To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

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

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

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

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

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

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

User's Manual

V850ES/SA2, V850ES/SA3

32-Bit Single-Chip Microcontrollers

Hardware

V850ES/SA2:	V850ES/SA3:
μPD703200	μPD703204
μPD703200Y	μPD703204Y
μPD703201	μPD70F3204
μPD703201Y	μPD70F3204Y
μPD70F3201	
μPD70F3201Y	

Document No. U15905EJ2V1UD00 (2nd edition)
Date Published August 2005 N CP(K)

Printed in Japan

[MEMO]

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 V_{IL} (MAX) and V_{IH} (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 V_{IL} (MAX) and V_{IH} (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 V_{DD} 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.

These commodities, technology or software, must be exported in accordance
with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of July, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "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.

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

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

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

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02.11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-265 40 10
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 87 200
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J05.6

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850ES/SA2 (μPD703200, 703200Y, 703201, 703201Y, 70F3201, 70F3201Y) and V850ES/SA3 (μPD703204, 703204Y, 70F3204, 70F3204Y) and design application systems using these products.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850ES/SA2 and V850ES/SA3 shown in the Organization below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (**V850ES Architecture User's Manual**).

Hardware	Architecture
• Pin functions	• Data types
• CPU function	• Register set
• On-chip peripheral functions	• Instruction format and instruction set
• Flash memory programming	• Interrupts and exceptions
• Electrical specifications	• Pipeline operation

How to Read This Manual

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

To learn the detailed functions of a register whose register name is known
→Refer to **APPENDIX A REGISTER INDEX**.

To understand the details of an instruction function
→Refer to the **V850ES Architecture User's Manual** available separately.

Register format
→The name of the bit whose number is in angle brackets (<>) in the figure of the register format of each register is defined as a reserved word in the device file.

To understand the overall functions of the V850ES/SA2 and V850ES/SA3
→Read this manual according to the **CONTENTS**.

To know the electrical specifications of the V850ES/SA2 and V850ES/SA3
→Refer to **CHAPTER 22 ELECTRICAL SPECIFICATIONS**.

The mask ★ shows the major revised points.

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representation: $\overline{xxx}$ (overscore over pin or signal name)

Memory map address: Higher addresses on the top and lower addresses on the bottom

Note: Footnote for item marked with **Note** in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numeric representation: Binary ... xxxx or xxxxB
Decimal ... xxxx
Hexadecimal ... xxxxH

Prefix indicating power of 2 (address space, memory capacity): K (kilo): 2^{10} = 1,024
M (mega): 2^{20} = $1,024^2$
G (giga): 2^{30} = $1,024^3$

Related Documents

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

Documents related to V850ES/SA2 and V850ES/SA3

Document Name	Document No.
V850ES Architecture User's Manual	U15943E
V850ES/SA2, V850ES/SA3 Hardware User's Manual	This manual
V850ES/SA2, V850ES/SA3 Application Note	U16764E

Documents related to development tools

Document Name		Document No.
IE-V850ES-G1 (In-Circuit Emulator)		U16313E
IE-703204-G1-EM1 (In-Circuit Emulator Option Board)		U16622E
CA850 C Compiler Package Ver. 2.50	Operation	U16053E
	C Language	U16054E
	Assembly Language	U16042E
PM plus Ver. 5.10		U16569E
ID850 Integrated Debugger Ver. 2.50	Operation	U16217E
RX850 Real-Time OS Ver. 3.13 or Later	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Real-Time OS Ver. 3.15	Basics	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Task Debugger Ver. 3.01		U13737E
RD850 Pro Task Debugger Ver. 3.01		U13916E
AZ850 System Performance Analyzer Ver. 3.20		U14410E
PG-FP4 Flash Memory Programmer		U15260E

CONTENTS

CHAPTER 1 INTRODUCTION

The V850ES/SA2 and V850ES/SA3 are low-power models of the NEC Electronics V850 Series of single-chip microcontrollers for real-time control.

1.1 Overview

The V850ES/SA2 and V850ES/SA3 are 32-bit single-chip microcontrollers that employ the V850ES CPU core and integrate peripheral functions such as ROM/RAM, timers/counters, serial interfaces, an A/D converter, a D/A converter, and a DMA controller.

These products are part of the V850/SA1 line of low-power microcontrollers, but employ the V850ES as the CPU and have additional peripheral functions such as a D/A converter and ROM correction. They also realized lower voltage and power consumption.

The V850ES/SA2 and V850ES/SA3 feature instructions ideal for digital servo control applications, such as multiplication instructions using a hardware multiplier, saturated operation instructions, and bit manipulation instructions, as well as basic instructions with a short real-time response speed and a 1-clock pitch. These microcontrollers can be used in real-time control systems requiring low power consumption, such as DVC and portable audio systems, with a high cost effectiveness.

Table 1-1 shows the products in the V850ES/SA2 and V850ES/SA3 lineup.

Table 1-1. V850ES/SA2 and V850ES/SA3 Product Lineup

Product Name		I^{2}C	ROM		RAM Size	Package
Commercial Name	Part Number		Type	Size		
★ V850ES/SA2	μPD703200	None	Mask ROM	128 KB	8 KB	100-pin TQFP (14 × 14)
★	μPD703200Y	Provided				
	μPD703201	None		256 KB	16 KB	
	μPD703201Y	Provided				
	μPD70F3201	None	Flash memory			
	μPD70F3201Y	Provided				
V850ES/SA3	μPD703204	None	Mask ROM	256 KB	16 KB	121-pin FBGA (12 × 12)
	μPD703204Y	Provided				
	μPD70F3204	None	Flash memory			
	μPD70F3204Y	Provided				

1.2 Features

○ Number of instructions	83
★ ○ Minimum instruction execution time	50 ns: Main clock = 20 MHz (μPD703200, 703201, 703204, 70F3201, 70F3204) 59 ns: Main clock = 17 MHz (μPD703200Y, 703201Y, 703204Y, 70F3201Y, 70F3204Y) 30.5 μs: Subclock = 32.768 kHz
○ General-purpose registers	32 bits × 32 registers
○ Instruction set	Signed multiplication (16 × 16 → 32): 1 to 2 clocks Signed multiplication (32 × 32 → 64): 1 to 5 clocks Saturated operation (with overflow/underflow detection function) 32-bit shift instruction: 1 clock Bit manipulation instruction Load/store instruction with long/short format
○ Memory space	64 MB linear address space (for program/data) External expansion: Up to 4 MB for V850ES/SA2 and 16 MB for V850ES/SA3 (of which 1 MB is used as internal ROM/RAM space) Memory block division function: 2, 2, 4, 8 MB (total: 4 blocks) Programmable wait function Idle state insertion function
○ External bus interface	Multiplexed bus/separate bus output selectable 8/16-bit data bus sizing function 4-space chip select function Wait functions • Programmable wait function • External wait function Idle state function Bus hold function
○ Internal memory	RAM: 16 KB (μPD703201, 703201Y, 70F3201, 70F3201Y, 703204, 703204Y, 70F3204, 70F3204Y) 8 KB (μPD703200, 703200Y) Mask ROM: 256 KB (μPD703201, 703201Y, 703204, 703204Y) 128 KB (μPD703200, 703200Y) Flash memory: 256 KB (μPD70F3201, 70F3201Y, 70F3204, 70F3204Y)

○ Interrupts/exceptions	External interrupts: 8 sources Internal interrupts: 30 sources (μPD703200, 703201, 70F3201) 31 sources (μPD703200Y, 703201Y, 70F3201Y) 31 sources (μPD703204, 70F3204) 32 sources (μPD703204Y, 70F3204Y) Software exception: 32 sources Exception trap: 1 source
○ I/O lines	Total: 82 (input ports: 14, I/O ports: 68) (V850ES/SA2) 102 (input ports: 18, I/O ports: 84) (V850ES/SA3)
○ Timer/counter	16-bit timer/event counter: 2 ch (PWM output) 8-bit timer/event counter: 4 ch (connectable in cascade)
○ Real-time counter (for watch)	Subclock/main clock operation: 1 ch Dedicated on-chip hardware counter for weeks, days, hours, minutes, and seconds Up to 4,095 weeks can be counted
○ Watchdog timer:	1 ch
○ Serial interface	Asynchronous serial interface (UART) Clocked serial interface (CSI) I^{2}C bus interface (I^{2}C) (μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only) CSI/UART: 1 ch UART: 1 ch CSI/I^{2}C: 1 ch CSI: 2 ch (V850ES/SA2), 3 ch (V850ES/SA3)
○ A/D converter	10-bit resolution: 12 ch (V850ES/SA2) 16 ch (V850ES/SA3)
○ D/A converter	8-bit resolution: 2 ch
○ DMA controller:	4 ch (transfer object: Internal memory, on-chip peripheral I/O, and external memory)
○ ROM correction:	4 places specifiable
○ Clock generator	Main clock/subclock operation CPU clock: 7 steps (f_X, $f_X/2$, $f_X/4$, $f_X/8$, $f_X/16$, $f_X/32$, f_{XT})
○ Power save function	HALT/IDLE/STOP mode
○ Package	100-pin plastic TQFP (fine pitch) (14 × 14) (V850ES/SA2) 121-pin plastic FBGA (12 × 12) (V850ES/SA3)

1.3 Application Fields

Mobile devices requiring low power consumption

Example DVC and portable audio systems

1.4 Ordering Information

1.4.1 V850ES/SA2

	Part Number	Package	Internal ROM
★	μPD703200GC-×××-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
★	μPD703200YGC-×××-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
★	μPD703201GC-×××-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
★	μPD703201YGC-×××-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
★	μPD70F3201GC-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
★	μPD70F3201YGC-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)	Flash memory (256 KB)

Remarks 1. ××× indicates ROM code suffix.

