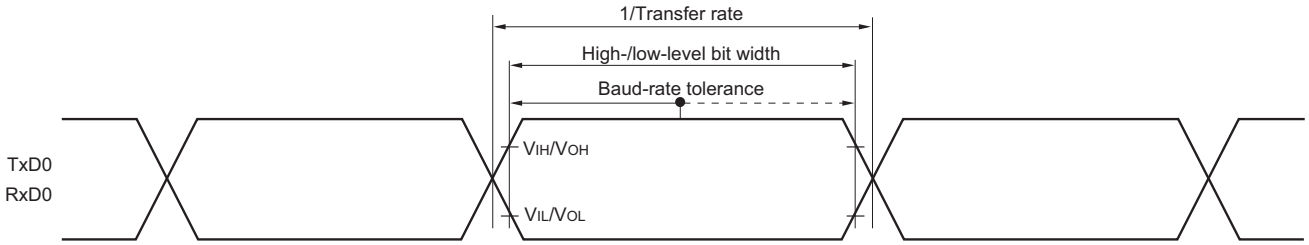


Figure 34-10. UART mode bit width (reference)



34.5.6 Serial interface: simplified I²C operation

T_A = -40 to +105 °C

<R> 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} = ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	Min.	Max.	Unit
SCL clock frequency	f_{SCL}	Rb = 3 kΩ, Cb = 100 pF		400	kHz
Hold time during SCL = ”L”	t_{LOW}	Rb = 3 kΩ, Cb = 100 pF	1150		ns
Hold time during SCL = ”H”	t_{HIGH}	Rb = 3kΩ, Cb = 100 pF	1150		ns
<R> Data set up time (reception)	$t_{SU;DAT}$	Rb = 3 kΩ, Cb = 100 pF	1/f_{MCK}+270		ns
Data hold time (transmission)	$t_{HD;DAT}$	Rb = 3 kΩ, Cb = 100 pF	0	355	ns

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 34-11 simplified I I²C connection diagram

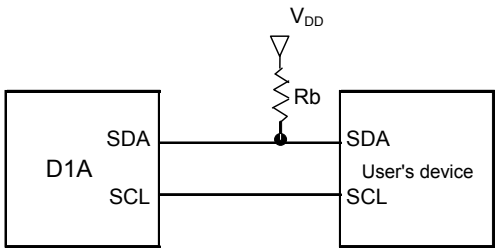
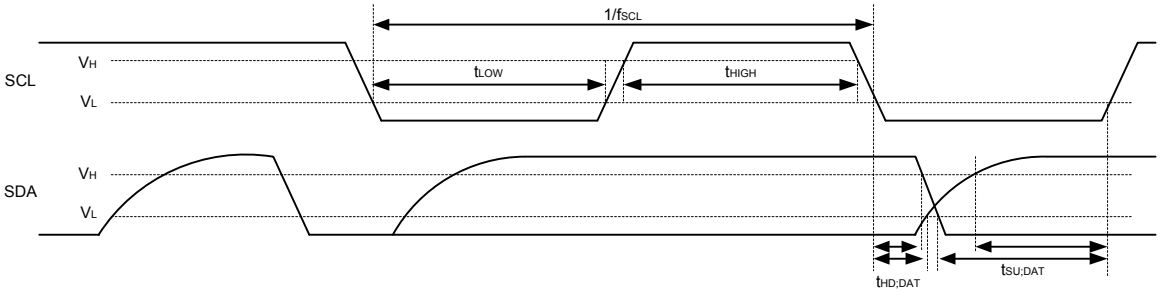


Figure 34-12. Simplified I²C mode serial transfer timing



34.5.7 Serial interface: LIN-UART(UARTF) operation

T_A = -40 to +105 °C

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	Min.	Max.	Unit
Transfer rate	T			1.0	Mbps

<R> **Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.**

34.5.8 Serial interface: CAN operation

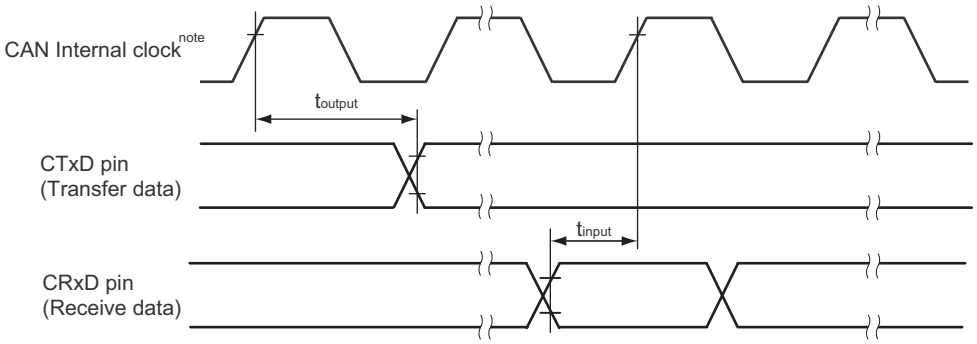
T_A = -40 to +105 °C,

2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	Min.	Max.	Unit
Transfer rate	T			1.0	Mbps
Internal delay time	t_{NODE}			100	ns
CRxD minimum pulse width for wake up	t_{CRXW}	Necessary width to detect wakeup signal	200		ns

<R> **Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.**

Figure 34-11. Internal delay time of CAN



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

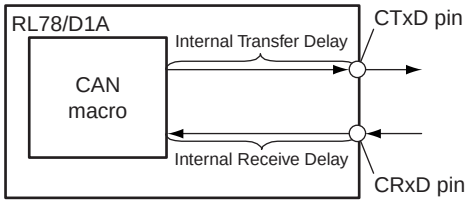


Image figure of internal delay

<R> 34.6 LCD Bus Interface characteristics (128-pin products only)

T_A = -40 to +105 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} =EV_{SS1} =0 V

Items	Symbols	Conditions	MIN.	MAX.	Unit
Transfer frequency	F			8	MHz
CycleTime	t_{CYC}		LBCYC x T		ns
Control low pulse width	t_{CL}		(LBWST + 1)T - 1		ns
Enable active pulse width	t_{ELH}		(LBWST + 1)T - 5		ns
Control setup time	t_{RWS}		$0.5T_S$ - 8		ns
Control hold time	t_{RWH}		0.5T - 3		ns
Data output setup time	t_{DOS}		$0.5T_S$ - 6	$0.5T_S$ + 17	ns
Data output hold time	t_{DOH}		{LBCYC - (LBWST + 1.5)} T - 27		ns
Data input setup time	t_{DIS}		50		ns
Data input hold time	t_{DIH}		0		ns
Output disable time	t_{OD}		0.5T - 14		ns

Remarks 1. T = (1/f_{CLK}) × n (n: LCD Bus Interface clock n divider setting)

2. T_S = (1/f_{CLK}) × N (N: LCD Bus Interface no clock divider N = 1, n divider N = n - 1)

3. F = 1/t_{CYC}

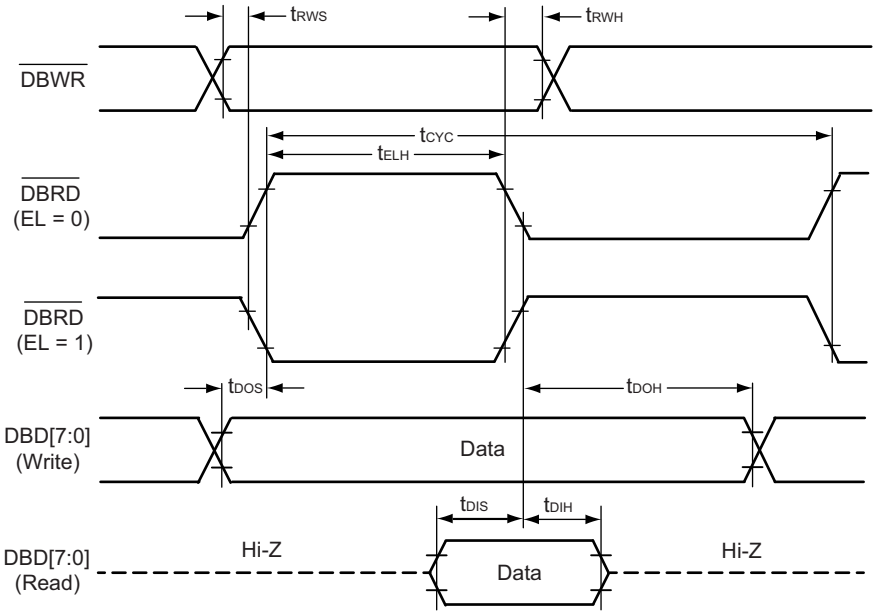
4. When EV_{DDx} = SMV_{DD}x ≤ V_{DD}, LCD controller/driver related registers must be initial value (LCDON = 0, SCOC = 0, MDSET1 – 0 = 00, LCDPFx = 0).

5. The above table shows the timing of LCD bus interface with Schmitt1 input characteristic.

Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

Figure 34-14. LCD Bus Interface AC timing (1/2)

(a) LCDB mod68 mode timing



(b) mode68 turnaround timing

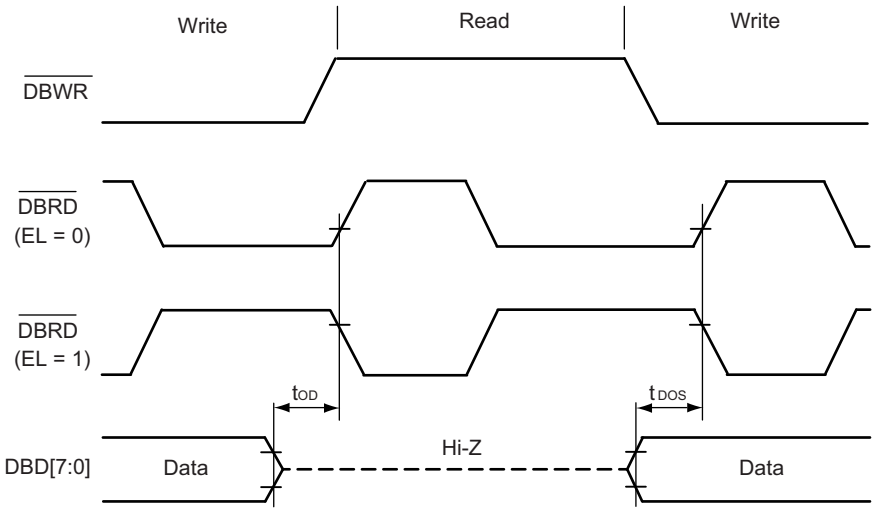
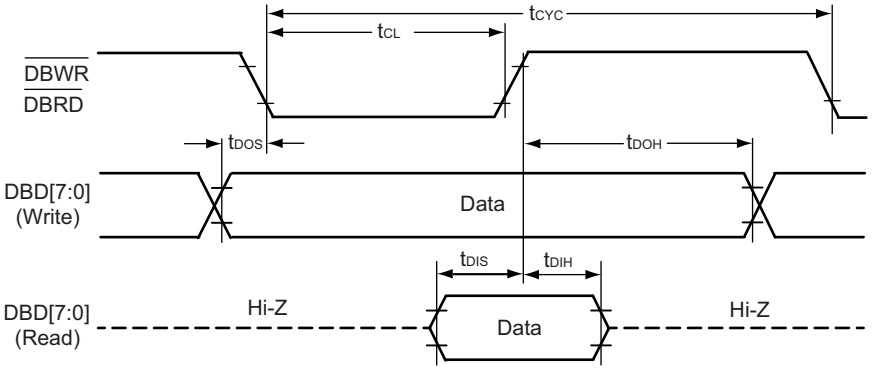
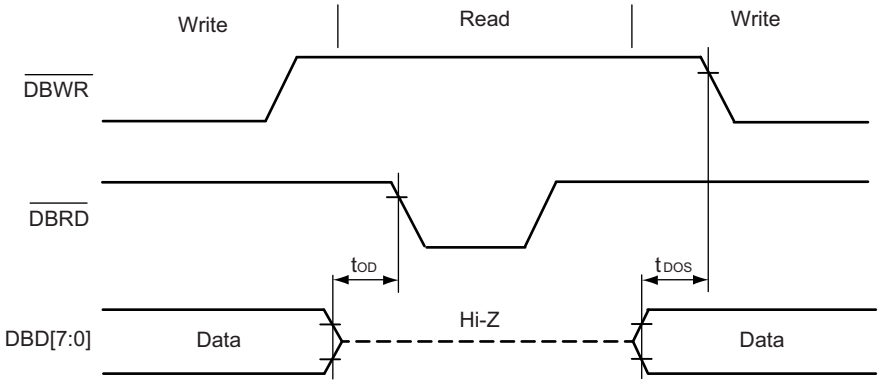


Figure 34-14. LCD Bus Interface AC timing (2/2)

(c) LCDB mod80 mode timing



(d) Mode80 turnaround timing



34.6 LCD characteristics

<R> T_A = -40 to +105 °C, 3.2 V ≤ EV_{DD0} = EV_{DD1} = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

	Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
	LCD division resistance Note 1	R_{LCD}				3	kΩ
	LCD Segment output voltage (unloaded)	V_{ODS}	I_O = ±1 μA	V_{LCDn}-0.05	V_{LCDn} Note 2	V_{LCDn}+0.05	V
	LCD Common output voltage (unloaded)	V_{ODC}	I_O = ±1 μA	V_{LCDn}-0.05	V_{LCDn}	V_{LCDn}+0.05	V
<R>	LCD Segment Output Voltage (loaded)	V_{ODSL0}	I_O = ±10 μA, all segment pins at same time	V_{LCD0}-0.6	V_{LCD0}	V_{LCD0}+0.6	V
<R>		V_{ODSL1}	I_O = ±10 μA, all segment pins at same time	V_{LCD1}-0.6	V_{LCD1}	V_{LCD1}+0.6	V
<R>		V_{ODSL2}	I_O = ±10 μA, all segment pins at same time	V_{LCD2}-0.6	V_{LCD2}	V_{LCD2}+0.6	V
<R>		V_{ODSL3}	I_O = ±10 μA, all segment pins at same time	V_{LCD3}-0.6	V_{LCD3}	V_{LCD3}+0.6	V
	LCD Common Output Voltage (loaded)	V_{ODCL0}	I_O = ±40 μA, single pin	V_{LCD0}-0.2	V_{LCD0}	V_{LCD0}+0.2	V
		V_{ODCL1}	I_O = ±40 μA, single pin	V_{LCD1}-0.2	V_{LCD1}	V_{LCD1}+0.2	V
		V_{ODCL2}	I_O = ±40 μA, single pin	V_{LCD2}-0.2	V_{LCD2}	V_{LCD2}+0.2	V
		V_{ODCL3}	I_O = ±40 μA, single pin	V_{LCD3}-0.2	V_{LCD3}	V_{LCD3}+0.2	V
	LCD split voltage drive capability Note 1	V_{LC0}	I_O = ±530 μA	V_{LCD0}-0.1	V_{LCD0}	V_{LCD0}+0.1	V
		V_{LC1}	I_O = ±530 μA	V_{LCD1}-0.1	V_{LCD1}	V_{LCD1}+0.1	V
		V_{LC2}	I_O = ±530 μA	V_{LCD2}-0.1	V_{LCD2}	V_{LCD2}+0.1	V
		V_{LC3}	I_O = ±530 μA	V_{LCD3}-0.1	V_{LCD3}	V_{LCD3}+0.1	V
	LCD output resistance (COM) Note 3	R_{ODC}				8	kΩ
	LCD output resistance (SEG) Note 3	R_{ODS}				8	kΩ

Notes 1. Only internal connection. The vaule is design specification.

2. V_{LCDn} (n= 0..3) represents one of the four possible voltage levels at the LCD pins. See table below for reference.

V_{LCDn}	no step-down transforming	step-down transforming
V_{LCD0}	V_{DD}	3/5 V_{DD}
V_{LCD1}	2/3 V_{DD}	2/5 V_{DD}
V_{LCD2}	1/3 V_{DD}	1/5 V_{DD}
V_{LCD3}	V_{SS}	V_{SS}

3. RODC is internal equivalent weight resistance from COM pin + COM IOBUF resistance.
RODS is internal equivalent weight resistance from SEG pin +SEG IOBUF resistance.

T_A = -40 to +105 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
LCD division resistance Note 1	R_{LCD}				3	kΩ
LCD Segment output voltage (unloaded)	V_{ODS}	Io = ±1 μA	V_{LCDn}-0.05	V_{LCDn} Note 2	V_{LCDn}+0.05	V
LCD Common output voltage (unloaded)	V_{ODC}	Io = ±1 μA	V_{LCDn}-0.05	V_{LCDn}	V_{LCDn}+0.05	V
LCD Segment Output Voltage (loaded)	V_{ODSL0}	Io = ±5 μA, all segment pins at same time	V_{LCD0}-0.6	V_{LCD0}	V_{LCD0}+0.6	V
	V_{ODSL1}	Io = ±5 μA, all segment pins at same time	V_{LCD1}-0.6	V_{LCD1}	V_{LCD1}+0.6	V
	V_{ODSL2}	Io = ±5 μA, all segment pins at same time	V_{LCD2}-0.6	V_{LCD2}	V_{LCD2}+0.6	V
	V_{ODSL3}	Io = ±5 μA, all segment pins at same time	V_{LCD3}-0.6	V_{LCD3}	V_{LCD3}+0.6	V
LCD Common Output Voltage (loaded)	V_{ODCL0}	Io = ±25 μA, single pin	V_{LCD0}-0.2	V_{LCD0}	V_{LCD0}+0.2	V
	V_{ODCL1}	Io = ±25 μA, single pin	V_{LCD1}-0.2	V_{LCD1}	V_{LCD1}+0.2	V
	V_{ODCL2}	Io = ±25 μA, single pin	V_{LCD2}-0.2	V_{LCD2}	V_{LCD2}+0.2	V
	V_{ODCL3}	Io = ±25 μA, single pin	V_{LCD3}-0.2	V_{LCD3}	V_{LCD3}+0.2	V
LCD split voltage drive capability	V_{LC0}	Io = ±530 μA	V_{LCD0}-0.1	V_{LCD0}	V_{LCD0}+0.1	V
	V_{LC1}	Io = ±530 μA	V_{LCD1}-0.1	V_{LCD1}	V_{LCD1}+0.1	V
	V_{LC2}	Io = ±530 μA	V_{LCD2}-0.1	V_{LCD2}	V_{LCD2}+0.1	V
	V_{LC3}	Io = ±530 μA	V_{LCD3}-0.1	V_{LCD3}	V_{LCD3}+0.1	V
LCD output resistance (COM) Note 3	R_{ODC}				10	kΩ
LCD output resistance (SEG) Note 3	R_{ODS}				10	kΩ

Notes 1. V_{LCDn} (n= 0..3) represents one of the four possible voltage levels at the LCD pins. See table below for reference.

V_{LCDn}	no step-down transforming
V_{LCD0}	V_{DD}
V_{LCD1}	2/3 V_{DD}
V_{LCD2}	1/3 V_{DD}
V_{LCD3}	V_{SS}

2. Only internal connection. The vaule is design specification.

3. RODC is internal equivalent weight resistance from COM pin + COM IOBUF resistance.
RODS is internal equivalent weight resistance from SEG pin +SEG IOBUF resistance.

34.7 Analog characteristics

34.7.1 A/D converter characteristics

<R>

T_A = -40 to +105 °C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} =0 V
Reference voltage (+) = AV_{REFP}, Reference voltage(–) = AV_{REFM}

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	Bit
Overall error Note 1,2	AINL	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V			1.2	±3.5	LSB
Conversion time	Tconv	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V	3.6 V≤V_{DD}<5.5 V	2.125		39	μs
			2.7 V≤V_{DD}<5.5 V	3.1875		39	μs
<R> Zero-scale error Note 1, 2	E_{ZS}	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				0.25	%FSR
<R> Full-scale error Note 1, 2	E_{FS}	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				0.25	%FSR
Integral non-linearity error Note 1	ILE	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				±2.5	LSB
Differential non-linearityerror Note 1	DLE	10 bit Resolution AV_{REFP} = V_{DD} Note 3 AV_{REFM} = 0 V				±1.5	LSB
REF voltage(+)	AV_{REFP}			2.7		V_{DD}	V
REF voltage(-)	AV_{REFM}			V_{SS}			V
Analog input voltage	V_{AIN}			AV_{REFM}		AV_{REFP}	V
REF supply Current	I_{REF}	AV_{REFP} = 3 V			38	60	μA
		AV_{REFP} = 5 V			63	100	μA

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Minimum V_{DD}-0.5 V is allowed for AV_{REFP} to keep characteristic values

Remark When reference voltage(+) is not AV_{REFP} pin or reference voltage(-) is not AV_{REFM} pin, the accuracy will become worse.
Renesas recommends to use A/D converter with AV_{REFP} and AV_{REFM} though other reference can be functionally selected.