2. No ROMless model is available.

3. Products with -A at the end of the part number are lead-free products.

1.4.2 V850ES/SA3

	Part Number	Package	Internal ROM
★	μPD703204F1-×××-EA6-A	121-pin plastic FBGA (12 × 12)	Mask ROM (256 KB)
★	μPD703204YF1-×××-EA6-A	121-pin plastic FBGA (12 × 12)	Mask ROM (256 KB)
★	μPD70F3204F1-EA6-A	121-pin plastic FBGA (12 × 12)	Flash memory (256 KB)
★	μPD70F3204YF1-EA6-A	121-pin plastic FBGA (12 × 12)	Flash memory (256 KB)

Remarks 1. ××× indicates ROM code suffix.

2. No ROMless model is available.

3. Products with -A at the end of the part number are lead-free products.

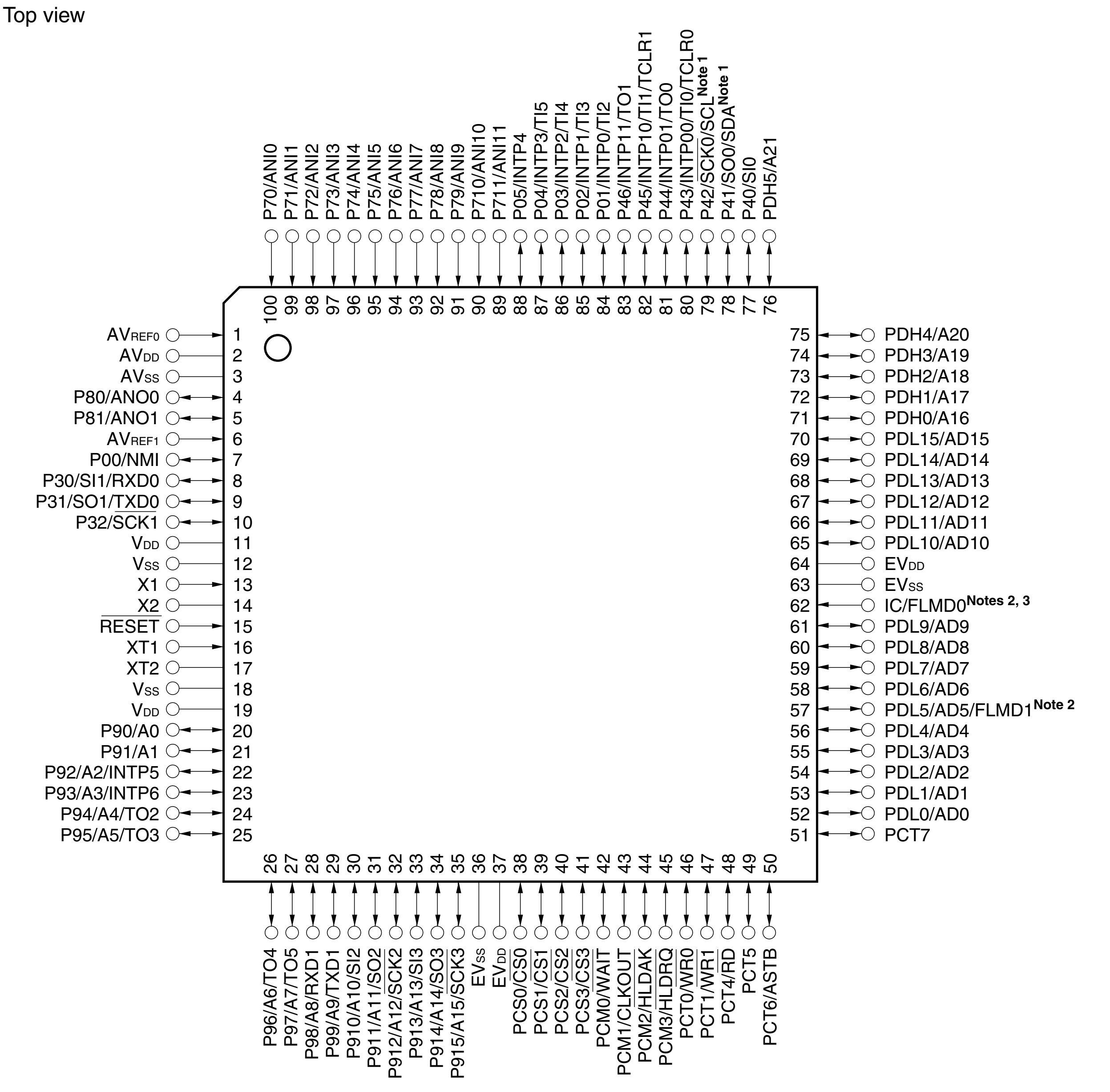
1.5 Pin Configuration

○ V850ES/SA2

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

★ • μPD703200GC-×××-YEU-A • μPD703201GC-×××-YEU-A • μPD70F3201GC-YEU-A

★ • μPD703200YGC-×××-YEU-A • μPD703201YGC-×××-YEU-A • μPD70F3201YGC-YEU-A



Notes 1. SCL and SDA are valid only in the μPD703200Y, 703201Y, and 70F3201Y.

2. FLMD0 and FLMD1 are valid only in the μPD70F3201 and 70F3201Y.

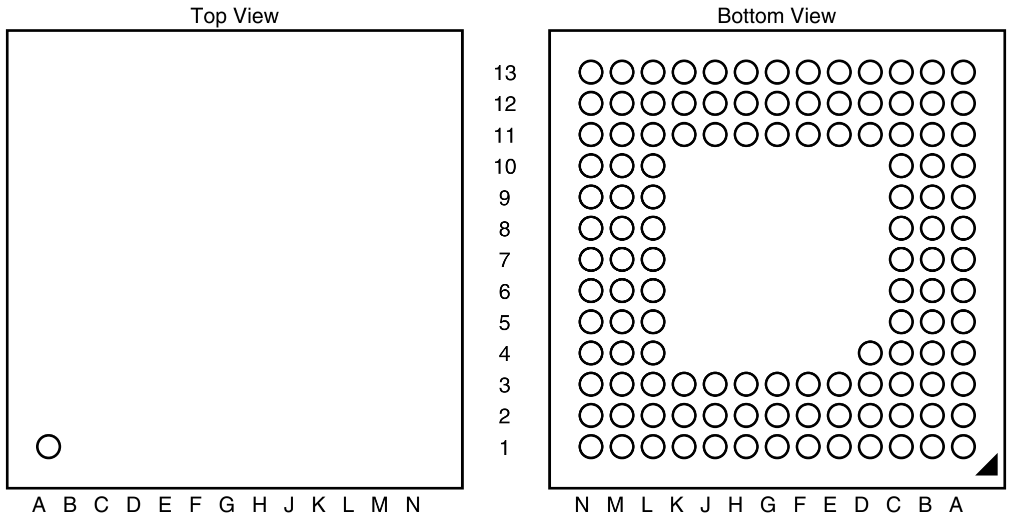
3. IC: Directly connect this pin to VSS (μPD703200, 703200Y, 703201 and 703201Y).
FLMD0: Connect this pin to VSS in the normal operation mode (μPD70F3201 and 70F3201Y).

○ V850ES/SA3

121-pin plastic FGBA (12 × 12)

★ μPD703204F1-×××-EA6-A μPD70F3204F1-EA6-A

★ μPD703204YF1-×××-EA6-A μPD70F3204YF1-EA6-A



Pin No.	Name	Pin No.	Name	Pin No.	Name
A1	P70/ANI0	B8	PCD3	D2	AV$_{REF1}$
A2	P71/ANI1	B9	P02/INTP1/TI3	D3	P00/NMI
A3	P73/ANI3	B10	P46/INTP11/TO1	D11	PDH0/A16
A4	P713/ANI13	B11	P42/$\overline{\text{SCK0}}$/SCL[Note]	D12	PDH2/A18
A5	P76/ANI6	B12	P40/SI0	D13	PDH1/A17
A6	P78/ANI8	B13	PDH4/A20	E1	P30/SI1/RXD0
A7	P711/ANI11	C1	P80/ANO0	E2	P31/SO1/TXD0
A8	P04/INTP3/TI5	C2	AV$_{SS}$	E3	P32/SCK1
A9	PCD2	C3	P74/ANI4	E11	PDL14/AD14
A10	P45/INTP10/TI1/TCLR1	C4	P714/ANI14	E12	PDH6/A22
A11	P43/INTP00/TI0/TCLR0	C5	P715/ANI15	E13	PDL15/AD15
A12	P41/SO0/SDA[Note]	C6	P79/ANI9	F1	V$_{SS}$
A13	PDH5/A21	C7	P05/INTP4	F2	X1
B1	AV$_{DD}$	C8	P03/INTP2/TI4	F3	V$_{DD}$
B2	AV$_{REF0}$	C9	PCD1	F11	PDL11/AD11
B3	P72/ANI2	C10	P01/INTP0/TI2	F12	PDL13/AD13
B4	P712/ANI12	C11	P44/INTP01/TO0	F13	PDL12/AD12
B5	P75/ANI5	C12	PDH3/A19	G1	$\overline{\text{RESET}}$
B6	P77/ANI7	C13	PDH7/A23	G2	XT1
B7	P710/ANI10	D1	P81/ANO1	G3	X2

Note SCL and SDA are valid only in the μPD703204Y and 70F3204Y.

Remark Leave the D4 pin open.

(2/2)

Pin No.	Name	Pin No.	Name	Pin No.	Name
G11	EV$_{SS}$	K13	PDL3/AD3	M7	PCS4
G12	PDL10/AD10	L1	P93/A3/INTP6	M8	PCM0/$\overline{\text{WAIT}}$
G13	EV$_{DD}$	L2	P94/A4/TO2	M9	PCM2/$\overline{\text{HLDAK}}$
H1	V$_{SS}$	L3	P911/A11/SO2	M10	PCT3
H2	V$_{DD}$	L4	P914/A14/SO3	M11	PCT4/$\overline{\text{RD}}$
H3	XT2	L5	P915/A15/$\overline{\text{SCK3}}$	M12	PCT7
H11	PDL8/AD8	L6	EV$_{DD}$	M13	PDL0/AD0
H12	IC/FLMD0[Notes 1, 2]	L7	PCS0/$\overline{\text{CS0}}$	N1	P96/A6/TO4
H13	PDL9/AD9	L8	PCS2/$\overline{\text{CS2}}$	N2	P98/A8/RXD1
J1	P20/SI4	L9	PCM4	N3	P910/A10/SI2
J2	P91/A1	L10	PCT2	N4	P912/A12/$\overline{\text{SCK2}}$
J3	P90/A0	L11	PCT0/$\overline{\text{WR0}}$	N5	PCS7
J11	PDL5/AD5/FLMD1[Note 1]	L12	PDL1/AD1	N6	PCS6
J12	PDL7/AD7	L13	PDL2/AD2	N7	PCS1/$\overline{\text{CS1}}$
J13	PDL6/AD6	M1	P95/A5/TO3	N8	PCS3/$\overline{\text{CS3}}$
K1	P22/$\overline{\text{SCK4}}$	M2	P97/A7/TO5	N9	PCM5
K2	P92/A2/INTP5	M3	P99/A9/TXD1	N10	PCM3/$\overline{\text{HLDRQ}}$
K3	P21/SO4	M4	P913/A13/SI3	N11	PCT1/$\overline{\text{WR1}}$
K11	PCM1/CLKOUT	M5	EV$_{SS}$	N12	PCT5
K12	PDL4/AD4	M6	PCS5	N13	PCT6/ASTB

Notes 1. FLMD0 and FLMD1 are valid only in the μPD70F3204 and 70F3204Y.

2. IC: Directly connect this pin to V$_{SS}$ (μPD703204 and 703204Y).
FLMD0: Connect this pin to V$_{SS}$ in the normal operation mode (μPD70F3204 and 70F3204Y).

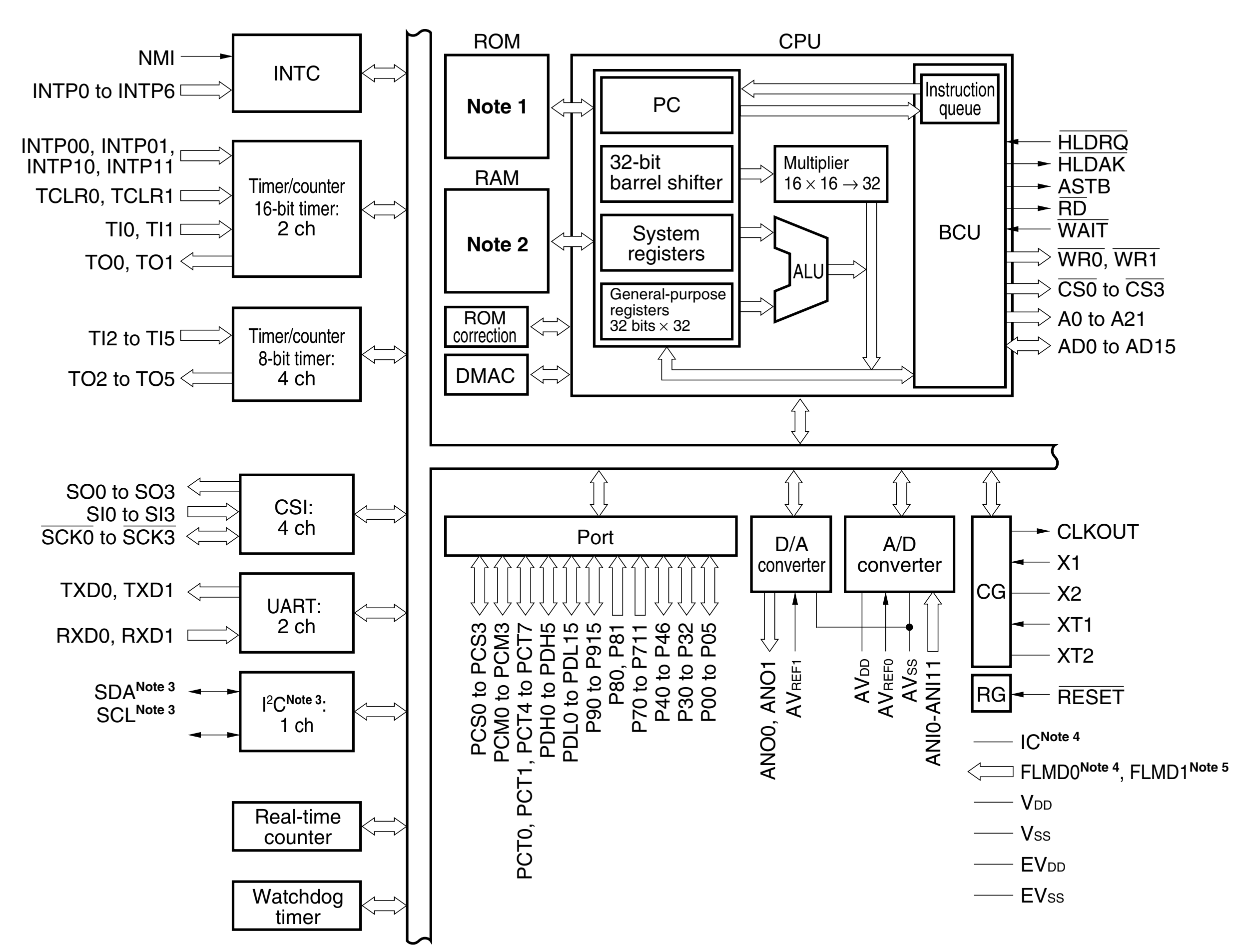
Pin Identification

A0 to A23:	Address bus
AD0 to AD15:	Address/data bus
ANI0 to ANI15:	Analog input
ANO0, ANO1:	Analog output
ASTB:	Address strobe
AV_{DD}:	Analog V_{DD}
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS}:	Analog V_{SS}
CLKOUT:	Clock output
$\overline{CS0}$ to $\overline{CS3}$:	Chip select
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
FLMD0, FLMD1:	Flash programming mode
$\overline{HLDAK}$:	Hold acknowledge
$\overline{HLDRQ}$:	Hold request
IC:	Internally connected
INTP0 to INTP6:	External interrupt input
INTP00, INTP01: INTP10, INTP11	Interrupt request to timer
NMI:	Non-maskable interrupt request
P00 to P05:	Port 0
P20 to P22:	Port 2
P30 to P32:	Port 3
P40 to P46:	Port 4
P70 to P715:	Port 7
P80, P81:	Port 8
P90 to P915:	Port 9
PCD1 to PCD3:	Port CD
PCM0 to PCM5:	Port CM
PCS0 to PCS7:	Port CS
PCT0 to PCT7:	Port CT
PDH0 to PDH7:	Port DH
PDL0 to PDL15:	Port DL
$\overline{RD}$:	Read strobe
$\overline{RESET}$:	Reset
RXD0, RXD1:	Receive data
$\overline{SCK0}$ to $\overline{SCK4}$:	Serial clock
SCL:	Serial clock
SDA:	Serial data
SI0 to SI4:	Serial input
SO0 to SO4:	Serial output
TCLR0, TCLR1:	Timer clear input
TI0 to TI5:	Timer input
TO0 to TO5:	Timer output
TXD0, TXD1:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
$\overline{WAIT}$:	Wait
$\overline{WR0}$:	Lower byte write strobe
$\overline{WR1}$:	Higher byte write strobe
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for subclock

1.6 Function Block Configuration

1.6.1 Internal block diagram

- V850ES/SA2



Notes 1. μPD703200, 703200Y: 128 KB (mask ROM)
μPD703201, 703201Y: 256 KB (mask ROM)
μPD70F3201, 70F3201Y: 256 KB (flash memory)

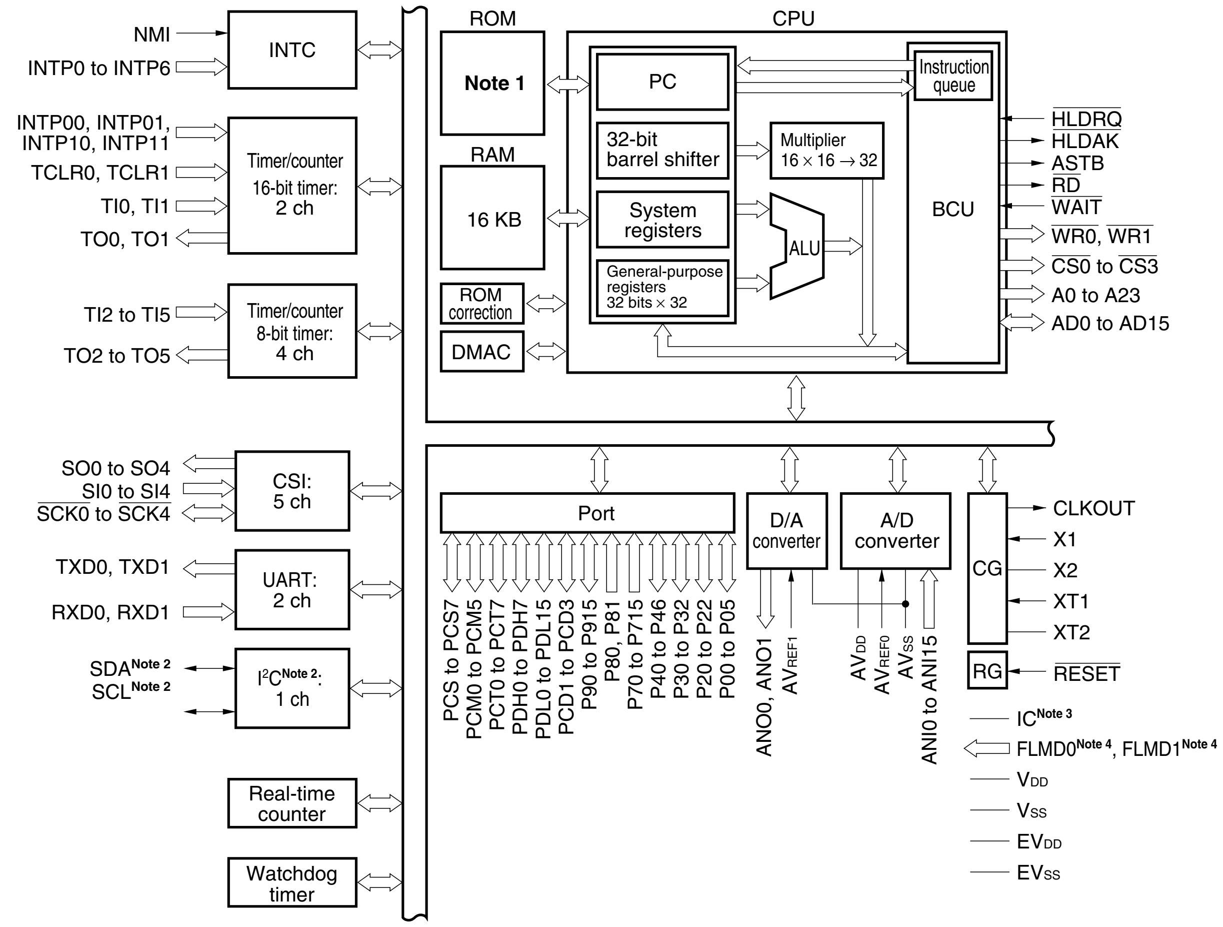
2. μPD703200, 703200Y: 8 KB
μPD703201, 703201Y, 70F3201, 70F3201Y: 16 KB