34.7.2 ZPD characteristics

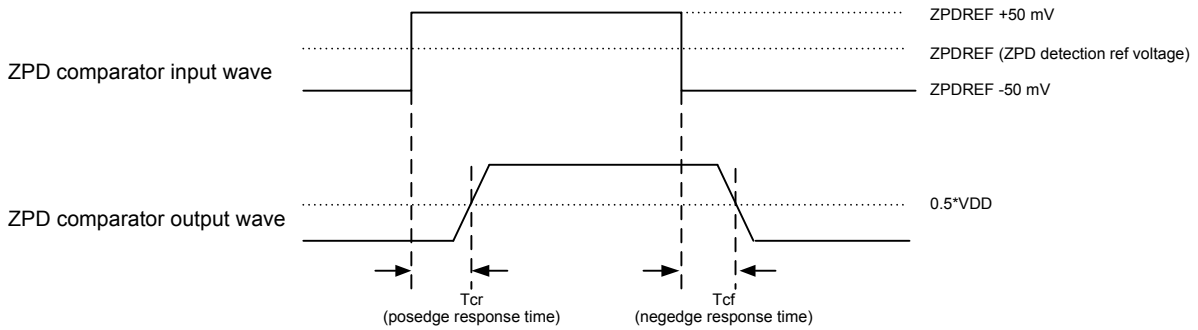
T_A = -40 to +105 °C,

<R> 2.7 V ≤ EV_{DD0} = EV_{DD1} = SMV_{DD0} = SMV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = SMV_{SS0} = SMV_{SS1} = 0 V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
Threshhold voltage	V_{ZPD}	0 Point detection voltage set = 000		6/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 001		10/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 010		14/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 011		18/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 100		22/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 101		9/200*SMV_{DD}±40 mv			V
		0 Point detection voltage set = 110		11/200*SMV_{DD}±40 mv			V
Detection delay	T_{ZPDD}	100 mV Step, 50 mV Overdrive (refer to the below figure)	SMV_{DD} = 4.75 V to 5.25 V			100	ns
			SMV_{DD} = 2.7 V to 5.5 V			100	ns
Operation Stabilization wait time	T_{ZPDW}	Ref voltage Stabilization +ZPD comparator Stabilization				1+5 = 6	μs

Figure 34-15. ZPD timing

<R>



34.7.3 POR characteristics

T_A = -40 to +105 °C

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.45	1.51	1.57	V
	V_{PDR}		1.44	1.5	1.56	V
Detection delay	T_{PD}				300	μs
Minimum pulse width	T_{PW}	Necessary width of internal voltage drop down below V_{PDR}	300			μs

Caution LVD reset or external $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

34.7.4 LVD characteristics

T_A = -40 to +105 °C, $V_{PDR} \leq EV_{DD0} = EV_{DD1} \leq V_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V

Items	Symbols	Conditions		MIN.	TYP.	MAX.	Unit
RESET and INTMODE	V_{LVI5}	VPOC0,1,2 = 0,1,1 Power down Reset Voltage: 2.7 V		2.70	2.75	2.81	V
	V_{LVI4}	LVIS0,1 = 1,0 (+0.1 V)	Power on Reset Release Voltage	2.86	2.92	2.97	V
			Power down Interrupt Voltage	2.80	2.86	2.91	V
	V_{LVI3}	LVIS0,1 = 0,1 (+0.2 V)	Power on Reset Release Voltage	2.96	3.02	3.08	V
			Power down Interrupt Voltage	2.90	2.96	3.02	V
	V_{LVI0}	LVIS0,1 = 0,0 (+1.2 V)	Power on Reset Release Voltage	3.98	4.06	4.14	V
			Power down Interrupt Voltage	3.90	3.98	4.06	V
Detection Delay time	T_{LD}					200	μs
<R> Minimum pulse width	T_{LW}	Necessary width of V_{DD} drop down below selected V_{LVIx} (x = 0, 3 to 5)		300			μs

Caution LVD reset or external $\overline{\text{RESET}}$ must be used during power supply rising up to 2.7 V.

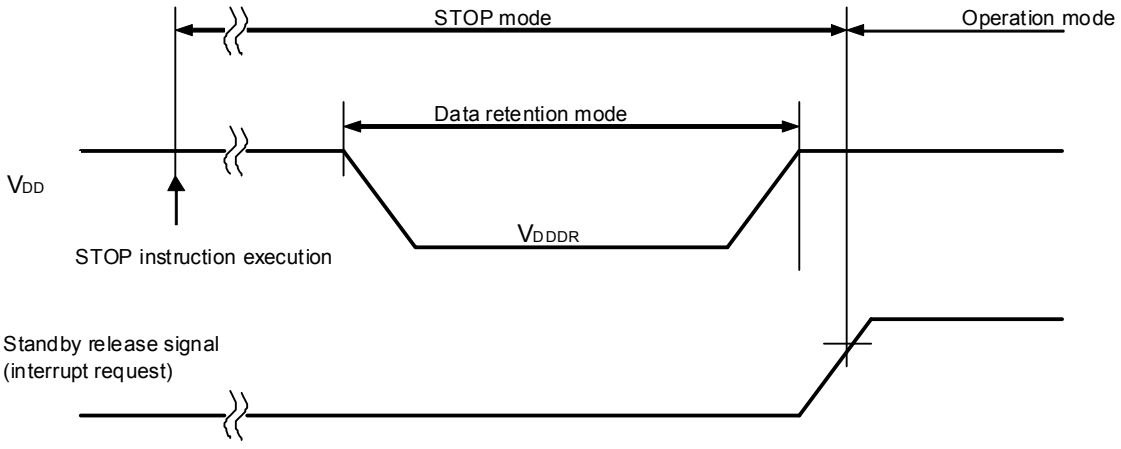
34.8 Data Retention Characteristics

T_A = -40 to +105 °C

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
<R> Data retention supply voltage	V_{DDDR}		1.44 Note1		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained until a POR reset is effected, but data is not retained when a POR reset is effected.

Figure 34-16. STOP Mode Data Retention timming



34.9 Capacitance Connected to REGC

T_A = -40 to +105 °C

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
Capacitance	C_{REG}		0.47		1.0	μF

34.10 Flash programming characteristics

T_A = -40 to +105 °C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V

Items	Symbols	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}	Programming current			12.2	mA
System Clcok frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	2		24	MHz
Number of Code Flash rewrites Notes 1, 2, 3	Cerwr	Retained for 20 years, T_A = +85 °C Note 4	1000			Times
Number of Data Flash rewrites Notes 1, 2, 3	Cerwr	Retained for 20 years, T_A = +85 °C Note 4	10000			Times

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite.
The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

4. The specified data retention time is given under the condition that the average temperature (T_A) is 85°C or below.

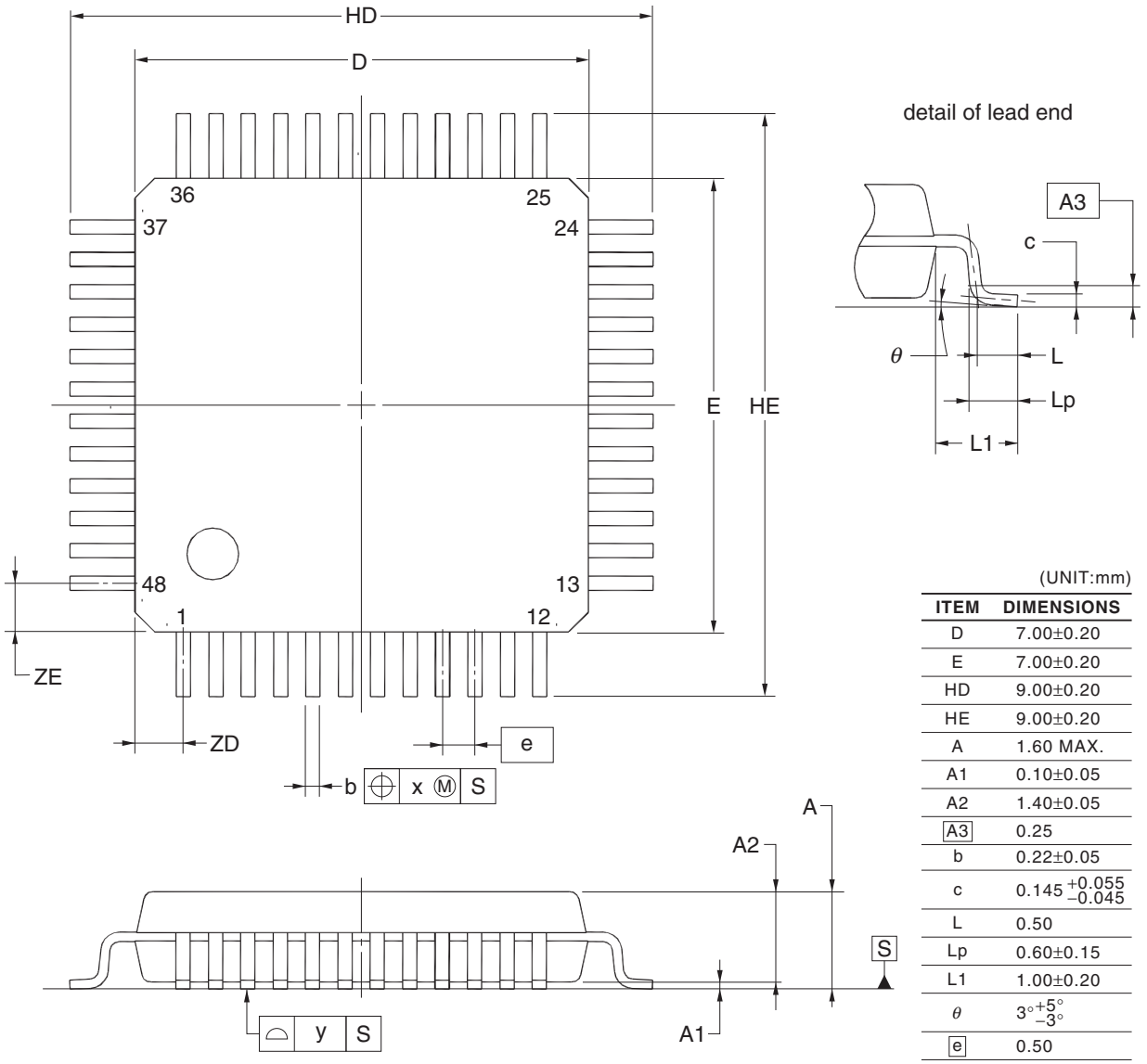
Caution Different voltage between EV_{DD} and V_{DD} is allowed only when LCDM register is initial value.

CHAPTER 35 PACKAGE DRAWINGS

35.1 48-pin products

R5F10CGBxFB, R5F10CGCxFB, R5F10CGDxFB, R5F10DGCxFB, R5F10DGDxFB, R5F10DGExFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP48-7x7-0.50	PLQP0048KF-A	P48GA-50-8EU-1	0.16



(UNIT:mm)

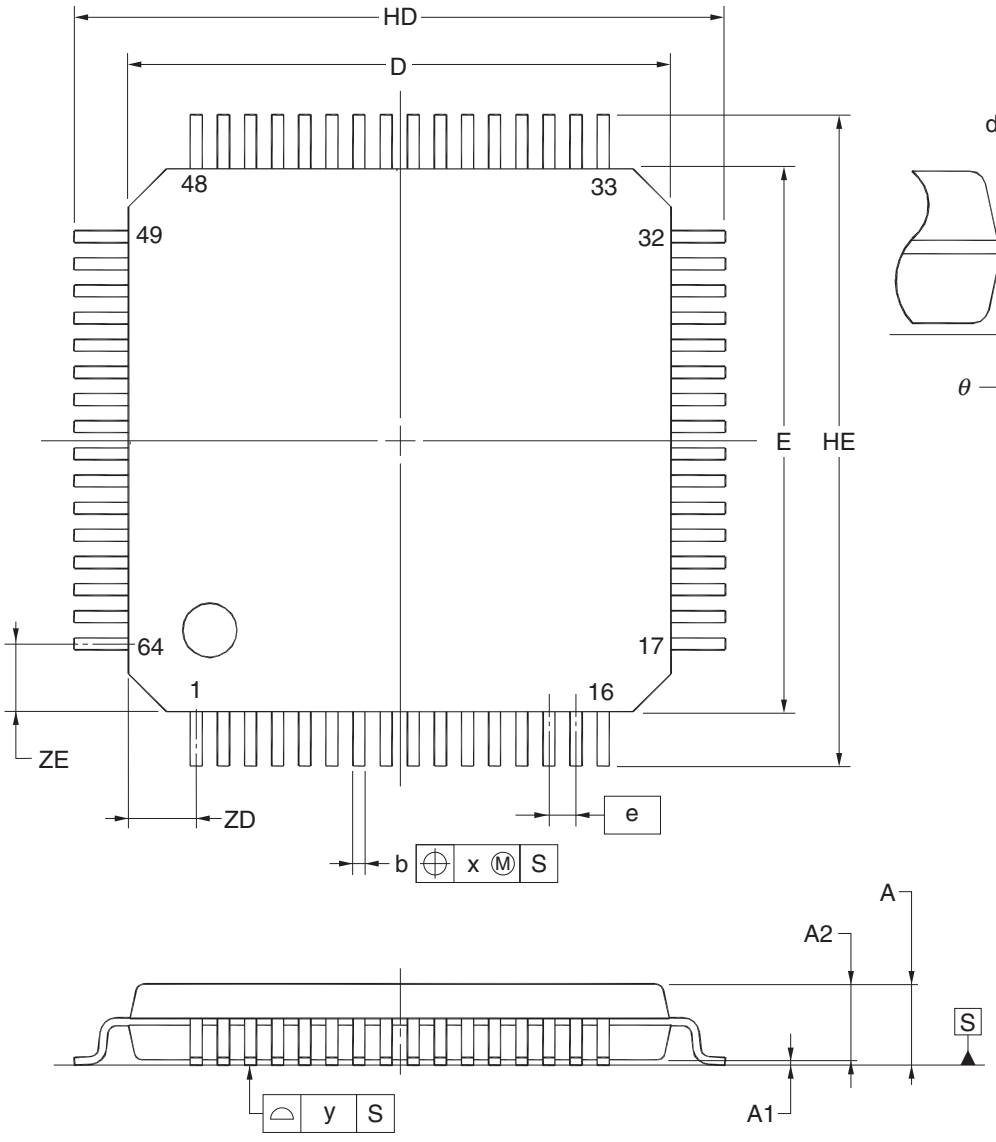
ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

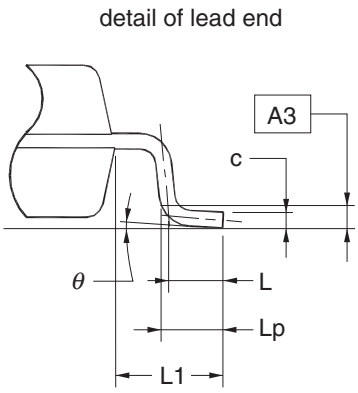
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

35.2 64-pin products

R5F10CLDxFB, R5F10DLDxFB, R5F10DLExFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP64-10x10-0.50	PLQP0064KF-A	P64GB-50-UEU-2	0.35





(UNIT:mm)

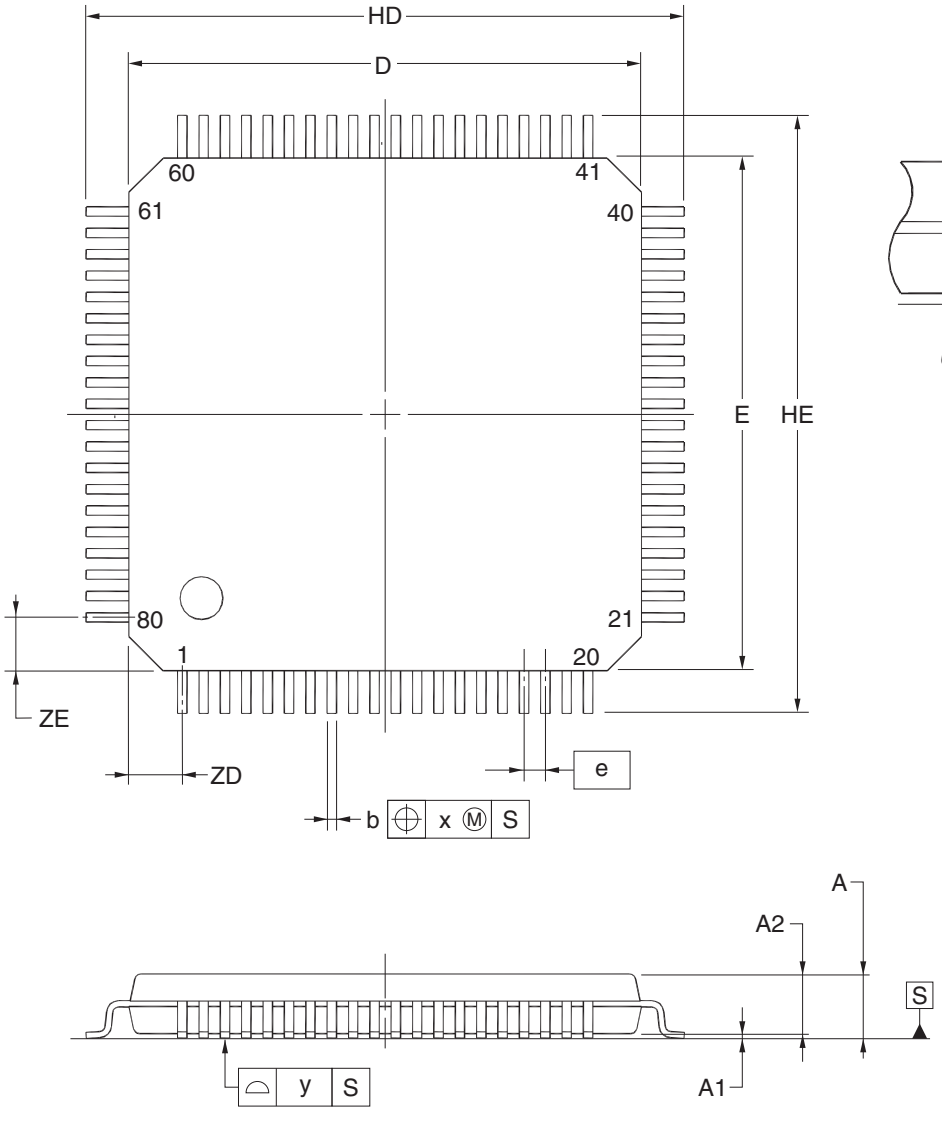
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

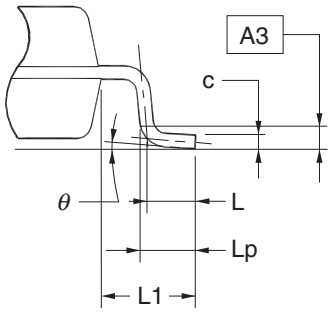
35.3 80-pin products

R5F10CMDxFB, R5F10CMExFB, R5F10DMDxFB, R5F10DMExFB, R5F10DMFxFB, R5F10DMGxFB, R5F10DMJxFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP80-12x12-0.50	PLQP0080KE-A	P80GK-50-8EU-2	0.53



detail of lead end



(UNIT:mm)

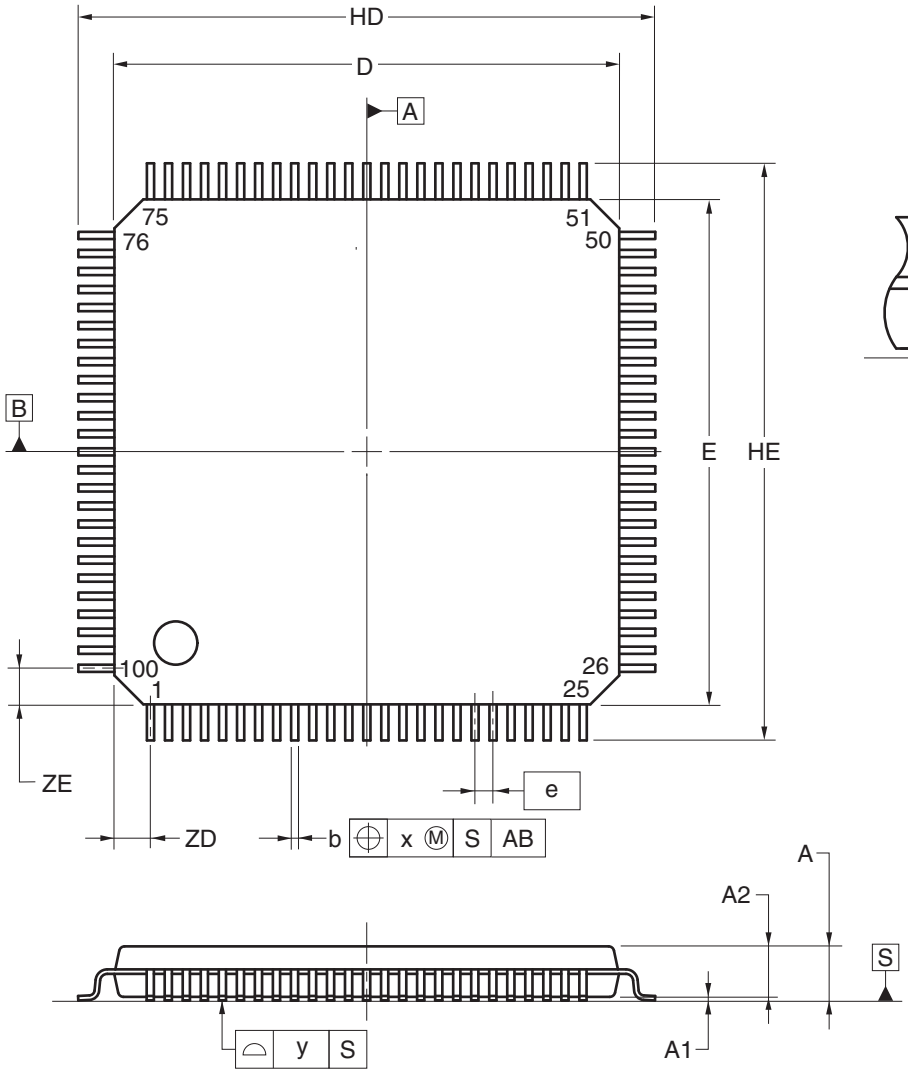
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