3. μPD703200Y, 703201Y, and 70F3201Y only

4. μPD703200, 703200Y, 703201, and 703201Y only

5. μPD70F3201 and 70F3201Y only

- V850ES/SA3



Notes 1. μPD703204, 703204Y: 256 KB (mask ROM)
μPD70F3204, 70F3204Y: 256 KB (flash memory)

2. μPD703204Y and 70F3204Y only

3. μPD703204 and 703204Y only

4. μPD70F3204 and 70F3204Y only

1.6.2 Internal units

(1) CPU

The CPU can execute almost all instruction processing, such as address calculation, arithmetic logic operations, and data transfer, with 1 clock, using a 5-stage pipeline.

The CPU has dedicated hardware units such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) to speed up complicated instruction processing.

(2) Bus control unit (BCU)

The BCU starts the required external bus cycles in accordance with the physical address obtained by the CPU. If the CPU does not request the start of a bus cycle when an instruction is fetched from the external memory area, the BCU generates a prefetch address and prefetches an instruction code. The prefetched instruction code is loaded to the internal instruction queue.

(3) ROM

This is a 256 KB or 128 KB mask ROM or flash memory mapped to addresses 0000000H to 003FFFFH or 0000000H to 001FFFFH. The CPU can access the ROM with 1 clock when an instruction is fetched.

(4) RAM

This is a 16 KB or 8 KB RAM mapped to addresses 3FFB000H to 3FFEFFFH or 3FFD000H to 3FFEFFFH. It can be accessed by the CPU with 1 clock when data is accessed.

(5) Interrupt controller (INTC)

The INTC processes hardware interrupt requests (NMI, INTP0 to INTP6) from the internal peripheral hardware and external sources. Eight levels of priority can be specified for these interrupt requests. Multiple interrupts can also be processed.

(6) Clock generator (CG)

Two oscillators, one for the main clock (f_X) and the other for the subclock (f_{XT}), are provided. Seven types of clocks (f_X, $f_X/2$, $f_X/4$, $f_X/8$, $f_X/16$, $f_X/32$, and f_{XT}) can be generated, of which one is supplied to the CPU as the operation clock (f_{CPU}). The subclock can be selected only as the operation clock for the real-time counter.

(7) Timer/counter

A two-channel 16-bit timer/event counter and four-channel 8-bit timer/event counter are available, enabling pulse interval and frequency measurement and programmable pulse output.

Two 8-bit timer/event counter channels can be connected in cascade and used as a 16-bit timer.

(8) Real-time counter (for watch)

This counter counts the reference time (1 second) for the watch count from the subclock (32.768 kHz) or main clock. It can also be used as an interval timer that operates with the main clock. Dedicated hardware counters for counting weeks, days, hours, minutes, and seconds, are provided, and up to 4095 weeks can be counted.

(9) Watchdog timer

A watchdog timer that detects program hang-up and system errors is provided.

This watchdog timer can also be used as an interval timer.

When used as a watchdog timer, a non-maskable interrupt request (INTWDT) is generated if the watchdog timer overflows. When used as an interval timer, a maskable interrupt request is generated when the timer overflows.

(10) Serial interface (SIO)

The V850ES/SA2 and V850ES/SA3 have asynchronous serial interfaces (UART0 and UART1), clocked serial interfaces (V850ES/SA2: CSI0 to CSI3, V850ES/SA3: CSI0 to CSI4), and an I^{2}C bus interface (I^{2}C) as the serial interfaces. The V850ES/SA2 can use up to four channels, and the V850ES/SA3 can use up to five channels at the same time. Of these channels, one can be switched between UART and CSI, and another can be switched between CSI and I^{2}C.

UART0 and UART1 transfer data using the TXD0, TXD1, RXD0, and RXD1 pins.

CSI0 to CSI3 transfer data using the SO0 to SO3, SI0 to SI3, and $\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$ pins.

CSI4 transfers data using the SO4, SI4, and $\overline{\text{SCK4}}$ pins (V850ES/SA3 only).

I^{2}C transfers data using the SDA and SCL pins.

I^{2}C is provided only in the μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y.

UART includes a dedicated baud rate generator.

(11) A/D converter

The V850ES/SA2 and V850ES/SA3 have a high-speed high-resolution, 10-bit A/D converter with 12 and 16 analog input pins, respectively. The A/D converter in both products is a successive approximation type.

(12) D/A converter

A two-channel 8-bit resolution R-string D/A converter is provided.

(13) DMA controller

A DMA controller with four channels is provided. This DMA controller transfers data between the internal RAM, on-chip peripheral I/O, and external memory, in response to interrupt requests from the on-chip peripheral I/O.

(14) ROM correction

This is a function to replace part of the program in the mask ROM with program in the internal RAM for execution. The program can be corrected at up to four places.

(15) Ports

Some port pins have a control function as well as a general-purpose port function, as shown below.

Port	I/O	Port Function	Control Function
P0	6-bit I/O	General-purpose port	NMI, external interrupt, timer input
P2[Note]	3-bit I/O		Serial interface
P3	3-bit I/O		Serial interface
P4	7-bit I/O		Serial interface, timer I/O, timer trigger
P7	12-bit input (V850ES/SA2) 16-bit input (V850ES/SA3)		A/D converter analog input
P8	2-bit input		D/A converter analog output
P9	16-bit I/O		External address bus, serial interface, timer output, external interrupt
PCD[Note]	3-bit I/O		–
PCM	4-bit I/O (V850ES/SA2) 6-bit I/O (V850ES/SA3)		External bus interface
PCS	4-bit I/O (V850ES/SA2) 8-bit I/O (V850ES/SA3)		Chip select output
PCT	6-bit I/O (V850ES/SA2) 8-bit I/O (V850ES/SA3)		External bus interface
PDH	6-bit I/O (V850ES/SA2) 8-bit I/O (V850ES/SA3)		External address bus
PDL	16-bit I/O		External address/data bus

Note V850ES/SA3 only

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

This chapter explains the names and functions of the pins in the V850ES/SA2 and V850ES/SA3, classified into port pins and non-port pins.

Two power supplies are available for the pin I/O buffers: AV_{DD} and EV_{DD}. The relationship between the power supplies and pins is shown below.

Table 2-1. I/O Buffer Power Supply for Each Pin

Power Supply	Corresponding Pin
AV_{DD}	Port 7, port 8
EV_{DD}	Port 0, port 2, port 3, port 4, port 9, port CD, port CM, port CS, port CT, port DH, port DL, $\overline{\text{RESET}}$

The differences in the pins of the V850ES/SA2 and V850ES/SA3 are shown below.

Table 2-2. Differences in Pins of V850ES/SA2 and V850ES/SA3

★

Pin	V850ES/SA2				V850ES/SA3			
	μPD703201, μPD703200	μPD70F3201	μPD703201Y, μPD703200Y	μPD70F3201Y	μPD703204	μPD70F3204	μPD703204Y	μPD70F3204Y
P20/SI4, P21/SO4, P22/$\overline{\text{SCK4}}$	None				Provided			
P712/ANI12 to P715/ANI15	None				Provided			
PCD1 to PCD3	None				Provided			
PCM4, PCM5	None				Provided			
PCS4 to PCS7	None				Provided			
PCT2, PCT3	None				Provided			
PDH6/A22, PDH7/A23	None				Provided			
SDA, SDA	None		Provided		None		Provided	
FLMD0, FLMD1	None	Provided	None	Provided	None	Provided	None	Provided
IC	Provided	None	Provided	None	Provided	None	Provided	None

(1) Port pins

(1/3)

Pin Name	I/O	On-Chip Pull-up Resistor	Function	Alternate-Function Pin
P00	I/O	Provided	Port 0. 6-bit I/O port. Can be set to input or output in 1-bit units.	NMI
P01				INTP0/TI2
P02				INTP1/TI3
P03				INTP2/TI4
P04				INTP3/TI5
P05				INTP4
[P20]	I/O	Provided	Port 2. 3-bit I/O port. Can be set to input or output in 1-bit units. Can be specified as an N-ch open drain port in 1-bit units (P21 and P22 only).	[SI4]
[P21]				[SO4]
[P22]				[$\overline{\mathrm{SCK4}}$]
P30	I/O	Provided	Port 3. 3-bit I/O port. Can be set to input or output in 1-bit units. Can be specified as an N-ch open drain port in 1-bit units (P31 and P32 only).	SI1/RXD0
P31				SO1/TXD0
P32				$\overline{\mathrm{SCK1}}$
P40	I/O	Provided	Port 4. 7-bit I/O port. Can be set to input or output in 1-bit units. Can be specified as an N-ch open drain port in 1-bit units (P41 and P42 only).	SI0
P41				SO0/SDA$^{\text{Note}}$
P42				$\overline{\mathrm{SCK0}}$/SCL$^{\text{Note}}$
P43				INTP00/TI0/TCLR0
P44				INTP01/TO0
P45				INTP10/TI1/TCLR1
P46				INTP11/TO1
P70	Input	None	Port 7. 12-bit input port (V850ES/SA2). 16-bit input port (V850ES/SA3).	ANI0
P71				ANI1
P72				ANI2
P73				ANI3
P74				ANI4
P75				ANI5
P76				ANI6
P77				ANI7
P78				ANI8
P79				ANI9
P710				ANI10
P711				ANI11
[P712]				[ANI12]
[P713]				[ANI13]
[P714]				[ANI14]
[P715]				[ANI15]