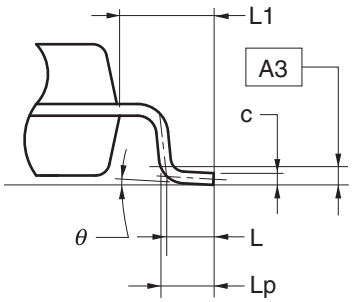
35.4 100-pin products

<R> R5F10DPExFB, R5F10DPFxFB, R5F10DPGxFB, R5F10TPJxFB, R5F10DPJxFB, R5F10DPLxFB, R5F10DPKxFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP100-14x14-0.50	PLQP0100KE-A	P100GC-50-GBR-1	0.69



detail of lead end



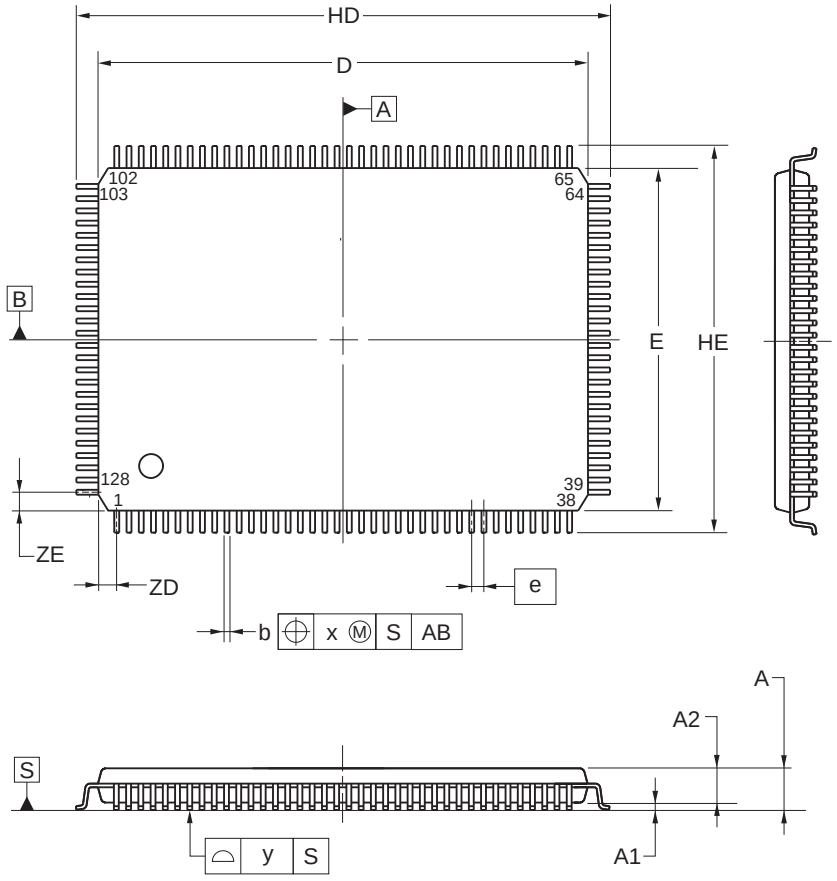
(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

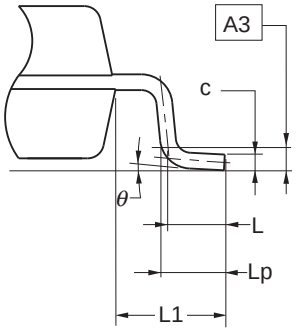
<R> 35.5 128-pin products

R5F10DSJxFB, R5F10DSKxFB, R5F10DSLxFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP128-14x20-0.50	PLQP0128KD-A	P128GF-50-GBP-1	0.92



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	14.00±0.20
HD	22.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

APPENDIX A NUMBER OF WAIT CYCLES TO ACCESS I/O REGISTERS

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0010	ADM2 (A/D converter mode register 2)								E	E	-	E	E	E	R	E	E	R	E
					ADREFP1	ADREFP0	ADREFM		ADRCK	AWC		ADTYP				E	E	E		E	E	-	E
-	-	-	-	F0011	ADUL (Conversion result comparison upper limit setting register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0012	ADLL (Conversion result comparison lower limit setting register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0013	ADTES (AD test register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0030	PU0 (Pull-up resistor option register 0)								E	E	-	E	E	E	E	E	E	E	E
					PU0_7	PU0_6	PU0_5	PU0_4	PU0_3	PU0_2	PU0_1	PU0_0				E	E	E	E	E	E	E	E
-	-	-	-	F0031	PU1 (Pull-up resistor option register 1)								E	E	-	E	E	E	E	E	E	E	E
					PU1_7	PU1_6	PU1_5	PU1_4	PU1_3	PU1_2	PU1_1	PU1_0				E	E	E	E	E	E	E	E
-	-	-	-	F0033	PU3 (Pull-up resistor option register 3)								E	E	-	E	E	E	E	E	E	E	E
					PU3_7	PU3_6	PU3_5	PU3_4	PU3_3	PU3_2	PU3_1	PU3_0				E	E	E	E	E	E	E	E
-	-	-	-	F0034	PU4 (Pull-up resistor option register 4)								E	E	-	R	R	R	R	R	R	R	E
												PU4_0				-	-	-	-	-	-	-	E
-	-	-	-	F0035	PU5 (Pull-up resistor option register 5)								E	E	-	E	E	E	E	E	E	E	E
					PU5_7	PU5_6	PU5_5	PU5_4	PU5_3	PU5_2	PU5_1	PU5_0				E	E	E	E	E	E	E	E
-	-	-	-	F0036	PU6 (Pull-up resistor option register 6)								E	E	-	R	E	E	E	E	E	E	E
						PU6_6	PU6_5	PU6_4	PU6_3	PU6_2	PU6_1	PU6_0				-	E	E	E	E	E	E	E
-	-	-	-	F0037	PU7 (Pull-up resistor option register 7)								E	E	-	R	R	E	E	E	E	E	E
							PU7_5	PU7_4	PU7_3	PU7_2	PU7_1	PU7_0				-	-	E	E	E	E	E	E
-	-	-	-	F0038	PU8 (Pull-up resistor option register 8)								E	E	-	E	E	E	E	E	E	E	E
					PU8_7	PU8_6	PU8_5	PU8_4	PU8_3	PU8_2	PU8_1	PU8_0				E	E	E	E	E	E	E	E
-	-	-	-	F0039	PU9 (Pull-up resistor option register 9)								E	E	-	E	E	E	E	E	E	E	E
					PU9_7	PU9_6	PU9_5	PU9_4	PU9_3	PU9_2	PU9_1	PU9_0				E	E	E	E	E	E	E	E
-	-	-	-	F003D	PU13 (Pull-up resistor option register 13)								E	E	-	R	E	E	E	E	E	E	R
						PU13_6	PU13_5	PU13_4	PU13_3	PU13_2	PU13_1					-	E	E	E	E	E	E	-
-	-	-	-	F003E	PU14 (Pull-up resistor option register 14)								E	E	-	R	R	R	R	R	R	R	E
												PU14_0				-	-	-	-	-	-	-	E
-	-	-	-	F0040	PIM0 (Port intput mode register 0)								E	E	-	R	R	R	R	R	R	E	R
											PIM0_1					-	-	-	-	-	-	E	-
-	-	-	-	F0041	PIM1 (Port intput mode register 1)								E	E	-	E	R	R	R	R	R	E	E
					PIM1_7						PIM1_1	PIM1_0				E	-	-	-	-	-	E	

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0043	PIM3 (Port intput mode register 3)								E	E	-	R	R	R	R	R	R	E	R
											PIM3_1					-	-	-	-	-	-	E	-
-	-	-	-	F0045	PIM5 (Port intput mode register 5)								E	E	-	E	E	E	R	R	E	E	E
					PIM5_7	PIM5_6	PIM5_5			PIM5_2	PIM5_1	PIM5_0				E	E	E	-	-	E	E	E
-	-	-	-	F0046	PIM6 (Port intput mode register 6)								E	E	-	R	R	R	R	E	R	E	R
									PIM6_3		PIM6_1					-	-	-	-	E	-	E	-
-	-	-	-	F0047	PIM7 (Port intput mode register 7)								E	E	-	R	R	R	R	R	R	R	E
												PIM7_0				-	-	-	-	-	-	-	E
-	-	-	-	F004D	PIM13 (Port intput mode register 13)								E	E	-	R	R	E	R	R	R	R	R
							PIM13_5									-	-	E	-	-	-	-	-
-	-	-	-	F0050	LCDPF0 (LCD port function register 0)								E	E	-	E	E	E	E	E	E	E	E
					LCDPF0_7	LCDPF0_6	LCDPF0_5	LCDPF0_4	LCDPF0_3	LCDPF0_2	LCDPF0_1	LCDPF0_0				E	E	E	E	E	E	E	E
-	-	-	-	F0051	LCDPF1 (LCD port function register 1)								E	E	-	E	E	E	E	E	E	E	E
					LCDPF1_7	LCDPF1_6	LCDPF1_5	LCDPF1_4	LCDPF1_3	LCDPF1_2	LCDPF1_1	LCDPF1_0				E	E	E	E	E	E	E	E
-	-	-	-	F0053	LCDPF3 (LCD port function register 3)								E	E	-	E	E	E	E	E	E	E	E
					LCDPF3_7	LCDPF3_6	LCDPF3_5	LCDPF3_4	LCDPF3_3	LCDPF3_2	LCDPF3_1	LCDPF3_0				E	E	E	E	E	E	E	E
-	-	-	-	F0055	LCDPF5 (LCD port function register 5)								E	E	-	E	E	E	E	E	E	E	E
					LCDPF5_7	LCDPF5_6	LCDPF5_5	LCDPF5_4	LCDPF5_3	LCDPF5_2	LCDPF5_1	LCDPF5_0				E	E	E	E	E	E	E	E
-	-	-	-	F0057	LCDPF7 (LCD port function register 7)								E	E	-	R	R	E	E	E	E	R	R
							LCDPF7_5	LCDPF7_4	LCDPF7_3	LCDPF7_2						-	-	E	E	E	E	-	-
-	-	-	-	F0058	LCDPF8 (LCD port function register 8)								E	E	-	E	E	E	E	E	E	E	E
					LCDPF8_7	LCDPF8_6	LCDPF8_5	LCDPF8_4	LCDPF8_3	LCDPF8_2	LCDPF8_1	LCDPF8_0				E	E	E	E	E	E	E	E
-	-	-	-	F0059	LCDPF9 (LCD port function register 9)								E	E	-	E	E	E	E	E	E	E	E
					LCDPF9_7	LCDPF9_6	LCDPF9_5	LCDPF9_4	LCDPF9_3	LCDPF9_2	LCDPF9_1	LCDPF9_0				E	E	E	E	E	E	E	E
-	-	-	-	F005D	LCDPF13 (LCD port function register 13)								E	E	-	R	E	R	R	R	R	R	R
						LCDPF13_6										-	E	-	-	-	-	-	-
-	-	-	-	F0060	TNFEN0 (Noise filter enable register for each channel of TAU unit0)								E	E	-	E	E	E	E	E	E	E	E
					TNFEN0_7	TNFEN0_6	TNFEN0_5	TNFEN0_4	TNFEN0_3	TNFEN0_2	TNFEN0_1	TNFEN0_0				E	E	E	E	E	E	E	E
-	-	-	-	F0061	TNFSMP0 (Sampling clock select of noise filter for unit0 (2set))								E	E	-	E	E	E	E	E	E	E	E
					TNFSMP013	TNFSMP012	TNFSMP011	TNFSMP010	TNFSMP003	TNFSMP002	TNFSMP001	TNFSMP000				E	E	E	E	E	E	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0062	TNFCS0 (Noise filter clock select register for each channel of TAU unit0)								E	E	-	E	E	E	E	E	E	E	E
					TNFCS0_7	TNFCS0_6	TNFCS0_5	TNFCS0_4	TNFCS0_3	TNFCS0_2	TNFCS0_1	TNFCS0_0				E	E	E	E	E	E	E	E
-	-	-	-	F0064	TNFEN1 (Noise filter enable register for each channel of TAU unit1)								E	E	-	E	E	E	E	E	E	E	E
					TNFEN1_7	TNFEN1_6	TNFEN1_5	TNFEN1_4	TNFEN1_3	TNFEN1_2	TNFEN1_1	TNFEN1_0				E	E	E	E	E	E	E	E
-	-	-	-	F0065	TNFSMP1 (Sampling clock select of noise filter for unit1 (2set))								E	E	-	E	E	E	E	E	E	E	E
					TNFSMP113	TNFSMP112	TNFSMP111	TNFSMP110	TNFSMP103	TNFSMP102	TNFSMP101	TNFSMP100				E	E	E	E	E	E	E	E
-	-	-	-	F0066	TNFCS1 (Noise filter clock select register for each channel of TAU unit1)								E	E	-	E	E	E	E	E	E	E	E
					TNFCS1_7	TNFCS1_6	TNFCS1_5	TNFCS1_4	TNFCS1_3	TNFCS1_2	TNFCS1_1	TNFCS1_0				E	E	E	E	E	E	E	E
-	-	-	-	F0068	TNFEN2 (Noise filter enable register for each channel of TAU unit2)								E	E	-	E	E	E	E	E	E	E	E
					TNFEN2_7	TNFEN2_6	TNFEN2_5	TNFEN2_4	TNFEN2_3	TNFEN2_2	TNFEN2_1	TNFEN2_0				E	E	E	E	E	E	E	E
-	-	-	-	F0069	TNFSMP2 (Sampling clock select of noise filter for unit2(2set))								E	E	-	E	E	E	E	E	E	E	E
					TNFSMP213	TNFSMP212	TNFSMP211	TNFSMP210	TNFSMP203	TNFSMP202	TNFSMP201	TNFSMP200				E	E	E	E	E	E	E	E
-	-	-	-	F006A	TNFCS2 (Noise filter clock select register for each channel of TAU unit2)								E	E	-	E	E	E	E	E	E	E	E
					TNFCS2_7	TNFCS2_6	TNFCS2_5	TNFCS2_4	TNFCS2_3	TNFCS2_2	TNFCS2_1	TNFCS2_0				E	E	E	E	E	E	E	E
-	-	-	-	F006E	ADPC (A/D port configuration register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F006F	POM (Port output mode register)								E	E	-	R	R	E	E	E	E	E	E
							POM_5	POM_4	POM_3	POM_2	POM_1	POM_0				-	-	E	E	E	E	E	E
-	-	-	-	F0070	TIS00 (Timer input select register 00)								E	E	-	E	E	R	E	E	E	R	E
					TIS031	TIS030		TIS020	TIS011	TIS010		TIS000				E	E	-	E	E	E	-	E
-	-	-	-	F0071	TIS01 (Timer input select register 01)								E	E	-	E	E	R	E	E	E	R	E
					TIS071	TIS070		TIS060	TIS051	TIS050		TIS040				E	E	-	E	E	E	-	E
-	-	-	-	F0072	TIS10 (Timer input select register 10)								E	E	-	E	E	R	E	E	E	R	R
					TIS131	TIS130		TIS120	TIS111	TIS110						E	E	-	E	E	E	-	-
-	-	-	-	F0073	TIS11 (Timer input select register 11)								E	E	-	E	E	E	E	E	E	E	E
					TIS171	TIS170	TIS161	TIS160	TIS151	TIS150	TIS141	TIS140				E	E	E	E	E	E	E	E
-	-	-	-	F0074	TIS20 (Timer input select register 20)								E	E	-	E	E	E	E	E	E	E	E
					TIS231	TIS230	TIS221	TIS220	TIS211	TIS210	TIS201	TIS200				E	E	E	E	E	E	E	E
-	-	-	-	F0075	TIS21 (Timer input select register 21)								E	E	-	E	E	E	E	E	E	E	E
					TIS271	TIS270	TIS261	TIS260	TIS251	TIS250	TIS241	TIS240				E	E	E	E	E	E	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0076	TOS00 (Timer output select register 00)								E	E	-	E	E	R	E	E	E	R	E
					TOS031	TOS030		TOS020	TOS011	TOS010		TOS000				E	E	-	E	E	E	-	E
-	-	-	-	F0077	TOS01 (Timer output select register 01)								E	E	-	E	E	R	E	E	E	R	E
					TOS071	TOS070		TOS060	TOS051	TOS050		TOS040				E	E	-	E	E	E	-	E
-	-	-	-	F0078	IAWCTL (Illegal-memory access detection control register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0079	TOS10 (Timer output select register 10)								E	E	-	E	E	R	E	E	E	R	R
					TOS131	TOS130		TOS120	TOS111	TOS110						E	E	-	E	E	E	-	-
-	-	-	-	F007A	TOS11 (Timer output select register 11)								E	E	-	E	E	E	E	E	E	E	E
					TOS171	TOS170	TOS161	TOS160	TOS151	TOS150	TOS141	TOS140				E	E	E	E	E	E	E	E
-	-	-	-	F007B	TOS20 (Timer output select register 20)								E	E	-	E	E	E	E	E	E	E	E
					TOS231	TOS230	TOS221	TOS220	TOS211	TOS210	TOS201	TOS200				E	E	E	E	E	E	E	E
-	-	-	-	F007C	TOS21 (Timer output select register 21)								E	E	-	E	E	E	E	E	E	E	E
					TOS271	TOS270	TOS261	TOS260	TOS251	TOS250	TOS241	TOS240				E	E	E	E	E	E	E	E
-	-	-	-	F0090	DFLCTL (Data flash control register)								E	E	-	R	R	R	R	R	R	R	E
												DFLEN				-	-	-	-	-	-	-	E
-	-	-	-	F00A0	HIOTRM (high-speed on-chip oscillator trimming register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F00E0	MDCL (Multiplication/division data registers C(L))								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F00E2	MDCH (Multiplication/division data registers C(H))								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F00E8	MDUC (Multiplication/division control register)								E	E	-	E	E	R	R	E	R	R	E
					DIVMODE	MACMODE			MDSM	MACOF	MACSF	DIVST				E	E	-	-	E	R	R	E
-	-	-	-	F00F0	PER0 (Peripheral enable register 0)								E	E	-	E	E	E	E	E	E	E	E
					RTCEN	LIN1EN	LIN0EN	SAU1EN	SAU0EN	TAU2EN	TAU1EN	TAU0EN				E	E	E	E	E	E	E	E
-	-	-	-	F00F1	PER1 (Peripheral enable register 1)								E	E	-	E	R	E	E	R	R	R	R
					ADCEN		MTRCEN	SGEN								E	-	E	E	-	-	-	-
-	-	-	-	F00F2	PCKSEL (Peripheral clock select register)								E	E	-	R	E	E	E	E	R	R	E
						CANMCKE1	CANMCK1	CANMCKE0	CANMCK0			SGCLKSEL				-	E	E	E	E	-	-	E
-	-	-	-	F00F3	OSMC (Operation speed mode control register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F00F5	RPECTL (RAM parity error control register)								E	E	-	E	R	R	R	R	R	R	E
					RPERDIS							RPEF				E	-	-	-	-	-	-	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F00F8	MDIV (FMP clock division selction register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F00F9	RTCCL (RTC clock selection register)								E	E	-	E	E	R	R	R	R	E	E
					RTCCL_7	RTCCL_6					RTCCKS1	RTCCKS0				E	E	-	-	-	-	E	E
-	-	-	-	F00FA	RESFCLM (CLM reset control flag register)								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F00FB	POCRES (POC reset confirm register)								E	E	-	R	R	R	R	R	R	R	E
												POCRES_0				-	-	-	-	-	-	-	E
-	-	-	-	F00FC	GUARD (Specific register manipulation protection register)								E	E	-	R	R	R	R	R	R	E	E
											GDRTC	GDPLL				-	-	-	-	-	-	E	E
-	-	-	-	F00FE	BCDADJ (BCD correction result register)								-	R	-	-	-	-	-	-	-	-	-
1	1	1	1	F0100	SSR00 (Serial status register 00)								-	-	R	-	-	-	-	-	-	-	-
					SSR00L								-	R	-	-	-	-	-	-	-	-	-
1	1	1	1	F0102	SSR01 (Serial status register 01)								-	-	R	-	-	-	-	-	-	-	-
					SSR01L								-	R	-	-	-	-	-	-	-	-	-
1	1	1	1	F0104	SIR00 (Serial flag clear trigger register 00)								-	-	E	-	-	-	-	-	-	-	-
					SIR00L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0106	SIR01 (Serial flag clear trigger register 01)								-	-	E	-	-	-	-	-	-	-	-
					SIR01L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0108	SMR00 (Serial mode register 00)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F010A	SMR01 (Serial mode register 01)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F010C	SCR00 (Serial communication operation setting register 00)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F010E	SCR01 (Serial communication operation setting register 01)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F0110	SE0 (Serial channel enable status register 0)								-	-	R	-	-	-	-	-	-	-	-
					SE0L								R	R	-	R	R	R	R	R	R	R	R
											SE0_1	SE0_0				-	-	-	-	-	-	R	R
1	1	1	1	F0112	SS0 (Serial channel start trigger register 0)								-	-	E	-	-	-	-	-	-	-	-
					SS0L								E	E	-	R	R	R	R	R	R	E	E
											SS0_1	SS0_0				-	-	-	-	-	-	E	E
1	1	1	1	F0114	ST0 (Serial channel stop trigger register 0)								-	-	E	-	-	-	-	-	-	-	-
					ST0L								E	E	-	R	R	R	R	R	R	E	E
											ST0_1	ST0_0				-	-	-	-	-	-	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
1	1	1	1	F0116	SPS0 (Serial clock select register 0)								-	-	E	-	-	-	-	-	-	-	-
					SPS0L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0118	SO0 (Serial output register 0)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F011A	SOE0 (Serial output enabel register 0)								-	-	E	-	-	-	-	-	-	-	-
					SOE0L								E	E	-	R	R	R	R	R	R	E	E
											SOE0_1	SOE0_0				-	-	-	-	-	-	E	E
1	1	1	1	F0120	SOL0 (Serial output level register 0)								-	-	E	-	-	-	-	-	-	-	-
					SOL0L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0128	PLLSTS (PLL status register)								R	R	-	R	R	R	R	R	R	R	R
					LOCK				SELPLLS							R	-	-	-	R	-	-	-
1	1	1	1	F0129	PLLCTL (PLL control register)								E	E	-	E	E	R	E	R	E	R	E
					LCKSEL1	LCKSEL0		PLLDIV0		SELPLL		PLLON				E	E	-	E	-	E	-	E
1	1	1	1	F0130	SSR10 (Serial status register 10)								-	-	R	-	-	-	-	-	-	-	-
					SSR10L								-	R	-	-	-	-	-	-	-	-	-
1	1	1	1	F0132	SSR11 (Serial status register 11)								-	-	R	-	-	-	-	-	-	-	-
					SSR11L								-	R	-	-	-	-	-	-	-	-	-
1	1	1	1	F0134	SIR10 (Serial flag clear trigger register 10)								-	-	E	-	-	-	-	-	-	-	-
					SIR10L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0136	SIR11 (Serial flag clear trigger register 11)								-	-	E	-	-	-	-	-	-	-	-
					SIR11L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0138	SMR10 (Serial mode register 10)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F013A	SMR11 (Serial mode register 11)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F013C	SCR10 (Serial communication operation setting register 10)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F013E	SCR11 (Serial communication operation setting register 11)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F0140	SE1 (Serial channel enable status register 1)								-	-	R	-	-	-	-	-	-	-	-
					SE1L								R	R	-	R	R	R	R	R	R	R	R
											SE1_1	SE1_0				-	-	-	-	-	-	R	R

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
1	1	1	1	F0142	SS1 (Serial channel start trigger register 1)								-	-	E	-	-	-	-	-	-	-	-
					SS1L								E	E	-	R	R	R	R	R	R	E	E
											SS1_1	SS1_0				-	-	-	-	-	-	E	E
1	1	1	1	F0144	ST1 (Serial channel stop trigger register 1)								-	-	E	-	-	-	-	-	-	-	-
					ST1L								E	E	-	R	R	R	R	R	R	E	E
											ST1_1	ST1_0				-	-	-	-	-	-	E	E
1	1	1	1	F0146	SPS1 (Serial clock select register 1)								-	-	E	-	-	-	-	-	-	-	-
					SPS1L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0148	SO1 (Serial output register 1)								-	-	E	-	-	-	-	-	-	-	-
					SO1L								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F014A	SOE1 (Serial output enabel register 1)								-	-	E	-	-	-	-	-	-	-	-
					SOE1L								E	E	-	R	R	R	R	R	R	E	E
											SOE1_1	SOE1_0				-	-	-	-	-	-	E	E
1	1	1	1	F0150	SOL1 (Serial output level register 1)								-	-	E	-	-	-	-	-	-	-	-
					SOL1L								-	E	-	-	-	-	-	-	-	-	-
1	1	5	7	F0158	MCMPC4 (Compare control register 4)								E	E	-	E	E	R	E	E	E	E	E
					AOUT4	TWIN4	ZPD4	TEN4	ADB41	ADB40	DIR41	DIR40				E	E	R	E	E	E	E	E
1	1	1	1	F015C	ZPDS0 (ZPD detection voltage setting register0/ZPD analog input control)								E	E	-	E	E	E	E	E	E	E	E
					ZPD2PC	ZPD2S2	ZPD2S1	ZPD2S0	ZPD1PC	ZPD1S2	ZPD1S1	ZPD1S0				E	E	E	E	E	E	E	E
1	1	1	1	F015D	ZPDS1 (ZPD detection voltage setting register1/ZPD analog input control)								E	E	-	E	E	E	E	E	E	E	E
					ZPD4PC	ZPD4S2	ZPD4S1	ZPD4S0	ZPD3PC	ZPD3S2	ZPD3S1	ZPD3S0				E	E	E	E	E	E	E	E
1	1	1	1	F015E	CMPCTL (ZPD flag detection clock setting register)								E	E	-	R	R	R	R	E	E	E	E
									DBCL3	DBCL2	DBCL1	DBCL0				-	-	-	-	E	E	E	E
1	1	1	1	F015F	ZPDEN (ZPD operation control register)								E	E	-	R	R	R	R	E	E	E	E
									ZPD4EN	ZPD3EN	ZPD2EN	ZPD1EN				-	-	-	-	E	E	E	E
1	1	5	7	F0160	MCNTC0 (Compare control register 0)								E	E	-	E	R	E	E	E	E	E	E
					CAE		FULL	PCE	PCS	SMCL2	SMCL1	SMCL0				E	-	E	E	E	E	E	E
1	1	1	1	F0162	MCMP1HW (Compare register 1HW)								-	-	E	-	-	-	-	-	-	-	-
					MCMP10 (Compare register 10)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0163	MCMP11 (Compare register 11)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
1	1	1	1	F0164	MCMP2HW (Compare register 2HW)								-	-	E	-	-	-	-	-	-	-	-
					MCMP20 (Compare register 20)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0165	MCMP21 (Compare register 21)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0166	MCMP3HW (Compare register 3HW)								-	-	E	-	-	-	-	-	-	-	-
					MCMP30 (Compare register 30)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0167	MCMP31 (Compare register 31)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0168	MCMP4HW (Compare register 4HW)								-	-	E	-	-	-	-	-	-	-	-
					MCMP40 (Compare register 40)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0169	MCMP41 (Compare register 41)								-	E	-	-	-	-	-	-	-	-	-
1	1	5	7	F016A	MCMPC1 (Compare control register 1)								E	E	-	E	E	R	E	E	E	E	E
					AOUT1	TWIN1	ZPD1	TEN1	ADB11	ADB10	DIR11	DIR10				E	E	R	E	E	E	E	E
1	1	5	7	F016C	MCMPC2 (Compare control register 2)								E	E	-	E	E	R	E	E	E	E	E
					AOUT2	TWIN2	ZPD2	TEN2	ADB21	ADB20	DIR21	DIR20				E	E	R	E	E	E	E	E
1	1	5	7	F016E	MCMPC3 (Compare control register 3)								E	E	-	E	E	R	E	E	E	E	E
					AOUT3	TWIN3	ZPD3	TEN3	ADB31	ADB30	DIR31	DIR30				E	E	R	E	E	E	E	E
1	1	1	1	F0170	DSA2 (DMA SFR address register 2)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0171	DSA3 (DMA SFR address register 3)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0172	DRA2 (DMA RAM address register 2)								-	-	E	-	-	-	-	-	-	-	-
					DRA2L (DMA RAM address register 2L)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0173	DRA2H (DMA RAM address register 2H)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0174	DRA3 (DMA RAM address register 3)								-	-	E	-	-	-	-	-	-	-	-
					DRA3L (DMA RAM address register 3L)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0175	DRA3H (DMA RAM address register 3H)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0176	DBC2 (DMA byte count register 2)								-	-	E	-	-	-	-	-	-	-	-
					DBC2L (DMA byte count register 2L)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0177	DBC2H (DMA byte count register 2H)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0178	DBC3 (DMA byte count register 3)								-	-	E	-	-	-	-	-	-	-	-
					DBC3L (DMA byte count register 3L)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0179	DBC3H (DMA byte count register 3H)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F017A	DMC2 (DMA mode control register 2)								E	E	-	E	E	E	E	E	E	E	E
					STG2	DRS2	DS2	DWAIT2	IFC23	IFC22	IFC21	IFC20				E	E	E	E	E	E	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
1	1	1	1	F017B	DMC3 (DMA mode control register 3)								E	E	-	E	E	E	E	E	E	E	E
					STG3	DRS3	DS3	DWAIT3	IFC33	IFC32	IFC31	IFC30				E	E	E	E	E	E	E	E
1	1	1	1	F017C	DRC2 (DMA operation control register 2)								E	E	-	E	R	R	R	R	R	R	E
					DEN2							DST2				E	-	-	-	-	-	-	E
1	1	1	1	F017D	DRC3 (DMA operation control register 3)								E	E	-	E	R	R	R	R	R	R	E
					DEN3							DST3				E	-	-	-	-	-	-	E
1	1	1	1	F017F	DWAITALL (DMA all-channel forced wait register)								E	E	-	E	R	R	R	R	R	R	E
					PRVARI							DWAITALL0				E	-	-	-	-	-	-	E
-	-	-	-	F0180	TCR00 (Timer counter register 00)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0182	TCR01 (Timer counter register 01)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0184	TCR02 (Timer counter register 02)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0186	TCR03 (Timer counter register 03)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0188	TCR04 (Timer counter register 04)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F018A	TCR05 (Timer counter register 05)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F018C	TCR06 (Timer counter register 06)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F018E	TCR07 (Timer counter register 07)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0190	TMR00 (Timer mode register 00)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0192	TMR01 (Timer mode register 01)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0194	TMR02 (Timer mode register 02)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0196	TMR03 (Timer mode register 03)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0198	TMR04 (Timer mode register 04)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F019A	TMR05 (Timer mode register 05)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F019C	TMR06 (Timer mode register 06)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F019E	TMR07 (Timer mode register 07)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01A0	TSR00 (Timer status register 00)								-	-	R	-	-	-	-	-	-	-	-
					TSR00L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01A2	TSR01 (Timer status register 01)								-	-	R	-	-	-	-	-	-	-	-
					TSR01L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01A4	TSR02 (Timer status register 02)								-	-	R	-	-	-	-	-	-	-	-
					TSR02L								-	R	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F01A6	TSR03 (Timer status register 03)								-	-	R	-	-	-	-	-	-	-	-
					TSR03L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01A8	TSR04 (Timer status register 04)								-	-	R	-	-	-	-	-	-	-	-
					TSR04L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01AA	TSR05 (Timer status register 05)								-	-	R	-	-	-	-	-	-	-	-
					TSR05L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01AC	TSR06 (Timer status register 06)								-	-	R	-	-	-	-	-	-	-	-
					TSR06L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01AE	TSR07 (Timer status register 07)								-	-	R	-	-	-	-	-	-	-	-
					TSR07L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01B0	TE0 (Timer channel enable status register0)								-	-	R	-	-	-	-	-	-	-	-
					TE0L								R	R	-	R	R	R	R	R	R	R	R
					TE0_7	TE0_6	TE0_5	TE0_4	TE0_3	TE0_2	TE0_1	TE0_0				R	R	R	R	R	R	R	R
-	-	-	-	F01B2	TS0 (Timer channel start trigger register 0)								-	-	E	-	-	-	-	-	-	-	-
					TS0L								E	E	-	E	E	E	E	E	E	E	E
					TS0_7	TS0_6	TS0_5	TS0_4	TS0_3	TS0_2	TS0_1	TS0_0				E	E	E	E	E	E	E	E
-	-	-	-	F01B4	TT0 (Timer channel stop trigger register 0)								-	-	E	-	-	-	-	-	-	-	-
					TT0L								E	E	-	E	E	E	E	E	E	E	E
					TT0_7	TT0_6	TT0_5	TT0_4	TT0_3	TT0_2	TT0_1	TT0_0				E	E	E	E	E	E	E	E
-	-	-	-	F01B6	TPS0 (Timer clock select register 0)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01B8	TO0 (Timer output register 0)								-	-	E	-	-	-	-	-	-	-	-
					TO0L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F01BA	TOE0 (Timer output enable register 0)								-	-	E	-	-	-	-	-	-	-	-
					TOE0L								E	E	-	E	E	E	E	E	E	E	E
					TOE0_7	TOE0_6	TOE0_5	TOE0_4	TOE0_3	TOE0_2	TOE0_1	TOE0_0				E	E	E	E	E	E	E	E
-	-	-	-	F01BC	TOL0 (Timer output level register 0)								-	-	E	-	-	-	-	-	-	-	-
					TOL0L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F01BE	TOM0 (Timer output mode register 0)								-	-	E	-	-	-	-	-	-	-	-
					TOM0L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F01C0	TCR10 (Timer counter register 10)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01C2	TCR11 (Timer counter register 11)								-	-	R	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F01C4	TCR12 (Timer counter register 12)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01C6	TCR13 (Timer counter register 13)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01C8	TCR14 (Timer counter register 14)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01CA	TCR15 (Timer counter register 15)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01CC	TCR16 (Timer counter register 16)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01CE	TCR17 (Timer counter register 17)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F01D0	TMR10 (Timer mode register 10)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01D2	TMR11 (Timer mode register 11)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01D4	TMR12 (Timer mode register 12)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01D6	TMR13 (Timer mode register 13)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01D8	TMR14 (Timer mode register 14)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01DA	TMR15 (Timer mode register 15)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01DC	TMR16 (Timer mode register 16)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01DE	TMR17 (Timer mode register 17)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01E0	TSR10 (Timer status register 10)								-	-	R	-	-	-	-	-	-	-	-
					TSR10L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01E2	TSR11 (Timer status register 11)								-	-	R	-	-	-	-	-	-	-	-
					TSR11L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01E4	TSR12 (Timer status register 12)								-	-	R	-	-	-	-	-	-	-	-
					TSR12L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01E6	TSR13 (Timer status register 13)								-	-	R	-	-	-	-	-	-	-	-
					TSR13L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01E8	TSR14 (Timer status register 14)								-	-	R	-	-	-	-	-	-	-	-
					TSR14L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01EA	TSR15 (Timer status register 15)								-	-	R	-	-	-	-	-	-	-	-
					TSR15L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01EC	TSR16 (Timer status register 16)								-	-	R	-	-	-	-	-	-	-	-
					TSR16L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F01EE	TSR17 (Timer status register 17)								-	-	R	-	-	-	-	-	-	-	-
					TSR17L								-	R	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F01F0	TE1 (Timer channel enable status register 1)								-	-	R	-	-	-	-	-	-	-	-
					TE1L								R	R	-	R	R	R	R	R	R	R	R
					TE1_7	TE1_6	TE1_5	TE1_4	TE1_3	TE1_2	TE1_1	TE1_0				R	R	R	R	R	R	R	R
-	-	-	-	F01F2	TS1 (Timer channel start trigger register 1)								-	-	E	-	-	-	-	-	-	-	-
					TS1L								E	E	-	E	E	E	E	E	E	E	E
					TS1_7	TS1_6	TS1_5	TS1_4	TS1_3	TS1_2	TS1_1	TS1_0				E	E	E	E	E	E	E	E
-	-	-	-	F01F4	TT1 (Timer channel stop trigger register 1)								-	-	E	-	-	-	-	-	-	-	-
					TT1L								E	E	-	E	E	E	E	E	E	E	E
					TT1_7	TT1_6	TT1_5	TT1_4	TT1_3	TT1_2	TT1_1	TT1_0				E	E	E	E	E	E	E	E
-	-	-	-	F01F6	TPS1 (Timer clock select register 1)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F01F8	TO1 (Timer output register 1)								-	-	E	-	-	-	-	-	-	-	-
					TO1L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F01FA	TOE1 (Timer output enable register 1)								-	-	E	-	-	-	-	-	-	-	-
					TOE1L								E	E	-	E	E	E	E	E	E	E	E
					TOE1_7	TOE1_6	TOE1_5	TOE1_4	TOE1_3	TOE1_2	TOE1_1	TOE1_0				E	E	E	E	E	E	E	E
-	-	-	-	F01FC	TOL1 (Timer output level register 1)								-	-	E	-	-	-	-	-	-	-	-
					TOL1L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F01FE	TOM1 (Timer output mode register 1)								-	-	E	-	-	-	-	-	-	-	-
					TOM1L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0200	TCR20 (Timer counter register 20)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0202	TCR21 (Timer counter register 21)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0204	TCR22 (Timer counter register 22)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0206	TCR23 (Timer counter register 23)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0208	TCR24 (Timer counter register 24)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F020A	TCR25 (Timer counter register 25)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F020C	TCR26 (Timer counter register 26)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F020E	TCR27 (Timer counter register 27)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0210	TMR20 (Timer mode register 20)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0212	TMR21 (Timer mode register 21)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0214	TMR22 (Timer mode register 22)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0216	TMR23 (Timer mode register 23)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0218	TMR24 (Timer mode register 24)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F021A	TMR25 (Timer mode register 25)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F021C	TMR26 (Timer mode register 26)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F021E	TMR27 (Timer mode register 27)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0220	TSR20 (Timer status register 20)								-	-	R	-	-	-	-	-	-	-	-
					TSR20L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F0222	TSR21 (Timer status register 21)								-	-	R	-	-	-	-	-	-	-	-
					TSR21L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F0224	TSR22 (Timer status register 22)								-	-	R	-	-	-	-	-	-	-	-
					TSR22L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F0226	TSR23 (Timer status register 23)								-	-	R	-	-	-	-	-	-	-	-
					TSR23L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F0228	TSR24 (Timer status register 24)								-	-	R	-	-	-	-	-	-	-	-
					TSR24L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F022A	TSR25 (Timer status register 25)								-	-	R	-	-	-	-	-	-	-	-
					TSR25L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F022C	TSR26 (Timer status register 26)								-	-	R	-	-	-	-	-	-	-	-
					TSR26L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F022E	TSR27 (Timer status register 27)								-	-	R	-	-	-	-	-	-	-	-
					TSR27L								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	F0230	TE2 (Timer channel enable status register 2)								-	-	R	-	-	-	-	-	-	-	-
					TE2L								R	R	-	R	R	R	R	R	R	R	R
					TE2_7	TE2_6	TE2_5	TE2_4	TE2_3	TE2_2	TE2_1	TE2_0				R	R	R	R	R	R	R	R
-	-	-	-	F0232	TS2 (Timer channel start trigger register 2)								-	-	E	-	-	-	-	-	-	-	-
					TS2L								E	E	-	E	E	E	E	E	E	E	E
					TS2_7	TS2_6	TS2_5	TS2_4	TS2_3	TS2_2	TS2_1	TS2_0				E	E	E	E	E	E	E	E
-	-	-	-	F0234	TT2 (Timer channel stop trigger register 2)								-	-	E	-	-	-	-	-	-	-	-
					TT2L								E	E	-	E	E	E	E	E	E	E	E
					TT2_7	TT2_6	TT2_5	TT2_4	TT2_3	TT2_2	TT2_1	TT2_0				E	E	E	E	E	E	E	E
-	-	-	-	F0236	TPS2 (Timer clock select register 2)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0238	TO2 (Timer output register 2)								-	-	E	-	-	-	-	-	-	-	-
					TO2L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F023A	TOE2 (Timer output enable register 2)								-	-	E	-	-	-	-	-	-	-	-
					TOE2L								E	E	-	E	E	E	E	E	E	E	E
					TOE2_7	TOE2_6	TOE2_5	TOE2_4	TOE2_3	TOE2_2	TOE2_1	TOE2_0				E	E	E	E	E	E	E	E
-	-	-	-	F023C	TOL2 (Timer output level register 2)								-	-	E	-	-	-	-	-	-	-	-
					TOL2L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F023E	TOM2 (Timer output mode register 2)								-	-	E	-	-	-	-	-	-	-	-
					TOM2L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0240	UF0CTL0 (LIN-UART0 control register 0)								E	E	-	R	E	E	E	E	E	E	E
						UF0TXE	UF0RXE	UF0DIR	UF0PS1	UF0PS0	UF0CL	UF0SL				-	E	E	E	E	E	E	E
-	-	-	-	F0241	UF0OPT0 (LIN-UART0 option control register 0)								E	E	-	R	E	E	E	E	E	E	E
					UF0BRF	UF0BRT	UF0BTT	UF0BLS2	UF0BLS1	UF0BLS0	UF0TDL	UF0RDL				R	E	E	E	E	E	E	E
-	-	-	-	F0242	UF0CTL1 (LIN-UART0 control register 1)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0244	UF0OPT1 (LIN-UART0 option control register 1)								E	E	-	E	E	E	E	E	E	E	E
					UF0EBE	UF0EBL	UF0EBC	UF0IPCS	UF0ACE	UF0MD1	UF0MD0	UF0DCS				E	E	E	E	E	E	E	E
-	-	-	-	F0245	UF0OPT2 (LIN-UART0 option control register 2)								E	E	-	R	R	R	R	R	R	E	E
											UF0RXFL	UF0ITS				-	-	-	-	-	-	E	E
-	-	-	-	F0246	UF0STR (LIN-UART0 status register)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0248	UF0STC (LIN-UART0 status clear register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F024A	UF0WTX (LIN-UART0 wait transmit data register)								-	-	E	-	-	-	-	-	-	-	-
					UF0WTXB (LIN-UART0 8-bit wait transmit data register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F024E	UF0ID (LIN-UART0 ID setting register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F024F	UF0BUF0 (LIN-UART0 buffer register 0)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0250	UF0BUF1 (LIN-UART0 buffer register 1)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0251	UF0BUF2 (LIN-UART0 buffer register 2)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0252	UF0BUF3 (LIN-UART0 buffer register 3)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0253	UF0BUF4 (LIN-UART0 buffer register 4)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0254	UF0BUF5 (LIN-UART0 buffer register 5)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0255	UF0BUF6 (LIN-UART0 buffer register 6)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0256	UF0BUF7 (LIN-UART0 buffer register 7)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	F0257	UF0BUF8 (LIN-UART0 buffer register 8)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0258	UF0BUCTL (LIN-UART0 buffer control register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0260	UF1CTL0 (LIN-UART1 control register 0)								E	E	-	R	E	E	E	E	E	E	E
						UF1TXE	UF1RXE	UF1DIR	UF1PS1	UF1PS0	UF1CL	UF1SL				-	E	E	E	E	E	E	E
-	-	-	-	F0261	UF1OPT0 (LIN-UART1 option control register 0)								E	E	-	R	E	E	E	E	E	E	E
					UF1BRF	UF1BRT	UF1BTT	UF1BLS2	UF1BLS1	UF1BLS0	UF1TDL	UF1RDL				R	E	E	E	E	E	E	E
-	-	-	-	F0262	UF1CTL1 (LIN-UART1 control register 1)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F0264	UF1OPT1 (LIN-UART1 option control register 1)								E	E	-	E	E	E	E	E	E	E	E
					UF1EBE	UF1EBL	UF1EBC	UF1IPCS	UF1ACE	UF1MD1	UF1MD0	UF1DCS				E	E	E	E	E	E	E	E
-	-	-	-	F0265	UF1OPT2 (LIN-UART0 option control register 2)								E	E	-	R	R	R	R	R	R	E	E
											UF1RXFL	UF1ITS				-	-	-	-	-	-	E	E
-	-	-	-	F0266	UF1STR (LIN-UART1 status register)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	F0268	UF1STC (LIN-UART1 status clear register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	F026A	UF1WTX (LIN-UART1 wait transmit data register)								-	-	E	-	-	-	-	-	-	-	-
					UF1WTXB (LIN-UART1 8-bit wait transmit data register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F026E	UF1ID (LIN-UART1 ID setting register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F026F	UF1BUF0 (LIN-UART1 buffer register 0)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0270	UF1BUF1 (LIN-UART1 buffer register 1)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0271	UF1BUF2 (LIN-UART1 buffer register 2)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0272	UF1BUF3 (LIN-UART1 buffer register 3)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0273	UF1BUF4 (LIN-UART1 buffer register 4)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0274	UF1BUF5 (LIN-UART1 buffer register 5)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0275	UF1BUF6 (LIN-UART1 buffer register 6)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0276	UF1BUF7 (LIN-UART1 buffer register 7)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0277	UF1BUF8 (LIN-UART1 buffer register 8)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	F0278	UF1BUCTL (LIN-UART1 buffer control register)								-	-	E	-	-	-	-	-	-	-	-
1	1	7	7	F0280	SG0FL (Frequency register SG0FL)								-	-	E	-	-	-	-	-	-	-	-
1	1	7	7	F0282	SG0FH (Frequency register SG0FH)								-	-	E	-	-	-	-	-	-	-	-
1	1	7	7	F0284	SG0PWM (Amplitude register)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F0286	SG0SDF (Duration factor register)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
1	1	1	1	F0287	SG0CTL (Control register)								E	E	-	R	R	R	E	R	R	E	E
								SG0PWR			SG0OS	SG0ALDS				-	-	-	E	-	-	E	E
1	1	1	1	F0288	SG0ITH (Interrupt threshold register)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F02F0	CRC0CTL (Flash memory CRC control register)								E	E	-	E	R	E	E	E	E	E	E
					CRC0EN		FEA5	FEA4	FEA3	FEA2	FEA1	FEA0				E	-	E	E	E	E	E	E
1	1	1	1	F02F2	PGCRCL (Flash memory CRC operation result register)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F02FA	CRCD (CRC data register)								-	-	E	-	-	-	-	-	-	-	-
1	1	1	1	F0300	SEG0 (LCD display data memory0)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0301	SEG1 (LCD display data memory1)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0302	SEG2 (LCD display data memory2)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0303	SEG3 (LCD display data memory3)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0304	SEG4 (LCD display data memory4)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0305	SEG5 (LCD display data memory5)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0306	SEG6 (LCD display data memory6)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0307	SEG7 (LCD display data memory7)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0308	SEG8 (LCD display data memory8)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0309	SEG9 (LCD display data memory9)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F030A	SEG10 (LCD display data memory10)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F030B	SEG11 (LCD display data memory11)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F030C	SEG12 (LCD display data memory12)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F030D	SEG13 (LCD display data memory13)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F030E	SEG14 (LCD display data memory14)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F030F	SEG15 (LCD display data memory15)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0310	SEG16 (LCD display data memory16)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0311	SEG17 (LCD display data memory17)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0312	SEG18 (LCD display data memory18)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0313	SEG19 (LCD display data memory19)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0314	SEG20 (LCD display data memory20)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0315	SEG21 (LCD display data memory21)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0316	SEG22 (LCD display data memory22)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0317	SEG23 (LCD display data memory23)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
1	1	1	1	F0318	SEG24 (LCD display data memory24)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0319	SEG25 (LCD display data memory25)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F031A	SEG26 (LCD display data memory26)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F031B	SEG27 (LCD display data memory27)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F031C	SEG28 (LCD display data memory28)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F031D	SEG29 (LCD display data memory29)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F031E	SEG30 (LCD display data memory30)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F031F	SEG31 (LCD display data memory31)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0320	SEG32 (LCD display data memory32)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0321	SEG33 (LCD display data memory33)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0322	SEG34 (LCD display data memory34)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0323	SEG35 (LCD display data memory35)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0324	SEG36 (LCD display data memory36)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0325	SEG37 (LCD display data memory37)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0326	SEG38 (LCD display data memory38)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0327	SEG39 (LCD display data memory39)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0328	SEG40 (LCD display data memory40)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0329	SEG41 (LCD display data memory41)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F032A	SEG42 (LCD display data memory42)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F032B	SEG43 (LCD display data memory43)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F032C	SEG44 (LCD display data memory44)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F032D	SEG45 (LCD display data memory45)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F032E	SEG46 (LCD display data memory46)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F032F	SEG47 (LCD display data memory47)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0330	SEG48 (LCD display data memory48)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0331	SEG49 (LCD display data memory49)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0332	SEG50 (LCD display data memory50)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0333	SEG51 (LCD display data memory51)								-	E	-	-	-	-	-	-	-	-	-
1	1	1	1	F0334	SEG52 (LCD display data memory52)								-	E	-	-	-	-	-	-	-	-	-
3	3	3	3	F0340	C1GMCTRL (CAN1 global module control register)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0342	C1GMCS (CAN1 global module clock select register)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
3	3	3	3	F0346	C1GMABT (CAN1 global block transmission controlregister)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0348	C1GMABTD (CAN1 global block transmission delay setting register)								-	E	-	-	-	-	-	-	-	-	-
3	3	3	3	F0380	C1MASK1L (CAN1 module mask 1 register L)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0382	C1MASK1H (CAN1 module mask 1 register H)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0384	C1MASK2L (CAN1 module mask 2 register L)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0386	C1MASK2H (CAN1 module mask 2 register H)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0388	C1MASK3L (CAN1 module mask 3 register L)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F038A	C1MASK3H (CAN1 module mask 3 register H)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F038C	C1MASK4L (CAN1 module mask 4 register L)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F038E	C1MASK4H (CAN1 module mask 4 register H)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0390	C1CTRL (CAN1 module control register)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0392	C1LEC (CAN1 module last error information registe)								-	E	-	-	-	-	-	-	-	-	-
3	3	3	3	F0393	C1INFO (CAN1 module information register)								-	R	-	-	-	-	-	-	-	-	-
3	3	3	3	F0394	C1ERC (CAN1 module error counter register)								-	-	R	-	-	-	-	-	-	-	-
3	3	3	3	F0396	C1IE (CAN1 module interrupt enable register)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F0398	C1INTS (CAN1 module interrupt status register)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F039A	C1BRP (CAN1 module bit rate prescaler register)								-	E	-	-	-	-	-	-	-	-	-
3	3	3	3	F039C	C1BTR (CAN1 module bit rate register)								-	-	E	-	-	-	-	-	-	-	-
3	3	3	3	F039E	C1LIPT (CAN1 module last in-pointer register)								-	R	-	-	-	-	-	-	-	-	-
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK})				F03A0	C1RGPT (CAN1 module receive history list register)								-	-	E	-	-	-	-	-	-	-	-
				F03A2	C1LOPT (CAN1 module last out-pointer register)								-	R	-	-	-	-	-	-	-	-	-
				F03A4	C1TGPT (CAN1 module transmit history list register)								-	-	E	-	-	-	-	-	-	-	-
				F03A6	C1TS (CAN1 module time stamp register)								-	-	E	-	-	-	-	-	-	-	-