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

(2/3)

Pin Name	I/O	On-Chip Pull-up Resistor	Function	Alternate-Function Pin
P80	Input	None	Port 8. 2-bit input port	ANO0
P81				ANO1
P90	I/O	Provided	Port 9. 16-bit I/O port. Can be set to input or output in 1-bit units. Can be specified as an N-ch open drain port in 1-bit units (P911, P912, P914, and P915 only).	A0
P91				A1
P92				A2/INTP5
P93				A3/INTP6
P94				A4/TO2
P95				A5/TO3
P96				A6/TO4
P97				A7/TO5
P98				A8/RXD1
P99				A9/TXD1
P910				A10/SI2
P911				A11/SO2
P912				A12/$\overline{\text{SCK2}}$
P913				A13/SI3
P914				A14/SO3
P915				A15/$\overline{\text{SCK3}}$
[PCD1]	I/O	None	Port CD. 3-bit I/O port. Can be set to input or output in 1-bit units.	–
[PCD2]				–
[PCD3]				–
PCM0	I/O	None	Port CM. 4-bit I/O port (V850ES/SA2). 6-bit I/O port (V850ES/SA3). Can be set to input or output in 1-bit units.	$\overline{\text{WAIT}}$
PCM1				CLKOUT
PCM2				$\overline{\text{HLDAK}}$
PCM3				$\overline{\text{HLDRQ}}$
[PCM4]				–
[PCM5]				–
PCS0	I/O	None	Port 10. 4-bit I/O port (V850ES/SA2). 8-bit I/O port (V850ES/SA3). Can be set to input or output in 1-bit units.	$\overline{\text{CS0}}$
PCS1				$\overline{\text{CS1}}$
PCS2				$\overline{\text{CS2}}$
PCS3				$\overline{\text{CS3}}$
[PCS4]				–
[PCS5]				–
[PCS6]				–
[PCS7]				–
PCT0	I/O	None	Port CT. 6-bit I/O port (V850ES/SA2). 8-bit I/O port (V850ES/SA3). Can be set to input or output in 1-bit units.	$\overline{\text{WR0}}$
PCT1				$\overline{\text{WR1}}$
[PCT2]				–
[PCT3]				–
PCT4				$\overline{\text{RD}}$
PCT5				–
PCT6				ASTB
PCT7				–

Remark [ ]: Pins provided only in the V850ES/SA3

(3/3)

Pin Name	I/O	On-Chip Pull-up Resistor	Function	Alternate-Function Pin
PDH0	I/O	None	Port DH. 6-bit I/O port (V850ES/SA2). 8-bit I/O port (V850ES/SA3). Can be set to input or output in 1-bit units.	A16
PDH1				A17
PDH2				A18
PDH3				A19
PDH4				A20
PDH5				A21
[PDH6]				[A22]
[PDH7]				[A23]
PDL0	I/O	None	Port DL. 16-bit I/O port. Can be set to input or output in 1-bit units.	AD0
PDL1				AD1
PDL2				AD2
PDL3				AD3
PDL4				AD4
PDL5				AD5/FLMD1^Note
PDL6				AD6
PDL7				AD7
PDL8				AD8
PDL9				AD9
PDL10				AD10
PDL11				AD11
PDL12				AD12
PDL13				AD13
PDL14				AD14
PDL15				AD15

Note μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

(2) Non-port pins

(1/3)

Pin Name	I/O	On-Chip Pull-up Resistor	Function	Alternate-Function Pin
A0	Output	Provided	Address bus for external memory (when separate bus is used)	P90
A1				P91
A2				P92/INTP5
A3				P93/INTP6
A4				P94/TO2
A5				P95/TO3
A6				P96/TO4
A7				P97/TO5
A8				P98/RXD1
A9				P99/TXD1
A10				P910/SI2
A11				P911/SO2
A12				P912/$\overline{\text{SCK2}}$
A13				P913/SI3
A14				P914/SO3
A15				P915/$\overline{\text{SCK3}}$
A16 to A21, [A22, A23]	Output	None	Address bus for external memory	PDH0 to PDH5, [PDH6, PDH7]
AD0 to AD4	I/O	None	Address/data bus for external memory	PDL0 to PDL4
AD5				PDL5/FLMD1$^{\text{Note}}$
AD6 to AD15				PDL6 to PDL15
ANI0	Input	None	Analog voltage input for A/D converter	P70
ANI1				P71
ANI2				P72
ANI3				P73
ANI4				P74
ANI5				P75
ANI6				P76
ANI7				P77
ANI8				P78
ANI9				P79
ANI10				P710
ANI11				P711
[ANI12]				[P712]
[ANI13]				[P713]
[ANI14]				[P714]
[ANI15]				[P715]
ANO0	Output	None	Analog voltage output for D/A converter	P80
ANO1				P81
ASTB	Output	None	Address strobe signal output for external memory	PCT6

Note μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

(2/3)

Pin Name	I/O	On-Chip Pull-up Resistor	Function	Alternate-Function Pin
AV_{DD}	–	–	Positive power supply for A/D converter (same potential as V_{DD})	–
AV_{REF0}	Input	–	Reference voltage input for A/D converter	–
AV_{REF1}			Reference voltage input for D/A converter	–
AV_{SS}	–	–	Ground potential for A/D and D/A converters (same potential as V_{SS})	–
CLKOUT	Output	None	Internal system clock output	PCM1
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	Output	None	Chip select output	PCS0 to PCS3
EV_{DD}	–	–	Positive power supply for external device (same potential as V_{DD})	–
EV_{SS}	–	–	Ground potential for external device (same potential as V_{SS})	–
FLMD0[Note 1]	Input	None	Flash programming mode setting pin	–
FLMD1[Note 1]				PDL5/AD5
$\overline{\text{HLDAK}}$	Output	None	Bus hold acknowledge output	PCM2
$\overline{\text{HLDRQ}}$	Input	None	Bus hold request input	PCM3
IC	–	–	Internally connected (connect this pin directly to V_{SS}) (μPD703200, 703200Y, 703201, 703201Y, 703204, and 703204Y only)	–
INTP0 to INTP3	Input	Provided	External interrupt request input (maskable, analog noise elimination)	P01/TI2 to P04/TI5
INTP4				P05
INTP5				P92/A2
INTP6				P93/A3
INTP00	Input	Provided	Capture trigger input (TM0)	P43/TI0/TCLR0
INTP01				P44/TO0
INTP10			Capture trigger input (TM1)	P45/TI1/TCLR1
INTP11				P46/TO1
NMI	Input	Provided	External interrupt input (non-maskable, analog noise elimination)	P00
$\overline{\text{RD}}$	Output	None	Read strobe signal output for external memory	PCT4
$\overline{\text{RESET}}$	Input	–	System reset input	–
RXD0	Input	Provided	Serial receive data input (UART0)	P30/SI1
RXD1			Serial receive data input (UART1)	P98/A8
$\overline{\text{SCK0}}$	I/O	Provided	Serial clock I/O (CSI0)	P42/SCL[Note 2]
$\overline{\text{SCK1}}$			Serial clock I/O (CSI1)	P32
$\overline{\text{SCK2}}$			Serial clock I/O (CSI2)	P912/A12
$\overline{\text{SCK3}}$			Serial clock I/O (CSI3)	P915/A15
[$\overline{\text{SCK4}}$]			Serial clock I/O (CSI4)	[P22]
SCL[Note 2]	I/O	Provided	Serial clock I/O (I^2C)	P42/$\overline{\text{SCK0}}$
SDA[Note 2]	I/O	Provided	Serial transmit/receive data I/O (I^2C)	P41/SO0
SI0	Input	Provided	Serial receive data input (CSI0)	P40
SI1			Serial receive data input (CSI1)	P30/RXD0
SI2			Serial receive data input (CSI2)	P910/A10
SI3			Serial receive data input (CSI3)	P913/A13
[SI4]			Serial receive data input (CSI4)	[P20]

Notes 1. μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y only

2. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

(3/3)

Pin Name	I/O	On-Chip Pull-up Resistor	Function	Alternate-Function Pin
SO0	Output	Provided	Serial transmit data output (CSI0)	P41/SDA[Note]
SO1			Serial transmit data output (CSI1)	P31/TXD0
SO2			Serial transmit data output (CSI2)	P911/A11
SO3			Serial transmit data output (CSI3)	P914/A14
[SO4]			Serial transmit data output (CSI4)	[P21]
TCLR0	Input	Provided	Timer clear input (TM0)	P43/INTP00/TI0
TCLR1			Timer clear input (TM1)	P45/INTP10/TI1
TI0	Input	Provided	External event/clock input (TM0)	P43/INTP00/TCLR0
TI1			External event/clock input (TM1)	P45/INTP10/TCLR1
TI2			External event/clock input (TM2)	P01/INTP0
TI3			External event/clock input (TM3)	P02/INTP1
TI4			External event/clock input (TM4)	P03/INTP2
TI5			External event/clock input (TM5)	P04/INTP3
TO0	Output	Provided	Timer output (TM0)	P44/INTP01
TO1			Timer output (TM1)	P46/INTP11
TO2			Timer output (TM2)	P94/A4
TO3			Timer output (TM3)	P95/A5
TO4			Timer output (TM4)	P96/A6
TO5			Timer output (TM5)	P97/A7
TXD0	Output	Provided	Serial transmit data output (UART0)	P31/SO1
TXD1			Serial transmit data output (UART1)	P99/A9
V$_{DD}$	–	–	Positive power supply pin for internal circuits	–
V$_{SS}$	–	–	Ground potential for internal circuits	–
$\overline{\text{WAIT}}$	Input	None	External wait input	PCM0
$\overline{\text{WR0}}$	Output	None	Write strobe for external memory (lower 8 bits)	PCT0
$\overline{\text{WR1}}$			Write strobe for external memory (higher 8 bits)	PCT1
X1	Input	None	Oscillator connection for main clock	–
X2	–			–
XT1	Input	None	Oscillator connection for subclock	–
XT2	–			–

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

2.2 Pin Status

The operating status of each pin in each operation mode is shown below.

Table 2-3. Operating Status of Each Pin in Each Operation Mode

Bus Control Pins	Reset	HALT Mode or DMA Transfer	IDLE and STOP Modes	Idle State[Note 2]	Bus Hold
AD0 to AD15	Hi-Z[Note 1]	**Note 3**	Hi-Z	Retained	Hi-Z
A16 to A23		Undefined[Note 4]			
A0 to A15		Undefined			
$\overline{\text{WAIT}}$		–	–	–	–
CLKOUT		Operates	L	Operates	Operates
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$		H	H	Retained	Hi-Z
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$				H	
$\overline{\text{RD}}$					
ASTB					
$\overline{\text{HLDAK}}$		Operates			L
$\overline{\text{HLDRQ}}$			–	–	Operates

Notes 1. Because the bus control pins function alternately as port pins, they are initialized to the input mode (port mode).

2. Indicates the pin status in the idle state that is inserted after the T3 state.

3. In separate mode: Hi-Z
In multiplexed mode: Undefined

4. In separate mode only

Remark

Hi-Z:	High impedance
Retained:	Status in external bus cycle immediately before is retained.
L:	Low-level output
H:	High-level output
–:	Input not sampled (not acknowledged)

2.3 Description of Pin Functions

(1) P00 to P05 (Port 0) … 3-state I/O

Port 0 is a 6-bit I/O port that can be set to the input or output in 1-bit units.

Besides functioning as I/O port pins, P00 to P05 also operate as NMI input, external interrupt request, and timer/counter input pins. Port or control mode can be selected as the operation mode for each bit, with the valid edge of each pin specified by using the INTR0 and INTF0 registers.

(a) Port mode

P00 to P05 can be set to the input or output mode in 1-bit units by using port mode register 0 (PM0).

(b) Control mode

(i) NMI (non-maskable interrupt request) … Input

This pin inputs a non-maskable interrupt request.

(ii) INTP0 to INTP4 (interrupt request from peripherals) … Input

These pins input an external interrupt request.

(iii) TI2 to TI5 (timer input 2 to 5) … Input

These pins input an external count clock to timers 2 to 5.

(2) P20 to P22 (Port 2) (V850ES/SA3 only) … 3-state I/O

Port 2 is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.

Besides functioning as I/O port pins, P20 to P22 also operate as the I/O pins of the serial interface.

These pins can be set to the port or control mode in 1-bit units.

The output mode of P21 and P22 can also be set to normal output or N-ch open-drain output.

(a) Port mode

P20 to P22 can be set to the input or output mode in 1-bit units by using port mode register 2 (PM2).

(b) Control mode

(i) SI4 (serial input 4) … Input

This pin inputs the serial receive data of CSI4.

(ii) SO4 (serial output 4) … Output

This pin outputs the serial transmit data of CSI4.

(iii) $\overline{\text{SCK4}}$ (serial clock 4) … 3-state I/O

This is the serial clock I/O pin of CSI4.

(3) P30 to P32 (Port 3) … 3-state I/O

Port 3 is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, P30 to P32 also operate as the I/O pins of the serial interface.
These pins can be set to the port or control mode in 1-bit units.
The output mode of P31 and P32 can also be set to normal output or N-ch open-drain output.

(a) Port mode

P30 to P32 can be set to the input or output mode in 1-bit units by using port mode register 3 (PM3).

(b) Control mode

(i) SI1 (serial input 1) … Input

This pin inputs the serial receive data of CSI1.

(ii) SO1 (serial output 1) … Output

This pin outputs the serial transmit data of CSI1.

(iii) $\overline{\text{SCK1}}$ (serial clock 1) … 3-state I/O

This is the serial clock I/O pin of CSI1.

(iv) RXD0 (receive data 0) … Input

This pin inputs the serial receive data of UART0.

(v) TXD0 (transmit data 0) … Output

This pin outputs the serial transmit data of UART0.

(4) P40 to P46 (Port 4) … 3-state I/O

Port 4 is a 7-bit I/O port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, P40 to P46 also operate as the I/O pins of the timer/counters and serial interface, and as the external interrupt request input pin. These pins can be set to the port or control mode in 1-bit units.
The output mode of P41 and P42 can also be set to normal output or N-ch open-drain output.

(a) Port mode

P40 to P46 can be set to the input or output mode in 1-bit units by using port mode register 4 (PM4).

(b) Control mode

(i) TI0, TI1 (timer input 0, 1) … Input

These pins input an external count clock to timers 0 and 1.

(ii) TO0, TO1 (timer output 0, 1) … Output

These pins output a pulse signal from timers 0 and 1.

(iii) TCLR0, TCLR1 (timer clear input 0, 1) … Input

These pins input an external clear signal to timers 0 and 1.

(iv) SI0 (serial input 0) … Input

This pin inputs the serial receive data of CSI0.

(v) SO0 (serial output 0) … Output

This pin outputs the serial transmit data of CSI0.

(vi) $\overline{\text{SCK0}}$ (serial clock 0) … 3-state I/O

This is the serial clock I/O pin of CSI0.

(vii) SDA (serial data) … I/O

This pin inputs/outputs the serial transmit/receive data of I^2C (μPD703200Y, 703201Y, 70F3201Y, 703204Y, and 70F3204Y only).

(viii) SCL (serial clock) … I/O

This pin inputs/outputs the serial clock to/from I^2C (μPD703200Y, 703201Y, 70F3201Y, 703204Y, and 70F3204Y only).

(ix) INTP00, INTP01, INTP10, INTP11 (interrupt request to timer) … Input

These pins input an external interrupt request to timers 0 and 1.

(5) P70 to P711 (Port 7) (V850ES/SA2) … Input
P70 to P715 (Port 7) (V850ES/SA3) … Input

[V850ES/SA2]

Port 7 is a 12-bit input port with all its bits fixed to the input mode.
Besides functioning as input port pins, P70 to P711 also operate as the analog input pins of the A/D converter in the control mode. However, the mode of these pins cannot be changed between the input port mode and analog input mode.

(a) Port mode

P70 to P711 function as input port pins.

(b) Control mode

P70 to P711 function as the ANI0 to ANI11 pins. The mode of these pins cannot be changed between the input port mode and analog input mode.

(i) ANI0 to ANI11 (analog input 0 to 11) … Input

These are the analog input pins of the A/D converter.
To prevent these pins malfunctioning due to noise, connect a capacitor between these pins and AV_{SS}.
Make sure that a voltage outside the range of AV_{SS} to AV_{REF0} is not applied to any pin that is being used as an input pin of the A/D converter. If there is a possibility that a noise greater than AV_{REF0} or lower than AV_{SS} will be superimposed on any of these pins, clamp the pins using a diode with a low V_F.

[V850ES/SA3]

Port 7 is a 16-bit input port with all its bits fixed to the input mode.
Besides functioning as input port pins, P70 to P715 also operate as the analog input pins of the A/D converter in the control mode. However, the mode of these pins cannot be changed between the input port mode and analog input mode.

(a) Port mode

P70 to P715 function as input port pins.

(b) Control mode

P70 to P715 function as the ANI0 to ANI15 pins. The mode of these pins cannot be changed between the input port mode and analog input mode.

(i) ANI0 to ANI15 (analog input 0 to 15) … Input

These are the analog input pins of the A/D converter.
To prevent these pins malfunctioning due to noise, connect a capacitor between these pins and AV_{SS}.
Make sure that a voltage outside the range of AV_{SS} to AV_{REF0} is not applied to any pin that is being used as an input pin of the A/D converter. If there is a possibility that a noise greater than AV_{REF0} or lower than AV_{SS} will be superimposed on any of these pins, clamp the pins using a diode with a low V_F.

(6) P80, P81 (Port 8) … Input/Output

Port 8 is a 2-bit input port.
Besides functioning as input port pins, P80 and P81 also operate as the analog output pins of the D/A converter in the control mode. The mode of these pins, however, cannot be changed between the control mode and input port mode.

(a) Port mode

P80 and P81 function as input port pins.

(b) Control mode

(i) ANO0, ANO1 (analog output 0, 1) … Output

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

(7) P90 to P915 (Port 9) … 3-state I/O

Port 9 is a 16-bit I/O port that can be set to the input or output mode in 1-bit units.

Besides functioning as I/O port pins, P90 to P915 also operate as the I/O pins of the serial interface and timer/counters, the address bus pins to extend the memory externally, and external interrupt request input pins.

These pins can be set to the port or control mode in 1-bit units.

The output mode of P911, P912, P914, and P915 can also be set to normal output or N-ch open-drain output.

(a) Port mode

P90 to P915 can be set to the input or output mode in 1-bit units by using port mode register 9 (PM9).

(b) Control mode

(i) SI2, SI3 (serial input 2, 3) … Input

These pins input the serial receive data of CSI2 and CSI3.

(ii) SO2, SO3 (serial output 2, 3) … Output

These pins output the serial transmit data of CSI2 and CSI3.

(iii) $\overline{\text{SCK2}}$, $\overline{\text{SCK3}}$ (serial clock 2, 3) … 3-state I/O

These pins are the serial clock I/O pins of CSI2 and CSI3.

(iv) RXD1 (receive data 1) … Input

This pin inputs the serial receive data of UART1.