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0400	C1MDB0100 (CAN1 message data byte 01 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB000								-	E	-	-	-	-	-	-	-	-	-
				F0401	C1MDB100								-	E	-	-	-	-	-	-	-	-	-
				F0402	C1MDB2300 (CAN1 message data byte 23 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB200								-	E	-	-	-	-	-	-	-	-	-
				F0403	C1MDB300								-	E	-	-	-	-	-	-	-	-	-
				F0404	C1MDB4500 (CAN1 message data byte 45 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB400								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0405	C1MDB500								-	E	-	-	-	-	-	-	-	-	-
				F0406	C1MDB6700 (CAN1 message data byte 67 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB600								-	E	-	-	-	-	-	-	-	-	-
				F0407	C1MDB700								-	E	-	-	-	-	-	-	-	-	-
				F0408	C1MDLC00 (CAN1 message data length register 00)								-	E	-	-	-	-	-	-	-	-	-
				F0409	C1MCONF00 (CAN1 message Configuration register 00)								-	E	-	-	-	-	-	-	-	-	-
				F040A	C1MIDL00 (CAN1 message ID register 00L)								-	-	E	-	-	-	-	-	-	-	-
				F040C	C1MIDH00 (CAN1 message ID register 00H)								-	-	E	-	-	-	-	-	-	-	-
				F040E	C1MCTRL00 (CAN1 message control register 00)								-	-	E	-	-	-	-	-	-	-	-
				F0410	C1MDB0101 (CAN1 message data byte 01 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB001								-	E	-	-	-	-	-	-	-	-	-
				F0411	C1MDB101								-	E	-	-	-	-	-	-	-	-	-
				F0412	C1MDB2301 (CAN1 message data byte 23 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB201								-	E	-	-	-	-	-	-	-	-	-
				F0413	C1MDB301								-	E	-	-	-	-	-	-	-	-	-
				F0414	C1MDB4501 (CAN1 message data byte 45 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB401								-	E	-	-	-	-	-	-	-	-	-
				F0415	C1MDB501								-	E	-	-	-	-	-	-	-	-	-
				F0416	C1MDB6701 (CAN1 message data byte 67 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB601								-	E	-	-	-	-	-	-	-	-	-
				F0417	C1MDB701								-	E	-	-	-	-	-	-	-	-	-
				F0418	C1MDLC01 (CAN1 message data length register 01)								-	E	-	-	-	-	-	-	-	-	-
				F0419	C1MCONF01 (CAN1 message Configuration register 01)								-	E	-	-	-	-	-	-	-	-	-
				F041A	C1MIDL01 (CAN1 message ID register 01L)								-	-	E	-	-	-	-	-	-	-	-
				F041C	C1MIDH01 (CAN1 message ID register 01H)								-	-	E	-	-	-	-	-	-	-	-
				F041E	C1MCTRL01 (CAN1 message control register 01)								-	-	E	-	-	-	-	-	-	-	-
				F0420	C1MDB0102 (CAN1 message data byte 01 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB002								-	E	-	-	-	-	-	-	-	-	-
				F0421	C1MDB102								-	E	-	-	-	-	-	-	-	-	-
				F0422	C1MDB2302 (CAN1 message data byte 23 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB202								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0423	C1MDB302								-	E	-	-	-	-	-	-	-	-	-
				F0424	C1MDB4502 (CAN1 message data byte 45 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB402								-	E	-	-	-	-	-	-	-	-	-
				F0425	C1MDB502								-	E	-	-	-	-	-	-	-	-	-
				F0426	C1MDB6702 (CAN1 message data byte 67 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB602								-	E	-	-	-	-	-	-	-	-	-
				F0427	C1MDB702								-	E	-	-	-	-	-	-	-	-	-
				F0428	C1MDLC02 (CAN1 message data length register 02)								-	E	-	-	-	-	-	-	-	-	-
				F0429	C1MCONF02 (CAN1 message Configuration register 02)								-	E	-	-	-	-	-	-	-	-	-
				F042A	C1MIDL02 (CAN1 message ID register 02L)								-	-	E	-	-	-	-	-	-	-	-
				F042C	C1MIDH02 (CAN1 message ID register 02H)								-	-	E	-	-	-	-	-	-	-	-
				F042E	C1MCTRL02 (CAN1 message control register 02)								-	-	E	-	-	-	-	-	-	-	-
				F0430	C1MDB0103 (CAN1 message data byte 01 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB003								-	E	-	-	-	-	-	-	-	-	-
				F0431	C1MDB103								-	E	-	-	-	-	-	-	-	-	-
				F0432	C1MDB2303 (CAN1 message data byte 23 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB203								-	E	-	-	-	-	-	-	-	-	-
				F0433	C1MDB303								-	E	-	-	-	-	-	-	-	-	-
				F0434	C1MDB4503 (CAN1 message data byte 45 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB403								-	E	-	-	-	-	-	-	-	-	-
				F0435	C1MDB503								-	E	-	-	-	-	-	-	-	-	-
				F0436	C1MDB6703 (CAN1 message data byte 67 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB603								-	E	-	-	-	-	-	-	-	-	-
				F0437	C1MDB703								-	E	-	-	-	-	-	-	-	-	-
				F0438	C1MDLC03 (CAN1 message data length register 03)								-	E	-	-	-	-	-	-	-	-	-
				F0439	C1MCONF03 (CAN1 message Configuration register 03)								-	E	-	-	-	-	-	-	-	-	-
				F043A	C1MIDL03 (CAN1 message ID register 03L)								-	-	E	-	-	-	-	-	-	-	-
				F043C	C1MIDH03 (CAN1 message ID register 03H)								-	-	E	-	-	-	-	-	-	-	-
				F043E	C1MCTRL03 (CAN1 message control register 03)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F0440	C1MDB0104 (CAN1 message data byte 01 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB004								-	E	-	-	-	-	-	-	-	-	-
				F0441	C1MDB104								-	E	-	-	-	-	-	-	-	-	-
				F0442	C1MDB2304 (CAN1 message data byte 23 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB204								-	E	-	-	-	-	-	-	-	-	-
				F0443	C1MDB304								-	E	-	-	-	-	-	-	-	-	-
				F0444	C1MDB4504 (CAN1 message data byte 45 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB404								-	E	-	-	-	-	-	-	-	-	-
				F0445	C1MDB504								-	E	-	-	-	-	-	-	-	-	-
				F0446	C1MDB6704 (CAN1 message data byte 67 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB604								-	E	-	-	-	-	-	-	-	-	-
				F0447	C1MDB704								-	E	-	-	-	-	-	-	-	-	-
				F0448	C1MDLC04 (CAN1 message data length register 04)								-	E	-	-	-	-	-	-	-	-	-
				F0449	C1MCONF04 (CAN1 message Configuration register 04)								-	E	-	-	-	-	-	-	-	-	-
				F044A	C1MIDL04 (CAN1 message ID register 04L)								-	-	E	-	-	-	-	-	-	-	-
				F044C	C1MIDH04 (CAN1 message ID register 04H)								-	-	E	-	-	-	-	-	-	-	-
				F044E	C1MCTRL04 (CAN1 message control register 04)								-	-	E	-	-	-	-	-	-	-	-
				F0450	C1MDB0105 (CAN1 message data byte 01 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB005								-	E	-	-	-	-	-	-	-	-	-
				F0451	C1MDB105								-	E	-	-	-	-	-	-	-	-	-
				F0452	C1MDB2305 (CAN1 message data byte 23 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB205								-	E	-	-	-	-	-	-	-	-	-
				F0453	C1MDB305								-	E	-	-	-	-	-	-	-	-	-
				F0454	C1MDB4505 (CAN1 message data byte 45 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB405								-	E	-	-	-	-	-	-	-	-	-
				F0455	C1MDB505								-	E	-	-	-	-	-	-	-	-	-
				F0456	C1MDB6705 (CAN1 message data byte 67 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB605								-	E	-	-	-	-	-	-	-	-	-
				F0457	C1MDB705								-	E	-	-	-	-	-	-	-	-	-
				F0458	C1MDLC05 (CAN1 message data length register 05)								-	E	-	-	-	-	-	-	-	-	-
				F0459	C1MCONF05 (CAN1 message Configuration register 05)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F045A	C1MIDL05 (CAN1 message ID register 05L)								-	-	E	-	-	-	-	-	-	-	-
				F045C	C1MIDH05 (CAN1 message ID register 05H)								-	-	E	-	-	-	-	-	-	-	-
				F045E	C1MCTRL05 (CAN1 message control register 05)								-	-	E	-	-	-	-	-	-	-	-
				F0460	C1MDB0106 (CAN1 message data byte 01 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB006								-	E	-	-	-	-	-	-	-	-	-
				F0461	C1MDB106								-	E	-	-	-	-	-	-	-	-	-
				F0462	C1MDB2306 (CAN1 message data byte 23 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB206								-	E	-	-	-	-	-	-	-	-	-
				F0463	C1MDB306								-	E	-	-	-	-	-	-	-	-	-
				F0464	C1MDB4506 (CAN1 message data byte 45 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB406								-	E	-	-	-	-	-	-	-	-	-
				F0465	C1MDB506								-	E	-	-	-	-	-	-	-	-	-
				F0466	C1MDB6706 (CAN1 message data byte 67 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB606								-	E	-	-	-	-	-	-	-	-	-
				F0467	C1MDB706								-	E	-	-	-	-	-	-	-	-	-
				F0468	C1MDLC06 (CAN1 message data length register 06)								-	E	-	-	-	-	-	-	-	-	-
				F0469	C1MCONF06 (CAN1 message Configuration register 06)								-	E	-	-	-	-	-	-	-	-	-
				F046A	C1MIDL06 (CAN1 message ID register 06L)								-	-	E	-	-	-	-	-	-	-	-
				F046C	C1MIDH06 (CAN1 message ID register 06H)								-	-	E	-	-	-	-	-	-	-	-
				F046E	C1MCTRL06 (CAN1 message control register 06)								-	-	E	-	-	-	-	-	-	-	-
				F0470	C1MDB0107 (CAN1 message data byte 01 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB007								-	E	-	-	-	-	-	-	-	-	-
				F0471	C1MDB107								-	E	-	-	-	-	-	-	-	-	-
				F0472	C1MDB2307 (CAN1 message data byte 23 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB207								-	E	-	-	-	-	-	-	-	-	-
				F0473	C1MDB307								-	E	-	-	-	-	-	-	-	-	-
				F0474	C1MDB4507 (CAN1 message data byte 45 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB407								-	E	-	-	-	-	-	-	-	-	-
				F0475	C1MDB507								-	E	-	-	-	-	-	-	-	-	-
				F0476	C1MDB6707 (CAN1 message data byte 67 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB607								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F0477	C1MDB707								-	E	-	-	-	-	-	-	-	-	-
				F0478	C1MDLC07 (CAN1 message data length register 07)								-	E	-	-	-	-	-	-	-	-	-
				F0479	C1MCONF07 (CAN1 message Configuration register 07)								-	E	-	-	-	-	-	-	-	-	-
				F047A	C1MIDL07 (CAN1 message ID register 07L)								-	-	E	-	-	-	-	-	-	-	-
				F047C	C1MIDH07 (CAN1 message ID register 07H)								-	-	E	-	-	-	-	-	-	-	-
				F047E	C1MCTRL07 (CAN1 message control register 07)								-	-	E	-	-	-	-	-	-	-	-
				F0480	C1MDB0108 (CAN1 message data byte 01 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB008								-	E	-	-	-	-	-	-	-	-	-
				F0481	C1MDB108								-	E	-	-	-	-	-	-	-	-	-
				F0482	C1MDB2308 (CAN1 message data byte 23 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB208								-	E	-	-	-	-	-	-	-	-	-
				F0483	C1MDB308								-	E	-	-	-	-	-	-	-	-	-
				F0484	C1MDB4508 (CAN1 message data byte 45 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB408								-	E	-	-	-	-	-	-	-	-	-
				F0485	C1MDB508								-	E	-	-	-	-	-	-	-	-	-
				F0486	C1MDB6708 (CAN1 message data byte 67 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB608								-	E	-	-	-	-	-	-	-	-	-
				F0487	C1MDB708								-	E	-	-	-	-	-	-	-	-	-
				F0488	C1MDLC08 (CAN1 message data length register 08)								-	E	-	-	-	-	-	-	-	-	-
				F0489	C1MCONF08 (CAN1 message Configuration register 08)								-	E	-	-	-	-	-	-	-	-	-
				F048A	C1MIDL08 (CAN1 message ID register 08L)								-	-	E	-	-	-	-	-	-	-	-
				F048C	C1MIDH08 (CAN1 message ID register 08H)								-	-	E	-	-	-	-	-	-	-	-
				F048E	C1MCTRL08 (CAN1 message control register 08)								-	-	E	-	-	-	-	-	-	-	-
				F0490	C1MDB0109 (CAN1 message data byte 01 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB009								-	E	-	-	-	-	-	-	-	-	-
				F0491	C1MDB109								-	E	-	-	-	-	-	-	-	-	-
				F0492	C1MDB2309 (CAN1 message data byte 23 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB209								-	E	-	-	-	-	-	-	-	-	-
				F0493	C1MDB309								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0494	C1MDB4509 (CAN1 message data byte 45 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB409								-	E	-	-	-	-	-	-	-	-	-
				F0495	C1MDB509								-	E	-	-	-	-	-	-	-	-	-
				F0496	C1MDB6709 (CAN1 message data byte 67 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB609								-	E	-	-	-	-	-	-	-	-	-
				F0497	C1MDB709								-	E	-	-	-	-	-	-	-	-	-
				F0498	C1MDLC09 (CAN1 message data length register 09)								-	E	-	-	-	-	-	-	-	-	-
				F0499	C1MCONF09 (CAN1 message Configuration register 09)								-	E	-	-	-	-	-	-	-	-	-
				F049A	C1MIDL09 (CAN1 message ID register 09L)								-	-	E	-	-	-	-	-	-	-	-
				F049C	C1MIDH09 (CAN1 message ID register 09H)								-	-	E	-	-	-	-	-	-	-	-
				F049E	C1MCTRL09 (CAN1 message control register 09)								-	-	E	-	-	-	-	-	-	-	-
				F04A0	C1MDB0110 (CAN1 message data byte 01 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB010								-	E	-	-	-	-	-	-	-	-	-
				F04A1	C1MDB110								-	E	-	-	-	-	-	-	-	-	-
				F04A2	C1MDB2310 (CAN1 message data byte 23 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB210								-	E	-	-	-	-	-	-	-	-	-
				F04A3	C1MDB310								-	E	-	-	-	-	-	-	-	-	-
				F04A4	C1MDB4510 (CAN1 message data byte 45 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB410								-	E	-	-	-	-	-	-	-	-	-
				F04A5	C1MDB510								-	E	-	-	-	-	-	-	-	-	-
				F04A6	C1MDB6710 (CAN1 message data byte 67 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB610								-	E	-	-	-	-	-	-	-	-	-
				F04A7	C1MDB710								-	E	-	-	-	-	-	-	-	-	-
				F04A8	C1MDLC10 (CAN1 message data length register 10)								-	E	-	-	-	-	-	-	-	-	-
				F04A9	C1MCONF10 (CAN1 message Configuration register 10)								-	E	-	-	-	-	-	-	-	-	-
				F04AA	C1MIDL10 (CAN1 message ID register 10L)								-	-	E	-	-	-	-	-	-	-	-
				F04AC	C1MIDH10 (CAN1 message ID register 10H)								-	-	E	-	-	-	-	-	-	-	-
				F04AE	C1MCTRL10 (CAN1 message control register 10)								-	-	E	-	-	-	-	-	-	-	-
				F04B0	C1MDB0111 (CAN1 message data byte 01 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB011								-	E	-	-	-	-	-	-	-	-	-
				F04B1	C1MDB111								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F04B2	C1MDB2311 (CAN1 message data byte 23 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB211								-	E	-	-	-	-	-	-	-	-	-
				F04B3	C1MDB311								-	E	-	-	-	-	-	-	-	-	-
				F04B4	C1MDB4511 (CAN1 message data byte 45 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB411								-	E	-	-	-	-	-	-	-	-	-
				F04B5	C1MDB511								-	E	-	-	-	-	-	-	-	-	-
				F04B6	C1MDB6711 (CAN1 message data byte 67 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB611								-	E	-	-	-	-	-	-	-	-	-
				F04B7	C1MDB711								-	E	-	-	-	-	-	-	-	-	-
				F04B8	C1MDLC11 (CAN1 message data length register 11)								-	E	-	-	-	-	-	-	-	-	-
				F04B9	C1MCONF11 (CAN1 message Configuration register 11)								-	E	-	-	-	-	-	-	-	-	-
				F04BA	C1MIDL11 (CAN1 message ID register 11L)								-	-	E	-	-	-	-	-	-	-	-
				F04BC	C1MIDH11 (CAN1 message ID register 11H)								-	-	E	-	-	-	-	-	-	-	-
				F04BE	C1MCTRL11 (CAN1 message control register 11)								-	-	E	-	-	-	-	-	-	-	-
				F04C0	C1MDB0112 (CAN1 message data byte 01 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB012								-	E	-	-	-	-	-	-	-	-	-
				F04C1	C1MDB112								-	E	-	-	-	-	-	-	-	-	-
				F04C2	C1MDB2312 (CAN1 message data byte 23 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB212								-	E	-	-	-	-	-	-	-	-	-
				F04C3	C1MDB312								-	E	-	-	-	-	-	-	-	-	-
				F04C4	C1MDB4512 (CAN1 message data byte 45 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB412								-	E	-	-	-	-	-	-	-	-	-
				F04C5	C1MDB512								-	E	-	-	-	-	-	-	-	-	-
				F04C6	C1MDB6712 (CAN1 message data byte 67 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB612								-	E	-	-	-	-	-	-	-	-	-
				F04C7	C1MDB712								-	E	-	-	-	-	-	-	-	-	-
				F04C8	C1MDLC12 (CAN1 message data length register 12)								-	E	-	-	-	-	-	-	-	-	-
				F04C9	C1MCONF12 (CAN1 message Configuration register 12)								-	E	-	-	-	-	-	-	-	-	-
				F04CA	C1MIDL12 (CAN1 message ID register 12L)								-	-	E	-	-	-	-	-	-	-	-
				F04CC	C1MIDH12 (CAN1 message ID register 12H)								-	-	E	-	-	-	-	-	-	-	-
				F04CE	C1MCTRL12 (CAN1 message control register 12)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F04D0	C1MDB0113 (CAN1 message data byte 01 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB013								-	E	-	-	-	-	-	-	-	-	-
				F04D1	C1MDB113								-	E	-	-	-	-	-	-	-	-	-
				F04D2	C1MDB2313 (CAN1 message data byte 23 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB213								-	E	-	-	-	-	-	-	-	-	-
				F04D3	C1MDB313								-	E	-	-	-	-	-	-	-	-	-
				F04D4	C1MDB4513 (CAN1 message data byte 45 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB413								-	E	-	-	-	-	-	-	-	-	-
				F04D5	C1MDB513								-	E	-	-	-	-	-	-	-	-	-
				F04D6	C1MDB6713 (CAN1 message data byte 67 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB613								-	E	-	-	-	-	-	-	-	-	-
				F04D7	C1MDB713								-	E	-	-	-	-	-	-	-	-	-
				F04D8	C1MDLC13 (CAN1 message data length register 13)								-	E	-	-	-	-	-	-	-	-	-
				F04D9	C1MCONF13 (CAN1 message Configuration register 13)								-	E	-	-	-	-	-	-	-	-	-
				F04DA	C1MIDL13 (CAN1 message ID register 13L)								-	-	E	-	-	-	-	-	-	-	-
				F04DC	C1MIDH13 (CAN1 message ID register 13H)								-	-	E	-	-	-	-	-	-	-	-
				F04DE	C1MCTRL13 (CAN1 message control register 13)								-	-	E	-	-	-	-	-	-	-	-
				F04E0	C1MDB0114 (CAN1 message data byte 01 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB014								-	E	-	-	-	-	-	-	-	-	-
				F04E1	C1MDB114								-	E	-	-	-	-	-	-	-	-	-
				F04E2	C1MDB2314 (CAN1 message data byte 23 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB214								-	E	-	-	-	-	-	-	-	-	-
				F04E3	C1MDB314								-	E	-	-	-	-	-	-	-	-	-
				F04E4	C1MDB4514 (CAN1 message data byte 45 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB414								-	E	-	-	-	-	-	-	-	-	-
				F04E5	C1MDB514								-	E	-	-	-	-	-	-	-	-	-
				F04E6	C1MDB6714 (CAN1 message data byte 67 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB614								-	E	-	-	-	-	-	-	-	-	-
				F04E7	C1MDB714								-	E	-	-	-	-	-	-	-	-	-
				F04E8	C1MDLC14 (CAN1 message data length register 14)								-	E	-	-	-	-	-	-	-	-	-
				F04E9	C1MCONF14 (CAN1 message Configuration register 14)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F04EA	C1MIDL14 (CAN1 message ID register 14L)								-	-	E	-	-	-	-	-	-	-	-
				F04EC	C1MIDH14 (CAN1 message ID register 14H)								-	-	E	-	-	-	-	-	-	-	-
				F04EE	C1MCTRL14 (CAN1 message control register 14)								-	-	E	-	-	-	-	-	-	-	-
				F04F0	C1MDB0115 (CAN1 message data byte 01 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB015								-	E	-	-	-	-	-	-	-	-	-
				F04F1	C1MDB115								-	E	-	-	-	-	-	-	-	-	-
				F04F2	C1MDB2315 (CAN1 message data byte 23 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB215								-	E	-	-	-	-	-	-	-	-	-
				F04F3	C1MDB315								-	E	-	-	-	-	-	-	-	-	-
				F04F4	C1MDB4515 (CAN1 message data byte 45 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB415								-	E	-	-	-	-	-	-	-	-	-
				F04F5	C1MDB515								-	E	-	-	-	-	-	-	-	-	-
				F04F6	C1MDB6715 (CAN1 message data byte 67 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C1MDB615								-	E	-	-	-	-	-	-	-	-	-
				F04F7	C1MDB715								-	E	-	-	-	-	-	-	-	-	-
				F04F8	C1MDLC15 (CAN1 message data length register 15)								-	E	-	-	-	-	-	-	-	-	-
				F04F9	C1MCONF15 (CAN1 message Configuration register 15)								-	E	-	-	-	-	-	-	-	-	-
				F04FA	C1MIDL15 (CAN1 message ID register 15L)								-	-	E	-	-	-	-	-	-	-	-
				F04FC	C1MIDH15 (CAN1 message ID register 15H)								-	-	E	-	-	-	-	-	-	-	-
				F04FE	C1MCTRL15 (CAN1 message control register 15)								-	-	E	-	-	-	-	-	-	-	-
				F05C0	C0GMCTRL (CAN0 global module control register)								-	-	E	-	-	-	-	-	-	-	-
				F05C6	C0GMABT (CAN0 global block transmission control register)								-	-	E	-	-	-	-	-	-	-	-
				F05C8	C0GMABTD (CAN0 global block transmission delay setting register)								-	E	-	-	-	-	-	-	-	-	-
				F05CE	C0GMCS (CAN0 global module clock select register)								-	E	-	-	-	-	-	-	-	-	-
				F05D0	C0MASK1L (CAN0 module mask 1 register L)								-	-	E	-	-	-	-	-	-	-	-
				F05D2	C0MASK1H (CAN0 module mask 1 register H)								-	-	E	-	-	-	-	-	-	-	-
				F05D4	C0MASK2L (CAN0 module mask 2 register L)								-	-	E	-	-	-	-	-	-	-	-
				F05D6	C0MASK2H (CAN0 module mask 2 register H)								-	-	E	-	-	-	-	-	-	-	-
				F05D8	C0MASK3L (CAN0 module mask 3 register L)								-	-	E	-	-	-	-	-	-	-	-
				F05DA	C0MASK3H (CAN0 module mask 3 register H)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F05DC	C0MASK4L (CAN0 module mask 4 register L)								-	-	E	-	-	-	-	-	-	-	-
				F05DE	C0MASK4H (CAN0 module mask 4 register H)								-	-	E	-	-	-	-	-	-	-	-
				F05E0	C0CTRL (CAN0 module control register)								-	-	E	-	-	-	-	-	-	-	-
				F05E2	C0LEC (CAN0 module last error information registe)								-	E	-	-	-	-	-	-	-	-	-
				F05E3	C0INFO (CAN0 module information register)								-	R	-	-	-	-	-	-	-	-	-
				F05E4	C0ERC (CAN0 module error counter register)								-	-	R	-	-	-	-	-	-	-	-
				F05E6	C0IE (CAN0 module interrupt enable register)								-	-	E	-	-	-	-	-	-	-	-
				F05E8	C0INTS (CAN0 module interrupt status register)								-	-	E	-	-	-	-	-	-	-	-
				F05EA	C0BRP (CAN0 module bit rate prescaler register)								-	E	-	-	-	-	-	-	-	-	-
				F05EC	C0BTR (CAN0 module bit rate register)								-	-	E	-	-	-	-	-	-	-	-
				F05EE	C0LIPT (CAN0 module last in-pointer register)								-	R	-	-	-	-	-	-	-	-	-
				F05F0	C0RGPT (CAN0 module receive history list register)								-	-	E	-	-	-	-	-	-	-	-
				F05F2	C0LOPT (CAN0 module last out-pointer register)								-	R	-	-	-	-	-	-	-	-	-
				F05F4	C0TGPT (CAN0 module transmit history list register)								-	-	E	-	-	-	-	-	-	-	-
				F05F6	C0TS (CAN0 module time stamp register)								-	-	E	-	-	-	-	-	-	-	-
				F0600	C0MDB0100 (CAN0 message data byte 01 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB000								-	E	-	-	-	-	-	-	-	-	-
				F0601	C0MDB100								-	E	-	-	-	-	-	-	-	-	-
				F0602	C0MDB2300 (CAN0 message data byte 23 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB200								-	E	-	-	-	-	-	-	-	-	-
				F0603	C0MDB300								-	E	-	-	-	-	-	-	-	-	-
				F0604	C0MDB4500 (CAN0 message data Byte 45 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB400								-	E	-	-	-	-	-	-	-	-	-
				F0605	C0MDB500								-	E	-	-	-	-	-	-	-	-	-
				F0606	C0MDB6700 (CAN0 message data byte 67 register 00)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB600								-	E	-	-	-	-	-	-	-	-	-
				F0607	C0MDB700								-	E	-	-	-	-	-	-	-	-	-
				F0608	C0MDLC00 (CAN0 message data length register 00)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F0609	C0MCONF00 (CAN0 message configuration register 00)								-	E	-	-	-	-	-	-	-	-	-
				F060A	C0MIDL00 (CAN0 message ID register 00L)								-	-	E	-	-	-	-	-	-	-	-
				F060C	C0MIDH00 (CAN0 message ID register 00H)								-	-	E	-	-	-	-	-	-	-	-
				F060E	C0MCTRL00 (CAN0 message control register 00)								-	-	E	-	-	-	-	-	-	-	-
				F0610	C0MDB0101 (CAN0 message data byte 01 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB001								-	E	-	-	-	-	-	-	-	-	-
				F0611	C0MDB101								-	E	-	-	-	-	-	-	-	-	-
				F0612	C0MDB2301 (CAN0 message data byte 23 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB201								-	E	-	-	-	-	-	-	-	-	-
				F0613	C0MDB301								-	E	-	-	-	-	-	-	-	-	-
				F0614	C0MDB4501 (CAN0 message data byte 45 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB401								-	E	-	-	-	-	-	-	-	-	-
				F0615	C0MDB501								-	E	-	-	-	-	-	-	-	-	-
				F0616	C0MDB6701 (CAN0 message data byte 67 register 01)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB601								-	E	-	-	-	-	-	-	-	-	-
				F0617	C0MDB701								-	E	-	-	-	-	-	-	-	-	-
				F0618	C0MDLC01 (CAN0 message data length register 01)								-	E	-	-	-	-	-	-	-	-	-
				F0619	C0MCONF01 (CAN0 message configuration register 01)								-	E	-	-	-	-	-	-	-	-	-
				F061A	C0MIDL01 (CAN0 message ID register 01L)								-	-	E	-	-	-	-	-	-	-	-
				F061C	C0MIDH01 (CAN0 message ID register 01H)								-	-	E	-	-	-	-	-	-	-	-
				F061E	C0MCTRL01 (CAN0 message control register 01)								-	-	E	-	-	-	-	-	-	-	-
				F0620	C0MDB0102 (CAN0 message data byte 01 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB002								-	E	-	-	-	-	-	-	-	-	-
				F0621	C0MDB102								-	E	-	-	-	-	-	-	-	-	-
				F0622	C0MDB2302 (CAN0 message data byte 23 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB202								-	E	-	-	-	-	-	-	-	-	-
				F0623	C0MDB302								-	E	-	-	-	-	-	-	-	-	-
				F0624	C0MDB4502 (CAN0 message data byte 45 register 02)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0625	C0MDB502								-	E	-	-	-	-	-	-	-	-	-
				F0626	C0MDB6702 (CAN0 message data byte 67 register 02)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB602								-	E	-	-	-	-	-	-	-	-	-
				F0627	C0MDB702								-	E	-	-	-	-	-	-	-	-	-
				F0628	C0MDLC02 (CAN0 message data length register 02)								-	E	-	-	-	-	-	-	-	-	-
				F0629	C0MCONF02 (CAN0 message configuration register 02)								-	E	-	-	-	-	-	-	-	-	-
				F062A	C0MIDL02 (CAN0 message ID register 02L)								-	-	E	-	-	-	-	-	-	-	-
				F062C	C0MIDH02 (CAN0 message ID register 02H)								-	-	E	-	-	-	-	-	-	-	-
				F062E	C0MCTRL02 (CAN0 message control register 02)								-	-	E	-	-	-	-	-	-	-	-
				F0630	C0MDB0103 (CAN0 message data byte 01 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB003								-	E	-	-	-	-	-	-	-	-	-
				F0631	C0MDB103								-	E	-	-	-	-	-	-	-	-	-
				F0632	C0MDB2303 (CAN0 message data byte 23 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB203								-	E	-	-	-	-	-	-	-	-	-
				F0633	C0MDB303								-	E	-	-	-	-	-	-	-	-	-
				F0634	C0MDB4503 (CAN0 message data byte 45 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB403								-	E	-	-	-	-	-	-	-	-	-
				F0635	C0MDB503								-	E	-	-	-	-	-	-	-	-	-
				F0636	C0MDB6703 (CAN0 message data byte 67 register 03)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB603								-	E	-	-	-	-	-	-	-	-	-
				F0637	C0MDB703								-	E	-	-	-	-	-	-	-	-	-
				F0638	C0MDLC03 (CAN0 message data length register 03)								-	E	-	-	-	-	-	-	-	-	-
				F0639	C0MCONF03 (CAN0 message Configuration register 03)								-	E	-	-	-	-	-	-	-	-	-
				F063A	C0MIDL03 (CAN0 message ID register 03L)								-	-	E	-	-	-	-	-	-	-	-
				F063C	C0MIDH03 (CAN0 message ID register 03H)								-	-	E	-	-	-	-	-	-	-	-
				F063E	C0MCTRL03 (CAN0 message control register 03)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0640	C0MDB0104 (CAN0 message data byte 01 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB004								-	E	-	-	-	-	-	-	-	-	-
				F0641	C0MDB104								-	E	-	-	-	-	-	-	-	-	-
				F0642	C0MDB2304 (CAN0 message data byte 23 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB204								-	E	-	-	-	-	-	-	-	-	-
				F0643	C0MDB304								-	E	-	-	-	-	-	-	-	-	-
				F0644	C0MDB4504 (CAN0 message data byte 45 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB404								-	E	-	-	-	-	-	-	-	-	-
				F0645	C0MDB504								-	E	-	-	-	-	-	-	-	-	-
				F0646	C0MDB6704 (CAN0 message data byte 67 register 04)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB604								-	E	-	-	-	-	-	-	-	-	-
				F0647	C0MDB704								-	E	-	-	-	-	-	-	-	-	-
				F0648	C0MDLC04 (CAN0 message data length register 04)								-	E	-	-	-	-	-	-	-	-	-
				F0649	C0MCONF04 (CAN0 message Configuration register 04)								-	E	-	-	-	-	-	-	-	-	-
				F064A	C0MIDL04 (CAN0 message ID register 04L)								-	-	E	-	-	-	-	-	-	-	-
				F064C	C0MIDH04 (CAN0 message ID register 04H)								-	-	E	-	-	-	-	-	-	-	-
				F064E	C0MCTRL04 (CAN0 message control register 04)								-	-	E	-	-	-	-	-	-	-	-
				F0650	C0MDB0105 (CAN0 message data byte 01 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB005								-	E	-	-	-	-	-	-	-	-	-
				F0651	C0MDB105								-	E	-	-	-	-	-	-	-	-	-
				F0652	C0MDB2305 (CAN0 message data byte 23 register 05)								-	-	E	-	-	-	-	-	-	-	-
				F0653	C0MDB305								-	E	-	-	-	-	-	-	-	-	-
				F0654	C0MDB4505 (CAN0 message data byte 45 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB405								-	E	-	-	-	-	-	-	-	-	-
				F0655	C0MDB505								-	E	-	-	-	-	-	-	-	-	-
				F0656	C0MDB6705 (CAN0 message data byte 67 register 05)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB605								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F0657	C0MDB705								-	E	-	-	-	-	-	-	-	-	-
				F0658	C0MDLC05 (CAN0 message data length register 05)								-	E	-	-	-	-	-	-	-	-	-
				F0659	C0MCONF05 (CAN0 message Configuration register 05)								-	E	-	-	-	-	-	-	-	-	-
				F065A	C0MIDL05 (CAN0 message ID register 05L)								-	-	E	-	-	-	-	-	-	-	-
				F065C	C0MIDH05 (CAN0 message ID register 05H)								-	-	E	-	-	-	-	-	-	-	-
				F065E	C0MCTRL05 (CAN0 message control register 05)								-	-	E	-	-	-	-	-	-	-	-
				F0660	C0MDB0106 (CAN0 message data byte 01 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB006								-	E	-	-	-	-	-	-	-	-	-
				F0661	C0MDB106								-	E	-	-	-	-	-	-	-	-	-
				F0662	C0MDB2306 (CAN0 message data byte 23 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB206								-	E	-	-	-	-	-	-	-	-	-
				F0663	C0MDB306								-	E	-	-	-	-	-	-	-	-	-
				F0664	C0MDB4506 (CAN0 message data byte 45 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB406								-	E	-	-	-	-	-	-	-	-	-
				F0665	C0MDB506								-	E	-	-	-	-	-	-	-	-	-
				F0666	C0MDB6706 (CAN0 message data byte 67 register 06)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB606								-	E	-	-	-	-	-	-	-	-	-
				F0667	C0MDB706								-	E	-	-	-	-	-	-	-	-	-
				F0668	C0MDLC06 (CAN0 message data length register 06)								-	E	-	-	-	-	-	-	-	-	-
				F0669	C0MCONF06 (CAN0 message Configuration register 06)								-	E	-	-	-	-	-	-	-	-	-
				F066A	C0MIDL06 (CAN0 message ID register 06L)								-	-	E	-	-	-	-	-	-	-	-
				F066C	C0MIDH06 (CAN0 message ID register 06H)								-	-	E	-	-	-	-	-	-	-	-
				F066E	C0MCTRL06 (CAN0 message control register 06)								-	-	E	-	-	-	-	-	-	-	-
				F0670	C0MDB0107 (CAN0 message data byte 01 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB007								-	E	-	-	-	-	-	-	-	-	-
				F0671	C0MDB107								-	E	-	-	-	-	-	-	-	-	-
				F0672	C0MDB2307 (CAN0 message data byte 23 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB207								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F0673	C0MDB307								-	E	-	-	-	-	-	-	-	-	-