(v) TXD1 (transmit data 1) … Output

This pin outputs the serial transmit data of UART1.

(vi) TO2 to TO5 (timer output 2 to 5) … Output

These pins output a pulse signal from timers 2 to 5.

(vii) A0 to A15 (address bus 0 to 15) … Output

These pins form a 16-bit address output bus to access an external memory. The output signal changes at the rising edge of the T1 state in the bus cycle. When the bus cycle is not active, they retain the address of the bus cycle immediately before.

(viii) INTP5, INTP6 (interrupt request from peripherals) … Input

These pins input an external interrupt request.

(8) PCD1 to PCD3 (Port CD) (V850ES/SA3 only) … 3-state I/O

Port CD is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.

PCD1 to PCD3 can be set to the input or output mode in 1-bit units by using port mode register CD (PMCD).

(9) PCM0 to PCM3 (Port CM) (V850ES/SA2) … 3-state I/O
PCM0 to PCM5 (Port CM) (V850ES/SA3) … 3 state I/O

[V850ES/SA2]

Port CM is a 4-bit I/O port that can be set to the input or output mode in 1-bit units.

Besides functioning as I/O port pins, in the control mode PCM0 to PCM3 also operate as the bus hold control signal output and bus clock output pins, and as the control signal that inserts a wait state ($\overline{\text{WAIT}}$) in the bus cycle.

(a) Port mode

PCM0 to PCM3 can be set to the input or output mode in 1-bit units by using port mode register CM (PMCM).

(b) Control mode

(i) $\overline{\text{HLDAK}}$ (hold acknowledge) … Output

This pin outputs an acknowledge signal indicating that the V850ES/SA2 has made the address bus, data bus, and control bus go into a high-impedance state, in response to a bus hold request.

While this signal is active, the address bus, data bus, and control bus are in the high-impedance state.

(ii) $\overline{\text{HLDRQ}}$ (hold request) … Input

This pin is used by an external device to request the V850ES/SA2 to release the address bus, data bus, and control bus. A signal can be input to this pin asynchronously to CLKOUT. When this pin is active, the V850ES/SA2 makes the address bus, data bus, and control bus go into a high-impedance state immediately or after completion of the bus cycle under execution, if any, and then asserts the $\overline{\text{HLDAK}}$ signal and releases the bus.

(iii) CLKOUT (clock output) … Output

This pin outputs the internally generated bus clock.

(iv) $\overline{\text{WAIT}}$ (wait) … Input

This pin inputs a control signal that inserts a wait state in the bus cycle. The signal is sampled at the fall of the CLKOUT signal of the T2 and TW states of the bus cycle in the multiplexed mode, and at the rise of the CLKOUT signal immediately after the T1 and TW states of the bus cycle in the separate mode.

The wait function is turned on/off by port mode control register CM (PMCCM).

[V850ES/SA3]

Port CM is a 6-bit I/O port that can be set to the input or output mode in 1-bit units.

Besides functioning as I/O port pins, in the control mode PCM0 to PCM5 also operate as the bus hold control signal output and bus clock output pins, and as the control signal that inserts a wait state ($\overline{\text{WAIT}}$) in the bus cycle.

(a) Port mode

PCM0 to PCM5 can be set to the input or output mode in 1-bit units by using port mode register CM (PMCM).

(b) Control mode

(i) $\overline{\text{HLDAK}}$ (hold acknowledge) … Output

This pin outputs an acknowledge signal indicating that the V850ES/SA3 has made the address bus, data bus, and control bus go into a high-impedance state, in response to a bus hold request.

While this signal is active, the address bus, data bus, and control bus are in the high-impedance state.

(ii) $\overline{\text{HLDRQ}}$ (hold request) … Input

This pin is used by an external device to request the V850ES/SA3 to release the address bus, data bus, and control bus. A signal can be input to this pin asynchronously to CLKOUT. When this pin is active, the V850ES/SA3 makes the address bus, data bus, and control bus go into a high-impedance state immediately or after completion of the bus cycle under execution, if any, and then asserts the $\overline{\text{HLDAK}}$ signal and releases the bus.

(iii) CLKOUT (clock output) … Output

This pin outputs the internally generated bus clock.

(iv) $\overline{\text{WAIT}}$ (wait) … Input

This pin inputs a control signal that inserts a wait state in the bus cycle. The signal is sampled at the fall of the CLKOUT signal of the T2 and TW states of the bus cycle in the multiplexed mode, and at the rise of the CLKOUT signal immediately after the T1 and TW states of the bus cycle in the separate mode.

The wait function is turned on/off by port mode control register CM (PMCCM).

(10) PCS0 to PSC3 (Port CS) (V850ES/SA2) … 3-state I/O
PCS0 to PCS5 (Port CS) (V850ES/SA3) … 3-state I/O

[V850ES/SA2]

Port CS is a 4-bit I/O port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, in the control mode PCS0 to PSC3 also operate as the control signal output pins when the memory and peripheral I/O are expanded externally.

(a) Port mode

PCS0 to PSC3 can be set to the input or output mode in 1-bit units by using port mode register CM (PMCS).

(b) Control mode

(i) $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ (chip select) … Output

These are the chip select signals for the SRAM, external ROM, and external peripheral I/O area.
The $\overline{\text{CSn}}$ signal is assigned to memory block n (n = 0 to 3).
Each of these signals is active while the bus cycle accessing the corresponding memory block is being executed, and inactive in the idle state (TI).

[V850ES/SA3]

Port CS is an 8-bit I/O port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, in the control mode PCS0 to PSC3 also operate as the control signal output pins when the memory and peripheral I/O are expanded externally.

(a) Port mode

PCS0 to PCS7 can be set to the input or output mode in 1-bit units by using port mode register CS (PMCS).

(b) Control mode

(i) $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ (chip select) … Output

These are the chip select signals for the SRAM, external ROM, and external peripheral I/O area.
The $\overline{\text{CSn}}$ signal is assigned to memory block n (n = 0 to 3).
Each of these signals is active while the bus cycle accessing the corresponding memory block is being executed, and inactive in the idle state (TI).

(11) PCT0, PCT1, PCT4 to PCT7 (Port CT) (V850ES/SA2) … 3-state I/O
PCT0 to PCT7 (Port CT) (V850ES/SA3) … 3-state I/O

[V850ES/SA2]

Port CT is a 6-bit port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, in the control mode PCT0, PCT1, and PCT4 to PCT7 also operate as the control signal output pins when the memory is expanded externally.

(a) Port mode

PCT0, PCT1, and PCT4 to PCT7 can be set to the input or output mode in 1-bit units by using port mode register CT (PMCT).

(b) Control mode

(i) $\overline{\text{WR0}}$ (lower byte write strobe) … Output

This is the write strobe signal output pin for the lower data of the external 16-bit data bus.

(ii) $\overline{\text{WR1}}$ (higher byte write strobe) … Output

This is the write strobe signal output pin for the higher data of the external 16-bit data bus.

(iii) $\overline{\text{RD}}$ (read strobe) … Output

This is the read strobe signal output pin for the external 16-bit data bus.

(iv) ASTB (address strobe) … Output

This is the latch strobe signal output pin of the external address bus. The output signal goes low at the falling edge of the T1 state in the bus cycle, and goes high at the falling edge of the T3 state. It is high when the bus cycle is not active.

[V850ES/SA3]

Port CT is an 8-bit port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, in the control mode PCT0 to PCT7 also operate as the control signal output pins when the memory is expanded externally.

(a) Port mode

PCT0 to PCT7 can be set to the input or output mode in 1-bit units by using port mode register CT (PMCT).

(b) Control mode

(i) $\overline{\text{WR0}}$ (lower byte write strobe) … Output

This is the write strobe signal output pin for the lower data of the external 16-bit data bus.

(ii) $\overline{\text{WR1}}$ (higher byte write strobe) … Output

This is the write strobe signal output pin for the higher data of the external 16-bit data bus.

(iii) $\overline{\text{RD}}$ (read strobe) … Output

This is the read strobe signal output pin for the external 16-bit data bus.

(iv) ASTB (address strobe) … Output

This is the latch strobe signal output pin of the external address bus. The output signal goes low at the falling edge of the T1 state in the bus cycle, and goes high at the falling edge of the T3 state. It is high when the bus cycle is not active.

(12) PDH0 to PDH5 (Port DH) (V850ES/SA2) … 3-state I/O
PDH0 to PDH7 (Port DH) (V850ES/SA3) … 3-state I/O

[V850ES/SA2]

Port DH is a 6-bit port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, in the control mode PDH0 to PDH5 also operate as the address bus pins when the memory is expanded externally.

(a) Port mode

PDH0 to PDH5 can be set to the input or output mode in 1-bit units by using port mode register DH (PMDH).

(b) Control mode

(i) A16 to A21 (address bus 16 to 21) … Output

These pins form a 6-bit address output bus to access an external device. The output signal changes at the rising edge of the T1 state in the bus cycle. The address of the immediately preceding bus cycle is retained when the bus cycle is inactive.

[V850ES/SA3]

Port DH is an 8-bit port that can be set to the input or output mode in 1-bit units.
Besides functioning as I/O port pins, in the control mode PDH0 to PDH7 also operate as the address bus pins when the memory is expanded externally.

(a) Port mode

PDH0 to PDH7 can be set to the input or output mode in 1-bit units by using port mode register DH (PMDH).

(b) Control mode

(i) A16 to A23 (address bus 16 to 23) … Output

These pins form an 8-bit address output bus to access an external device. The output signal changes at the rising edge of the T1 state in the bus cycle. The address of the immediately preceding bus cycle is retained when the bus cycle is inactive.

(13) PDL0 to PDL15 (Port DL) … 3-state I/O

Port DL is a 16-bit I/O port that can be set to the input or output mode in 1-bit units.

Besides functioning as I/O port pins, PDL0 to PDL15 also operate a time-division address/data bus (AD0 to AD15) when the memory is externally expanded. Each bit of the port can be individually set to the port or control mode.