				F0674	C0MDB4507 (CAN0 message data byte 45 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB407								-	E	-	-	-	-	-	-	-	-	-
				F0675	C0MDB507								-	E	-	-	-	-	-	-	-	-	-
				F0676	C0MDB6707 (CAN0 message data byte 67 register 07)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB607								-	E	-	-	-	-	-	-	-	-	-
				F0677	C0MDB707								-	E	-	-	-	-	-	-	-	-	-
				F0678	C0MDLC07 (CAN0 message data length register 07)								-	E	-	-	-	-	-	-	-	-	-
				F0679	C0MCONF07 (CAN0 message Configuration register 07)								-	E	-	-	-	-	-	-	-	-	-
				F067A	C0MIDL07 (CAN0 message ID register 07L)								-	-	E	-	-	-	-	-	-	-	-
				F067C	C0MIDH07 (CAN0 message ID register 07H)								-	-	E	-	-	-	-	-	-	-	-
				F067E	C0MCTRL07 (CAN0 message control register 07)								-	-	E	-	-	-	-	-	-	-	-
				F0680	C0MDB0108 (CAN0 message data byte 01 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB008								-	E	-	-	-	-	-	-	-	-	-
				F0681	C0MDB108								-	E	-	-	-	-	-	-	-	-	-
				F0682	C0MDB2308 (CAN0 message data byte 23 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB208								-	E	-	-	-	-	-	-	-	-	-
				F0683	C0MDB308								-	E	-	-	-	-	-	-	-	-	-
				F0684	C0MDB4508 (CAN0 message data byte 45 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB408								-	E	-	-	-	-	-	-	-	-	-
				F0685	C0MDB508								-	E	-	-	-	-	-	-	-	-	-
				F0686	C0MDB6708 (CAN0 message data byte 67 register 08)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB608								-	E	-	-	-	-	-	-	-	-	-
				F0687	C0MDB708								-	E	-	-	-	-	-	-	-	-	-
				F0688	C0MDLC08 (CAN0 message data length register 08)								-	E	-	-	-	-	-	-	-	-	-
				F0689	C0MCONF08 (CAN0 message Configuration register 08)								-	E	-	-	-	-	-	-	-	-	-
				F068A	C0MIDL08 (CAN0 message ID register 08L)								-	-	E	-	-	-	-	-	-	-	-
				F068C	C0MIDH08 (CAN0 message ID register 08H)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F068E	C0MCTRL08 (CAN0 message control register 08)								-	-	E	-	-	-	-	-	-	-	-
				F0690	C0MDB0109 (CAN0 message data byte 01 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB009								-	E	-	-	-	-	-	-	-	-	-
				F0691	C0MDB109								-	E	-	-	-	-	-	-	-	-	-
				F0692	C0MDB2309 (CAN0 message data byte 23 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB209								-	E	-	-	-	-	-	-	-	-	-
				F0693	C0MDB309								-	E	-	-	-	-	-	-	-	-	-
				F0694	C0MDB4509 (CAN0 message data byte 45 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB409								-	E	-	-	-	-	-	-	-	-	-
				F0695	C0MDB509								-	E	-	-	-	-	-	-	-	-	-
				F0696	C0MDB6709 (CAN0 message data byte 67 register 09)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB609								-	E	-	-	-	-	-	-	-	-	-
				F0697	C0MDB709								-	E	-	-	-	-	-	-	-	-	-
				F0698	C0MDLC09 (CAN0 message data length register 09)								-	E	-	-	-	-	-	-	-	-	-
				F0699	C0MCONF09 (CAN0 message Configuration register 09)								-	E	-	-	-	-	-	-	-	-	-
				F069A	C0MIDL09 (CAN0 message ID register 09L)								-	-	E	-	-	-	-	-	-	-	-
				F069C	C0MIDH09 (CAN0 message ID register 09H)								-	-	E	-	-	-	-	-	-	-	-
				F069E	C0MCTRL09 (CAN0 message control register 09)								-	-	E	-	-	-	-	-	-	-	-
				F06A0	C0MDB0110 (CAN0 message data byte 01 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB010								-	E	-	-	-	-	-	-	-	-	-
				F06A1	C0MDB110								-	E	-	-	-	-	-	-	-	-	-
				F06A2	C0MDB2310 (CAN0 message data byte 23 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB210								-	E	-	-	-	-	-	-	-	-	-
				F06A3	C0MDB310								-	E	-	-	-	-	-	-	-	-	-
				F06A4	C0MDB4510 (CAN0 message data byte 45 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB410								-	E	-	-	-	-	-	-	-	-	-
				F06A5	C0MDB510								-	E	-	-	-	-	-	-	-	-	-
				F06A6	C0MDB6710 (CAN0 message data byte 67 register 10)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB610								-	E	-	-	-	-	-	-	-	-	-
				F06A7	C0MDB710								-	E	-	-	-	-	-	-	-	-	-
				F06A8	C0MDLC10 (CAN0 message data length register 10)								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F06A9	C0MCONF10 (CAN0 message Configuration register 10)								-	E	-	-	-	-	-	-	-	-	-
				F06AA	C0MIDL10 (CAN0 message ID register 10L)								-	-	E	-	-	-	-	-	-	-	-
				F06AC	C0MIDH10 (CAN0 message ID register 10H)								-	-	E	-	-	-	-	-	-	-	-
				F06AE	C0MCTRL10 (CAN0 message control register 10)								-	-	E	-	-	-	-	-	-	-	-
				F06B0	C0MDB0111 (CAN0 message data byte 01 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB011								-	E	-	-	-	-	-	-	-	-	-
				F06B1	C0MDB111								-	E	-	-	-	-	-	-	-	-	-
				F06B2	C0MDB2311 (CAN0 message data byte 23 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB211								-	E	-	-	-	-	-	-	-	-	-
				F06B3	C0MDB311								-	E	-	-	-	-	-	-	-	-	-
				F06B4	C0MDB4511 (CAN0 message data byte 45 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB411								-	E	-	-	-	-	-	-	-	-	-
				F06B5	C0MDB511								-	E	-	-	-	-	-	-	-	-	-
				F06B6	C0MDB6711 (CAN0 message data byte 67 register 11)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB611								-	E	-	-	-	-	-	-	-	-	-
				F06B7	C0MDB711								-	E	-	-	-	-	-	-	-	-	-
				F06B8	C0MDLC11 (CAN0 message data length register 11)								-	E	-	-	-	-	-	-	-	-	-
				F06B9	C0MCONF11 (CAN0 message Configuration register 11)								-	E	-	-	-	-	-	-	-	-	-
				F06BA	C0MIDL11 (CAN0 message ID register 11L)								-	-	E	-	-	-	-	-	-	-	-
				F06BC	C0MIDH11 (CAN0 message ID register 11H)								-	-	E	-	-	-	-	-	-	-	-
				F06BE	C0MCTRL11 (CAN0 message control register 11)								-	-	E	-	-	-	-	-	-	-	-
				F06C0	C0MDB0112 (CAN0 message data byte 01 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB012								-	E	-	-	-	-	-	-	-	-	-
				F06C1	C0MDB112								-	E	-	-	-	-	-	-	-	-	-
				F06C2	C0MDB2312 (CAN0 message data byte 23 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB212								-	E	-	-	-	-	-	-	-	-	-
				F06C3	C0MDB312								-	E	-	-	-	-	-	-	-	-	-
				F06C4	C0MDB4512 (CAN0 message data byte 45 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB412								-	E	-	-	-	-	-	-	-	-	-
				F06C5	C0MDB512								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/f_{CAN}) × 3/ (1/f_{CLK}) Max. (1/f_{CAN}) × 4/ (1/f_{CLK}) For 8 bit-write Min. (1/f_{CAN}) × 4/ (1/f_{CLK}) Max. (1/f_{CAN}) × 5/ (1/f_{CLK}) For 16 bit-write Min. (1/f_{CAN}) × 2/ (1/f_{CLK}) Max. (1/f_{CAN}) × 3/ (1/f_{CLK})				F06C6	C0MDB6712 (CAN0 message data byte 67 register 12)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB612								-	E	-	-	-	-	-	-	-	-	-
				F06C7	C0MDB712								-	E	-	-	-	-	-	-	-	-	-
				F06C8	C0MDLC12 (CAN0 message data length register 12)								-	E	-	-	-	-	-	-	-	-	-
				F06C9	C0MCONF12 (CAN0 message Configuration register 12)								-	E	-	-	-	-	-	-	-	-	-
				F06CA	C0MIDL12 (CAN0 message ID register 12L)								-	-	E	-	-	-	-	-	-	-	-
				F06CC	C0MIDH12 (CAN0 message ID register 12H)								-	-	E	-	-	-	-	-	-	-	-
				F06CE	C0MCTRL12 (CAN0 message control register 12)								-	-	E	-	-	-	-	-	-	-	-
				F06D0	C0MDB0113 (CAN0 message data byte 01 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB013								-	E	-	-	-	-	-	-	-	-	-
				F06D1	C0MDB113								-	E	-	-	-	-	-	-	-	-	-
				F06D2	C0MDB2313 (CAN0 message data byte 23 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB213								-	E	-	-	-	-	-	-	-	-	-
				F06D3	C0MDB313								-	E	-	-	-	-	-	-	-	-	-
				F06D4	C0MDB4513 (CAN0 message data byte 45 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB413								-	E	-	-	-	-	-	-	-	-	-
				F06D5	C0MDB513								-	E	-	-	-	-	-	-	-	-	-
				F06D6	C0MDB6713 (CAN0 message data byte 67 register 13)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB613								-	E	-	-	-	-	-	-	-	-	-
				F06D7	C0MDB713								-	E	-	-	-	-	-	-	-	-	-
				F06D8	C0MDLC13 (CAN0 message data length register 13)								-	E	-	-	-	-	-	-	-	-	-
				F06D9	C0MCONF13 (CAN0 message Configuration register 13)								-	E	-	-	-	-	-	-	-	-	-
				F06DA	C0MIDL13 (CAN0 message ID register 13L)								-	-	E	-	-	-	-	-	-	-	-
				F06DC	C0MIDH13 (CAN0 message ID register 13H)								-	-	E	-	-	-	-	-	-	-	-
				F06DE	C0MCTRL13 (CAN0 message control register 13)								-	-	E	-	-	-	-	-	-	-	-
				F06E0	C0MDB0114 (CAN0 message data byte 01 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB014								-	E	-	-	-	-	-	-	-	-	-
				F06E1	C0MDB114								-	E	-	-	-	-	-	-	-	-	-
				F06E2	C0MDB2314 (CAN0 message data byte 23 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB214								-	E	-	-	-	-	-	-	-	-	-
				F06E3	C0MDB314								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
Retry-number × 2 +1 where the retry-number is calculated according to the following method: (Abort under radix point) For read Min. (1/fCAN) × 3/ (1/fCLK) Max. (1/fCAN) × 4/ (1/fCLK) For 8 bit-write Min. (1/fCAN) × 4/ (1/fCLK) Max. (1/fCAN) × 5/ (1/fCLK) For 16 bit-write Min. (1/fCAN) × 2/ (1/fCLK) Max. (1/fCAN) × 3/ (1/fCLK)				F06E4	C0MDB4514 (CAN0 message data byte 45 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB414								-	E	-	-	-	-	-	-	-	-	-
				F06E5	C0MDB514								-	E	-	-	-	-	-	-	-	-	-
				F06E6	C0MDB6714 (CAN0 message data byte 67 register 14)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB614								-	E	-	-	-	-	-	-	-	-	-
				F06E7	C0MDB714								-	E	-	-	-	-	-	-	-	-	-
				F06E8	C0MDLC14 (CAN0 message data length register 14)								-	E	-	-	-	-	-	-	-	-	-
				F06E9	C0MCONF14 (CAN0 message Configuration register 14)								-	E	-	-	-	-	-	-	-	-	-
				F06EA	C0MIDL14 (CAN0 message ID register 14L)								-	-	E	-	-	-	-	-	-	-	-
				F06EC	C0MIDH14 (CAN0 message ID register 14H)								-	-	E	-	-	-	-	-	-	-	-
				F06EE	C0MCTRL14 (CAN0 message control register 14)								-	-	E	-	-	-	-	-	-	-	-
				F06F0	C0MDB0115 (CAN0 message data byte 01 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB015								-	E	-	-	-	-	-	-	-	-	-
				F06F1	C0MDB115								-	E	-	-	-	-	-	-	-	-	-
				F06F2	C0MDB2315 (CAN0 message data byte 23 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB215								-	E	-	-	-	-	-	-	-	-	-
				F06F3	C0MDB315								-	E	-	-	-	-	-	-	-	-	-
				F06F4	C0MDB4515 (CAN0 message data byte 45 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB415								-	E	-	-	-	-	-	-	-	-	-
				F06F5	C0MDB515								-	E	-	-	-	-	-	-	-	-	-
				F06F6	C0MDB6715 (CAN0 message data byte 67 register 15)								-	-	E	-	-	-	-	-	-	-	-
					C0MDB615								-	E	-	-	-	-	-	-	-	-	-
				F06F7	C0MDB715								-	E	-	-	-	-	-	-	-	-	-
				F06F8	C0MDLC15 (CAN0 message data length register 15)								-	E	-	-	-	-	-	-	-	-	-
				F06F9	C0MCONF15 (CAN0 message Configuration register 15)								-	E	-	-	-	-	-	-	-	-	-
				F06FA	C0MIDL15 (CAN0 message ID register 15L)								-	-	E	-	-	-	-	-	-	-	-
				F06FC	C0MIDH15 (CAN0 message ID register 15H)								-	-	E	-	-	-	-	-	-	-	-
				F06FE	C0MCTRL15 (CAN0 message control register 15)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFF00	P0 (Port register 0)								E	E	-	E	E	E	E	E	E	E	E
					P0_7	P0_6	P0_5	P0_4	P0_3	P0_2	P0_1	P0_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF01	P1 (Port register 1)								E	E	-	E	E	E	E	E	E	E	E
					P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF02	P2 (Port register 2)								E	E	-	E	E	E	E	E	E	E	E
					P2_7	P2_6	P2_5	P2_4	P2_3	P2_2	P2_1	P2_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF03	P3 (Port register 3)								E	E	-	E	E	E	E	E	E	E	E
					P3_7	P3_6	P3_5	P3_4	P3_3	P3_2	P3_1	P3_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF04	P4 (Port register 4)								E	E	-	R	R	R	R	R	R	R	E
												P4_0				-	-	-	-	-	-	-	E
-	-	-	-	FFF05	P5 (Port register 5)								E	E	-	E	E	E	E	E	E	E	E
					P5_7	P5_6	P5_5	P5_4	P5_3	P5_2	P5_1	P5_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF06	P6 (Port register 6)								E	E	-	R	E	E	E	E	E	E	E
						P6_6	P6_5	P6_4	P6_3	P6_2	P6_1	P6_0				-	E	E	E	E	E	E	E
-	-	-	-	FFF07	P7 (Port register 7)								E	E	-	R	R	E	E	E	E	E	E
							P7_5	P7_4	P7_3	P7_2	P7_1	P7_0				-	-	E	E	E	E	E	E
-	-	-	-	FFF08	P8 (Port register 8)								E	E	-	E	E	E	E	E	E	E	E
					P8_7	P8_6	P8_5	P8_4	P8_3	P8_2	P8_1	P8_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF09	P9 (Port register 9)								E	E	-	E	E	E	E	E	E	E	E
					P9_7	P9_6	P9_5	P9_4	P9_3	P9_2	P9_1	P9_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF0C	P12 (Port register 12)								R	R	-	R	R	R	R	R	R	R	R
								P12_4	P12_3	P12_2	P12_1					-	-	-	R	R	R	R	-
-	-	-	-	FFF0D	P13 (Port register 13)								E	E	-	R	E	E	E	E	E	E	E
					P13_7	P13_6	P13_5	P13_4	P13_3	P13_2	P13_1	P13_0				R	E	E	E	E	E	E	E
-	-	-	-	FFF0E	P14 (Port register 14)								E	E	-	R	R	R	R	R	R	R	E
												P14_0				-	-	-	-	-	-	-	E
-	-	-	-	FFF0F	P15 (Port register 15)								E	E	-	R	R	R	R	R	R	R	E
												P15_0				-	-	-	-	-	-	-	E
-	-	-	-	FFF10	SDR00 (Serial data register 00)								-	-	E	-	-	-	-	-	-	-	-
					SDR00L								-	E	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFF12	SDR01 (Serial data register 01)								-	-	E	-	-	-	-	-	-	-	-
					SDR01L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF14	SDR10 (Serial data register 10)								-	-	E	-	-	-	-	-	-	-	-
					SDR10L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF16	SDR11 (Serial data register 11)								-	-	E	-	-	-	-	-	-	-	-
					SDR11L								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF18	TDR00 (Timer data register 00)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF1A	TDR01 (Timer data register 01)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF1E	ADCR (10 bit A/D conversion result register)								-	-	R	-	-	-	-	-	-	-	-
-	-	-	-	FFF1F	ADCRH (8 bit A/D conversion result register)								-	R	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF20	PM0 (Port mode register 0)								E	E	-	E	E	E	E	E	E	E	E
					PM0_7	PM0_6	PM0_5	PM0_4	PM0_3	PM0_2	PM0_1	PM0_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF21	PM1 (Port mode register 1)								E	E	-	E	E	E	E	E	E	E	E
					PM1_7	PM1_6	PM1_5	PM1_4	PM1_3	PM1_2	PM1_1	PM1_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF22	PM2 (Port mode register 2)								E	E	-	E	E	E	E	E	E	E	E
					PM2_7	PM2_6	PM2_5	PM2_4	PM2_3	PM2_2	PM2_1	PM2_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF23	PM3 (Port mode register 3)								E	E	-	E	E	E	E	E	E	E	E
					PM3_7	PM3_6	PM3_5	PM3_4	PM3_3	PM3_2	PM3_1	PM3_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF24	PM4 (Port mode register 4)								E	E	-	R	R	R	R	R	R	R	E
												PM4_0				-	-	-	-	-	-	-	E
-	-	-	-	FFF25	PM5 (Port mode register 5)								E	E	-	E	E	E	E	E	E	E	E
					PM5_7	PM5_6	PM5_5	PM5_4	PM5_3	PM5_2	PM5_1	PM5_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF26	PM6 (Port mode register 6)								E	E	-	R	E	E	E	E	E	E	E
						PM6_6	PM6_5	PM6_4	PM6_3	PM6_2	PM6_1	PM6_0				-	E	E	E	E	E	E	E
-	-	-	-	FFF27	PM7 (Port mode register 7)								E	E	-	R	R	E	E	E	E	E	E
							PM7_5	PM7_4	PM7_3	PM7_2	PM7_1	PM7_0				-	-	E	E	E	E	E	E
-	-	-	-	FFF28	PM8 (Port mode register 8)								E	E	-	E	E	E	E	E	E	E	E
					PM8_7	PM8_6	PM8_5	PM8_4	PM8_3	PM8_2	PM8_1	PM8_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF29	PM9 (Port mode register 9)								E	E	-	E	E	E	E	E	E	E	E
					PM9_7	PM9_6	PM9_5	PM9_4	PM9_3	PM9_2	PM9_1	PM9_0				E	E	E	E	E	E	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFF2D	PM13 (Port mode register 13)								E	E	-	R	E	E	E	E	E	E	R
						PM13_6	PM13_5	PM13_4	PM13_3	PM13_2	PM13_1					-	E	E	E	E	E	E	-
-	-	-	-	FFF2E	PM14 (Port mode register 14)								E	E	-	R	R	R	R	R	R	R	E
												PM14_0				-	-	-	-	-	-	-	E
-	-	-	-	FFF2F	PM15 (Port mode register 15)								E	E	-	R	R	R	R	R	R	R	E
												PM15_0				-	-	-	-	-	-	-	E
-	-	-	-	FFF30	ADM0 (A/D converter mode register 0)								E	E	-	E	E	E	E	E	E	E	E
					ADCS	ADMD	FR2	FR1	FR0	LV1	LV0	ADCE				E	E	E	E	E	E	E	E
-	-	-	-	FFF31	ADS (Analog input channel specification register)								E	E	-	R	R	R	R	E	E	E	E
									ADS_3	ADS_2	ADS_1	ADS_0				-	-	-	-	E	E	E	E
-	-	-	-	FFF32	ADM1 (A/D converter mode register 1)								E	E	-	E	E	E	R	R	R	E	E
					ADTMD1	ADTMD0	ADSCM				ADTRS1	ADTRS0				E	E	E	-	-	-	E	E
-	-	-	-	FFF34	SUBCUDW (Watch error correction register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF36	RTCSEL (RTC1Hz pin select register)								E	E	-	E	E	R	R	E	E	E	E
					RTCOSEL1	RTCOSEL0			RTCTIS11	RTCTIS10	RTCTIS01	RTCTIS00				E	E	-	-	E	E	E	E
-	-	-	-	FFF37	SMPC (SM port mode control register)								E	E	-	E	E	E	E	E	E	E	E
					MOD4	MOD3	MOD2	MOD1	EN4	EN3	EN2	EN1				E	E	E	E	E	E	E	E
-	-	-	-	FFF38	EGP0 (External interrupt rising edge enable register 0)								E	E	-	E	E	E	E	E	E	E	E
					EGP0_7	EGP0_6	EGP0_5	EGP0_4	EGP0_3	EGP0_2	EGP0_1	EGP0_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF39	EGN0 (External interrupt falling edge enable register 0)								E	E	-	E	E	E	E	E	E	E	E
					EGN0_7	EGN0_6	EGN0_5	EGN0_4	EGN0_3	EGN0_2	EGN0_1	EGN0_0				E	E	E	E	E	E	E	E
-	-	-	-	FFF3C	STSEL0 (Serial communication pin select register 0)								E	E	-	R	E	R	E	E	E	E	E
						SCSI100		SCSI010	SCSI001	SCSI000	SUARTF1	SUARTF0				-	E	-	E	E	E	E	E
-	-	-	-	FFF3D	STSEL1 (Serial communication pin select register 1)								E	E	-	E	E	R	R	E	E	E	E
					SIIC1	SIIC0			SCAN1	SCAN0	TMCAN1	TMCAN0				E	E	-	-	E	E	E	E
-	-	-	-	FFF3E	TISELSE (Timer input select else register)								E	E	-	E	E	R	R	R	R	E	E
					TOTICON1	TOTICON0					TI05SEL1	TI05SEL0				E	E	-	-	-	-	E	E
-	-	-	-	FFF3F	SGSEL (Sound generator pin select register)								E	E	-	R	R	R	R	E	E	E	E
									PCLSEL	SGSEL_2	SGSEL_1	SGSEL_0				-	-	-	-	E	E	E	E
-	-	-	-	FFF40	LCDMD (LCD mode register)								E	E	-	R	R	E	E	R	R	R	R
							MDSET1	MDSET0								-	-	E	E	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFF41	LCDM (LCD display mode register)								E	E	-	E	E	R	R	R	E	R	E
					LCDON	SCOC				LCDM_2		LCDM_0				E	E	-	-	-	E	-	E
-	-	-	-	FFF42	LCDC0 (LCD clock control register)								E	E	-	R	E	E	E	R	E	E	E
						LCDC0_6	LCDC0_5	LCDC0_4		LCDC0_2	LCDC0_1	LCDC0_0				-	E	E	E	-	E	E	E
-	-	-	-	FFF48	UF0TX (LIN-UART0 transmit data register)								-	-	E	-	-	-	-	-	-	-	-
					UF0TXB (LIN-UART0 8-bit transmit data register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF4A	UF0RX (LIN-UART0 receive data register)								-	-	E	-	-	-	-	-	-	-	-
					UF0RXB (LIN-UART0 receive data register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF4C	UF1TX (LIN-UART1 transmit data register)								-	-	E	-	-	-	-	-	-	-	-
					UF1TXB (LIN-UART1 8-bit transmit data register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF4E	UF1RX (LIN-UART1 receive data register)								-	-	E	-	-	-	-	-	-	-	-
					UF1RXB (LIN-UART1 receive data register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF50	ITMC (Interval timer control register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF52	SEC (Second count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF53	MIN (Minute count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF54	HOUR (Hour count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF55	WEEK (Week count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF56	DAY (Day count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF57	MONTH (Month count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF58	YEAR (Year count register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF59	SUBCUD (Watch error correction register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF5A	ALARMWM (Alarm minute register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF5B	ALARMWH (Alarm hour register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF5C	ALARMWW (Alarm day register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFF5D	RTCC0 (Real time counter control register 0)								E	E	-	E	R	E	R	E	E	E	E
					RTCE		RCLOE1		AMPM	CT2	CT1	CT0				E	-	E	-	E	E	E	E
-	-	-	-	FFF5E	RTCC1 (Real time counter control register 1)								E	E	-	E	E	R	E	E	R	R	E
					WALE	WALIE		WAFG	RIFG		RWST	RWAIT				E	E	-	E	E	-	R	E
-	-	-	-	FFF64	TDR02 (Timer data register 02)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF66	TDR03 (Timer data register 03)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF68	TDR04 (Timer data register 04)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFF6A	TDR05 (Timer data register 05)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF6C	TDR06 (Timer data register 06)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF6E	TDR07 (Timer data register 07)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF70	TDR10 (Timer data register 10)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF72	TDR11 (Timer data register 11)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF74	TDR12 (Timer data register 12)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF76	TDR13 (Timer data register 13)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF78	TDR14 (Timer data register 14)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF7A	TDR15 (Timer data register 15)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF7C	TDR16 (Timer data register 16)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF7E	TDR17 (Timer data register 17)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF90	TDR20 (Timer data register 20)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF92	TDR21 (Timer data register 21)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF94	TDR22 (Timer data register 22)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF96	TDR23 (Timer data register 23)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF98	TDR24 (Timer data register 24)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF9A	TDR25 (Timer data register 25)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF9C	TDR26 (Timer data register 26)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFF9E	TDR27 (Timer data register 27)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFFA0	CMC (Clock operation mode control register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFA1	CSC (Clock operation status control register)								E	E	-	E	E	R	R	R	R	R	E
					MSTOP	XTSTOP						HIOSTOP				E	E	-	-	-	-	-	E
-	-	-	-	FFFA2	OSTC (Oscillation stabilization time counter status register)								R	R	-	R	R	R	R	R	R	R	R
					MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18				R	R	R	R	R	R	R	R
-	-	-	-	FFFA3	OSTS (Oscillation stabilization time select register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFA4	CKC (Clock control register)								E	E	-	R	E	R	E	R	R	R	R
					CLS	CSS	MCS	MCM0								R	E	R	E	-	-	-	-
-	-	-	-	FFFA5	CKS0 (Clock output select register 0)								E	E	-	E	R	R	R	E	E	E	E
					PCLOE0				CSEL0	CCS02	CCS01	CCS00				E	-	-	-	E	E	E	E
-	-	-	-	FFFA8	RESF (Reset control flag register)								-	R	-	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFFA9	LVIM (Low voltage detect register)								E	E	-	E	R	R	R	R	R	R	R
					LVISEN						LVIOMSK	LVIF				E	-	-	-	-	-	R	R
-	-	-	-	FFFAA	LVIS (Low voltage detect level select register)								E	E	-	E	R	R	R	R	R	R	E
					LVIMD							LVILV				E	-	-	-	-	-	-	E
-	-	-	-	FFFAB	WDTE (Watchdog timer enable register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFAC	CRCIN (CRC input register)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB0	DSA0 (DMA SFR address register 0)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB1	DSA1 (DMA SFR address register 1)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB2	DRA0 (DMA RAM address register 0)								-	-	E	-	-	-	-	-	-	-	-
					DRA0L (DMA RAM address register 0L)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB3	DRA0H (DMA RAM address register 0H)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB4	DRA1 (DMA RAM address register 1)								-	-	E	-	-	-	-	-	-	-	-
					DRA1L (DMA RAM address register 1L)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB5	DRA1H (DMA RAM address register 1H)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB6	DBC0 (DMA byte count register 0)								-	-	E	-	-	-	-	-	-	-	-
					DBC0L (DMA byte count register 0L)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB7	DBC0H (DMA byte count register 0H)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB8	DBC1 (DMA byte count register 1)								-	-	E	-	-	-	-	-	-	-	-
					DBC1L (DMA byte count register 1L)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFB9	DBC1H (DMA byte count register 1H)								-	E	-	-	-	-	-	-	-	-	-
-	-	-	-	FFFBA	DMC0 (DMA mode control register 0)								E	E	-	E	E	E	E	E	E	E	E
					STG0	DRS0	DS0	DWAIT0	IFC03	IFC02	IFC01	IFC00				E	E	E	E	E	E	E	E
-	-	-	-	FFFBB	DMC1 (DMA mode control register 1)								E	E	-	E	E	E	E	E	E	E	E
					STG1	DRS1	DS1	DWAIT1	IFC13	IFC12	IFC11	IFC10				E	E	E	E	E	E	E	E
-	-	-	-	FFFBC	DRC0 (DMA operation control register 0)								E	E	-	E	R	R	R	R	R	R	E
					DEN0							DST0				E	-	-	-	-	-	-	E
-	-	-	-	FFFBD	DRC1 (DMA operation control register 1)								E	E	-	E	R	R	R	R	R	R	E
					DEN1							DST1				E	-	-	-	-	-	-	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFFD0	IF2 (Interrupt request flag register 2)								-	-	E	-	-	-	-	-	-	-	-
					IF2L (Interrupt request flag register 2L)								E	E	-	E	E	E	E	E	E	E	E
					C0RECIF	C0WUPIF	C0ERRIF	C1WUPIF	C1ERRIF	TMIF07	TMIF06	TMIF05				E	E	E	E	E	E	E	E
-	-	-	-	FFFD1	IF2H (Interrupt request flag register 2H)								E	E	-	E	E	E	E	E	E	E	E
					FLIF	C1RECIF	MDIF	TMIF13	TMIF12	TMIF11	TMIF10	C0TRXIF				E	E	E	E	E	E	E	E
-	-	-	-	FFFD2	IF3 (Interrupt request flag register 3)								-	-	E	-	-	-	-	-	-	-	-
					IF3L (Interrupt request flag register 3L)								E	E	-	E	E	E	E	E	E	E	E
					TMIF22	TMIF21	TMIF20	TMIF17	TMIF16	TMIF15	TMIF14	C1TRXIF				E	E	E	E	E	E	E	E
-	-	-	-	FFFD3	IF3H (Interrupt request flag register 3H)								E	E	-	R	R	R	E	E	E	E	E
								DMAIF3	DMAIF2	TMIF26	TMIF24	TMIF23				-	-	-	E	E	E	E	E
-	-	-	-	FFFD4	MK2 (Interrupt mask flag register 2)								-	-	E	-	-	-	-	-	-	-	-
					MK2L (Interrupt mask flag register 2L)								E	E	-	E	E	E	E	E	E	E	E
					C0RECMK	C0WUPMK	C0ERRMK	C1WUPMK	C1ERRMK	TMMK07	TMMK06	TMMK05				E	E	E	E	E	E	E	E
-	-	-	-	FFFD5	MK2H (Interrupt mask flag register 2H)								E	E	-	E	E	E	E	E	E	E	E
					FLMK	C1RECMK	MDMK	TMMK13	TMMK12	TMMK11	TMMK10	C0TRXMK				E	E	E	E	E	E	E	E
-	-	-	-	FFFD6	MK3 (Interrupt mask flag register 3)								-	-	E	-	-	-	-	-	-	-	-
					MK3L (Interrupt mask flag register 3L)								E	E	-	E	E	E	E	E	E	E	E
					TMMK22	TMMK21	TMMK20	TMMK17	TMMK16	TMMK15	TMMK14	C1TRXMK				E	E	E	E	E	E	E	E
-	-	-	-	FFFD7	MK3H (Interrupt mask flag register 3H)								E	E	-	R	R	R	E	E	E	E	E
								DMAMK3	DMAMK2	TMMK26	TMMK24	TMMK23				-	-	-	E	E	E	E	E
-	-	-	-	FFFD8	PR02 (Priority specification flag register 02)								-	-	E	-	-	-	-	-	-	-	-
					PR02L (Priority specification flag register 02L)								E	E	-	E	E	E	E	E	E	E	E
					C0RECPR0	C0WUPPR0	C0ERRPR0	C1WUPPR0	C1ERRPR0	TMPR007	TMPR006	TMPR005				E	E	E	E	E	E	E	E
-	-	-	-	FFFD9	PR02H (Priority specification flag register 02H)								E	E	-	E	E	E	E	E	E	E	E
					FLPR0	C1RECPR0	MDPR0	TMPR013	TMPR012	TMPR011	TMPR010	C0TRXPR0				E	E	E	E	E	E	E	E
-	-	-	-	FFFDA	PR03 (Priority specification flag register 03)								-	-	E	-	-	-	-	-	-	-	-
					PR03L (Priority specification flag register 03L)								E	E	-	E	E	E	E	E	E	E	E
					TMPR022	TMPR021	TMPR020	TMPR017	TMPR016	TMPR015	TMPR014	C1TRXPR0				E	E	E	E	E	E	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFFDB	PR03H (Priority specification flag register 03H)								E	E	-	R	R	R	E	E	E	E	E
								DMAPR03	DMAPR02	TMPR026	TMPR024	TMPR023				-	-	-	E	E	E	E	E
-	-	-	-	FFFDC	PR12 (Priority specification flag register 12)								-	-	E	-	-	-	-	-	-	-	-
					PR12L (Priority specification flag register 12L)								E	E	-	E	E	E	E	E	E	E	E
					C0RECPR1	C0WUPPR1	C0ERRPR1	C1WUPPR1	C1ERRPR1	TMPR107	TMPR106	TMPR105				E	E	E	E	E	E	E	E
-	-	-	-	FFFDD	PR12H (Priority specification flag register 12H)								E	E	-	E	E	E	E	E	E	E	E
					FLPR1	C1RECPR1	MDPR1	TMPR113	TMPR112	TMPR111	TMPR110	C0TRXPR1				E	E	E	E	E	E	E	E
-	-	-	-	FFFDE	PR13 (Priority specification flag register 13)								-	-	E	-	-	-	-	-	-	-	-
					PR13L (Priority specification flag register 13L)								E	E	-	E	E	E	E	E	E	E	E
					TMPR122	TMPR121	TMPR120	TMPR117	TMPR116	TMPR115	TMPR114	C1TRXPR1				E	E	E	E	E	E	E	E
-	-	-	-	FFFDF	PR13H (Priority specification flag register 13H)								E	E	-	R	R	R	E	E	E	E	E
								DMAPR13	DMAPR12	TMPR126	TMPR124	TMPR123				-	-	-	E	E	E	E	E
-	-	-	-	FFFE0	IF0 (Interrupt request flag register 0)								-	-	E	-	-	-	-	-	-	-	-
					IF0L (Interrupt request flag register 0L)								E	E	-	E	E	E	E	E	E	E	E
					PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF				E	E	E	E	E	E	E	E
-	-	-	-	FFFE1	IF0H (Interrupt request flag register 0H)								E	E	-	E	E	E	E	E	E	E	E
					LTIF0	ITIF	RTCIF	DMAIF1	DMAIF0	CSIIF01	CSIIF00	CLMIF				E	E	E	E	E	E	E	E
-	-	-	-	FFFE2	IF1 (Interrupt request flag register 1)								-	-	E	-	-	-	-	-	-	-	-
					IF1L (Interrupt request flag register 1L)								E	E	-	E	E	E	E	E	E	E	E
					TMIF03	TMIF02	TMIF01	TMIF00	SGIF	PIFLR0	LSIF0	LRIF0				E	E	E	E	E	E	E	E
-	-	-	-	FFFE3	IF1H (Interrupt request flag register 1H)								E	E	-	E	E	E	E	E	E	E	E
					TMIF04	IICIF11	CSIIF10	PIFLR1	LSIF1	LRIF1	LTIF1	ADIF				E	E	E	E	E	E	E	E
-	-	-	-	FFFE4	MK0 (Interrupt mask flag register 0)								-	-	E	-	-	-	-	-	-	-	-
					MK0L (Interrupt mask flag register 0L)								E	E	-	E	E	E	E	E	E	E	E
					PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK				E	E	E	E	E	E	E	E
-	-	-	-	FFFE5	MK0H (Interrupt mask flag register 0H)								E	E	-	E	E	E	E	E	E	E	E
					LTMK0	ITMK	RTCMK	DMAMK1	DMAMK0	CSIMK01	CSIMK00	CLMMK				E	E	E	E	E	E	E	E

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFFE6	MK1 (Interrupt mask flag register 1)								-	-	E	-	-	-	-	-	-	-	-
					MK1L (Interrupt mask flag register 1L)								E	E	-	E	E	E	E	E	E	E	E
					TMMK03	TMMK02	TMMK01	TMMK00	SGMK	PMKLR0	LSMK0	LRMK0				E	E	E	E	E	E	E	E
-	-	-	-	FFFE7	MK1H (Interrupt mask flag register 1H)								E	E	-	E	E	E	E	E	E	E	E
					TMMK04	IICMK11	CSIMK10	PMKLR1	LSMK1	LRMK1	LTMK1	ADMK				E	E	E	E	E	E	E	E
-	-	-	-	FFFE8	PR00 (Priority specification flag register 00)								-	-	E	-	-	-	-	-	-	-	-
					PR00L (Priority specification flag register 00L)								E	E	-	E	E	E	E	E	E	E	E
					PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0				E	E	E	E	E	E	E	E
-	-	-	-	FFFE9	PR00H (Priority specification flag register 00H)								E	E	-	E	E	E	E	E	E	E	E
					LTPR00	ITPR0	RTCPR0	DMAPR01	DMAPR00	CSIPR001	CSIPR000	CLMPR0				E	E	E	E	E	E	E	E
-	-	-	-	FFFEA	PR01 (Priority specification flag register 01)								-	-	E	-	-	-	-	-	-	-	-
					PR01L (Priority specification flag register 01L)								E	E	-	E	E	E	E	E	E	E	E
					TMPR003	TMPR002	TMPR001	TMPR000	SGPR0	PPR0LR0	LSPR00	LRPR00				E	E	E	E	E	E	E	E
-	-	-	-	FFFEB	PR01H (Priority specification flag register 01H)								E	E	-	E	E	E	E	E	E	E	E
					TMPR004	IICPR011	CSIPR010	PPR0LR1	LSPR01	LRPR01	LTPR01	ADPR0				E	E	E	E	E	E	E	E
-	-	-	-	FFFEC	PR10 (Priority specification flag register 10)								-	-	E	-	-	-	-	-	-	-	-
					PR10L (Priority specification flag register 10L)								E	E	-	E	E	E	E	E	E	E	E
					PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1				E	E	E	E	E	E	E	E
-	-	-	-	FFFED	PR10H (Priority specification flag register 10H)								E	E	-	E	E	E	E	E	E	E	E
					LTPR10	ITPR1	RTCPR1	DMAPR11	DMAPR10	CSIPR101	CSIPR100	CLMPR1				E	E	E	E	E	E	E	E
-	-	-	-	FFFEE	PR11 (Priority specification flag register 11)								-	-	E	-	-	-	-	-	-	-	-
					PR11L (Priority specification flag register 11L)								E	E	-	E	E	E	E	E	E	E	E
					TMPR103	TMPR102	TMPR101	TMPR100	SGPR1	PPR1LR0	LSPR10	LRPR10				E	E	E	E	E	E	E	E
-	-	-	-	FFFEF	PR11H (Priority specification flag register 11H)								E	E	-	E	E	E	E	E	E	E	E
					TMPR104	IICPR111	CSIPR110	PPR1LR1	LSPR11	LRPR11	LTPR11	ADPR1				E	E	E	E	E	E	E	E
-	-	-	-	FFFF0	MDAL (Multiplication input data register A(L))								-	-	E	-	-	-	-	-	-	-	-
					MULA (Multiplication input data register A)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFFF2	MDAH (Multiplication input data register A(H))								-	-	E	-	-	-	-	-	-	-	-
					MULB (Multiplication input data register B)								-	-	E	-	-	-	-	-	-	-	-

Necessary WAIT				Address	I/O register(SFR) name								R/W			Bit R/W							
READ(MIN.)	READ(MAX.)	WRITE(MIN.)	WRITE(MAX.)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	1	8	16	7	6	5	4	3	2	1	0
-	-	-	-	FFFF4	MDBH (Multiplication input data register B(H))								-	-	E	-	-	-	-	-	-	-	-
					MULOH (Higher multiplication result storage register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFFF6	MDBL (Multiplication input data register B(L))								-	-	E	-	-	-	-	-	-	-	-
					MULOL (Lower multiplication result storage register)								-	-	E	-	-	-	-	-	-	-	-
-	-	-	-	FFFFE	PMC (Processor mode control register)								E	E	-	R	R	R	R	R	R	R	E
												MAA				-	-	-	-	-	-	-	E

REVISION HISTORY

Major Revisions in This Edition

(1/1)

Page	Description	Classification
Throughout	Addition of 128-pin products, and 384 KB and 512 KB of flash ROM products	(d)
	Deletion of HOCODIV register and line over SCK signal	(a)
CHAPTER 4 PORT FUNCTIONS		
p.291	Change of Figure 4-75. Format of STSEL0 Register	(a)
p.293	Change of Figure 4-77. Format of SGSEL Register	(a)
p.294	Change of Figure 4-78. Format of RTCSEL Register	(a)
CHAPTER 5 CLOCK GENERATOR		
p.316	Change of Figure 5-2. Block Diagram of PLL Circuit	(a)
CHAPTER 7 REAL-TIME CLOCK		
p.494	Change of Figure 7-4. Format of Watch Error Correction Register (SUBCUDW)	(a)
p.504	Change of Figure 7-15. Format of Watch Error Correction Register (SUBCUD)	(a)
p.514, 515	Change of 7.4.6 Example of watch error correction of real-time clock	(a)
CHAPTER 12 SERIAL ARRAY UNIT		
p.635	Addition of description in Figure 12-27. Initial Setting Procedure for Master Transmission	(c)
p.637	Addition of description in Figure 12-29. Procedure for Resuming Master Transmission	(c)
p.647	Addition of description in Figure 12-37. Procedure for Resuming Master Reception	(c)
p.648	Addition of description in Figure 12-38. Timing Chart of Master Reception (in Single-Reception Mode)	(c)
p.658	Addition of description in Figure 12-45. Procedure for Resuming Master Transmission/Reception	(c)
p.667	Addition of description in Figure 12-51. Initial Setting Procedure for Slave Transmission	(c)
p.669	Addition of description in Figure 12-53. Procedure for Resuming Slave Transmission	(c)
p.677	Addition of description in Figure 12-59. Initial Setting Procedure for Slave Reception	(c)
p.678	Addition of description in Figure 12-61. Procedure for Resuming Slave Reception	(c)
p.686	Addition of description in Figure 12-67. Procedure for Resuming Slave Transmission/Reception	(c)
p.721	Addition of description in Figure 12-91. Initial Setting Procedure for Address Field Transmission	(c)
p.748	Addition of Table 12-15. Relationship between register settings and pins (Channel 0 and 1 of unit 0: UART0)	(c)
CHAPTER 21 INTERRUPT FUNCTIONS		
p.1147	Change of title 21.3 (4) External interrupt rising edge enable register 0 (EGP0) and external interrupt falling edge enable register 0 (EGN0)	(c)
CHAPTER 25 VOLTAGE DETECTOR		
p.1214	Change of Figure 25-10. Delay from the Time LVD Reset Source is Generated until the Time LVD Reset Has been Generated or Released	(b)
CHAPTER 33 ELECTRICAL SPECIFICATIONS (J GRADE PRODUCT)		
Throughout	Addition and change of descriptions	(b) (c)
CHAPTER 34 ELECTRICAL SPECIFICATIONS (L GRADE PRODUCT)		
Throughout	Addition and change of descriptions	(b) (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

RL78/D1A User’s Manual: Hardware

Publication Date: Rev.0.04 Feb 22, 2013
Rev.1.10 Mar 23, 2015

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0

RL78/D1A

Renesas Electronics Corporation

R01UH0317EJ0110