In addition, the PDL5 pin of the μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y functions as the FLMD1 pin when the flash memory is programmed (when a high level is input to FLMD0). At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

PDL0 to PDL15 can be set to the input or output mode in 1-bit units by using port mode register DL (PMDL).

(b) Control mode

(i) AD0 to AD15 (address/data bus 0 to 15) … 3-state I/O

This is a multiplexed address/data bus that is used to access an external device. In the multiplexed bus mode, it outputs an address or inputs/outputs data. In the separate bus mode, the bus inputs or outputs data.

(14) $\overline{\text{RESET}}$ (Reset) … Input

The $\overline{\text{RESET}}$ signal is input asynchronously. If a signal having a specific low-level width is input to this pin, regardless of the operation clock, the system is reset as a priority over all other operations.

This pin is used to release the standby mode (HALT, IDLE, or STOP) as well as for normal initialization/starting.

(15) X1, X2 (Crystal for main clock)

Connect an oscillator for system clock generation to these pins.

(16) XT1, XT2 (Crystal for subclock)

Connect an oscillator for subclock generation to these pins.

(17) AV_{DD} (Analog V_{DD})

This pin supplies positive power to the A/D converter and alternate-function port pins.

(18) AV_{SS} (Analog V_{SS})

This is a ground pin for the A/D converter and alternate-function port pins.

(19) AV_{REF0} (Analog reference voltage) … Input

This pin supplies a reference voltage to the A/D converter.

(20) AV_{REF1} (Analog reference voltage) … Input

This pin supplies a reference voltage to the D/A converter.

(21) EV$_{DD}$ (Power supply for port)

This pin supplies positive power for the I/O ports and pins with alternate functions.

(22) EV$_{SS}$ (Ground for port)

This is a ground pin for the I/O ports and pins with alternate functions.

(23) V$_{DD}$ (Power supply)

This pin supplies positive power. Connect all the V$_{DD}$ pins to the positive power supply.

(24) V$_{SS}$ (Ground)

This is the ground pin. Connect all the V$_{SS}$ pins to ground.

(25) FLMD0, FLMD1 (Flash programming mode)

These pins supply positive power for flash memory programming mode.

These pins are provided only in the μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y. In the normal operation mode, connect these pins to V$_{SS}$.

(26) IC (Internally connected)

These pins are internally connected and provided only in the μPD703200, 703200Y, 70F3201, 70F3201Y, 70F3204, and 70F3204Y. In the normal operation mode, connect these pins to V$_{SS}$.

2.4 Types of Pin I/O Circuits, I/O Buffer Power Supplies, and Connection of Unused Pins

(1/2)

Pin	Alternate Function	I/O Circuit Type	Recommended Connection
P00	NMI	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01 to P04	INTP0/TI2 to INTP3/TI5		
P05	INTP4		
[P20]	[SI4]	5-W	
[P21]	[SO4]	10-E	
[P22]	[$\overline{\text{SCK4}}$]	10-F	
P30	SI1/RXD0	5-W	
P31	SO1/TXD0	10-E	
P32	$\overline{\text{SCK1}}$	10-F	
P40	SI0	5-W	
P41	SO0/SDA[Note]	10-F	
P42	$\overline{\text{SCK0}}$/SCL[Note]	10-F	
P43	INTP00/TI0/TCLR0	5-W	
P44	INTP01/TO0		
P45	INTP10/TI1/TCLR1		
P46	INTP11/TO1		
P70 to P711, [P712 to P715]	ANI0 to ANI11 [ANI12 to ANI15]	9	Independently connect to AV_{DD} or AV_{SS} via a resistor.
P80, P81	ANO0, ANO1	34	
P90, P91	A0, A1	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P92, P93	A2/INTP5, A3/INTP6	5-W	
P94 to P97	A4/TO2 to A7/TO5	5-A	
P98	A8/RXD1	5-W	
P99	A9/TXD1	5-A	
P910	A10/SI2	5-W	
P911	A11/SO2	10-E	
P912	A12/$\overline{\text{SCK2}}$	10-F	
P913	A13/SI3	5-W	
P914	A14/SO3	10-E	
P915	A15/$\overline{\text{SCK3}}$	10-F	
[PCD1 to PCD3]	–	5	
PCM0	$\overline{\text{WAIT}}$		
PCM1	CLKOUT		
PCM2	$\overline{\text{HLDAK}}$		
PCM3	$\overline{\text{HLDRQ}}$		
[PCM4]	–		
[PCM5]	–		

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

(2/2)

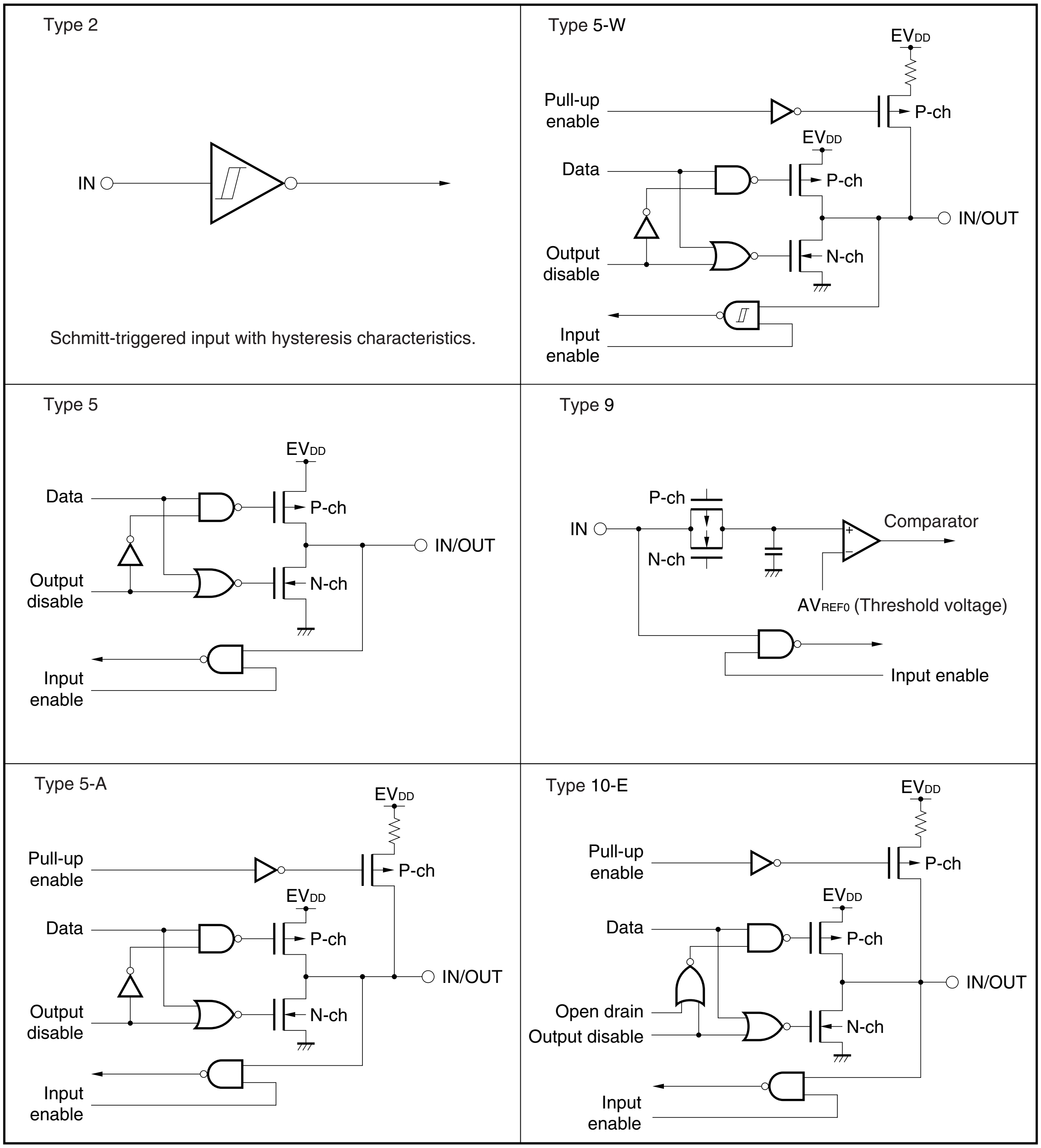
Pin	Alternate Function	I/O Circuit Type	Recommended Connection
PCS0 to PCS3	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	5	Input: Independently connect to EV$_{DD}$ or EV$_{SS}$ via a resistor. Output: Leave open.
[PCS4 to PCS7]	–		
PCT0, PCT1	$\overline{\text{WR0}}$, $\overline{\text{WR1}}$		
[PCT2, PCT3]	–		
PCT4	$\overline{\text{RD}}$		
PCT5	–		
PCT6	ASTB		
PCT7	–		
PDH0 to PDH5, [PDH6, PDH7]	A16 to A21, [A22, A23]		
PDL0 to PDL4	AD0 to AD4		
PDL5	AD5/FLMD1 Note 1		
PDL6 to PDL15	AD6 to AD15		
AV$_{DD}$	–	–	–
AV$_{REF0}$	–	–	Connect to AV$_{SS}$ via a resistor.
AV$_{REF1}$	–	–	Connect to AV$_{SS}$ via a resistor.
AV$_{SS}$	–	–	–
EV$_{DD}$	–	–	–
EV$_{SS}$	–	–	–
FLMD0 Note 1	–	–	Connect to V$_{SS}$.
IC Note 2	–	–	–
$\overline{\text{RESET}}$	–	2	–
V$_{DD}$	–	–	–
V$_{SS}$	–	–	–
X1	–	–	–
X2	–	–	–
XT1	–	16	Connect to V$_{SS}$ via a resistor.
XT2	–	16	Leave open.

Notes 1. μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y only

2. μPD703200, 703200Y, 703201, 703201Y, 703204, and 703204Y only

Remark [ ]: Pins provided only in the V850ES/SA3

Figure 2-1. Pin I/O Circuits (1/2)



Type 2

Schmitt-triggered input with hysteresis characteristics.

Type 5-W

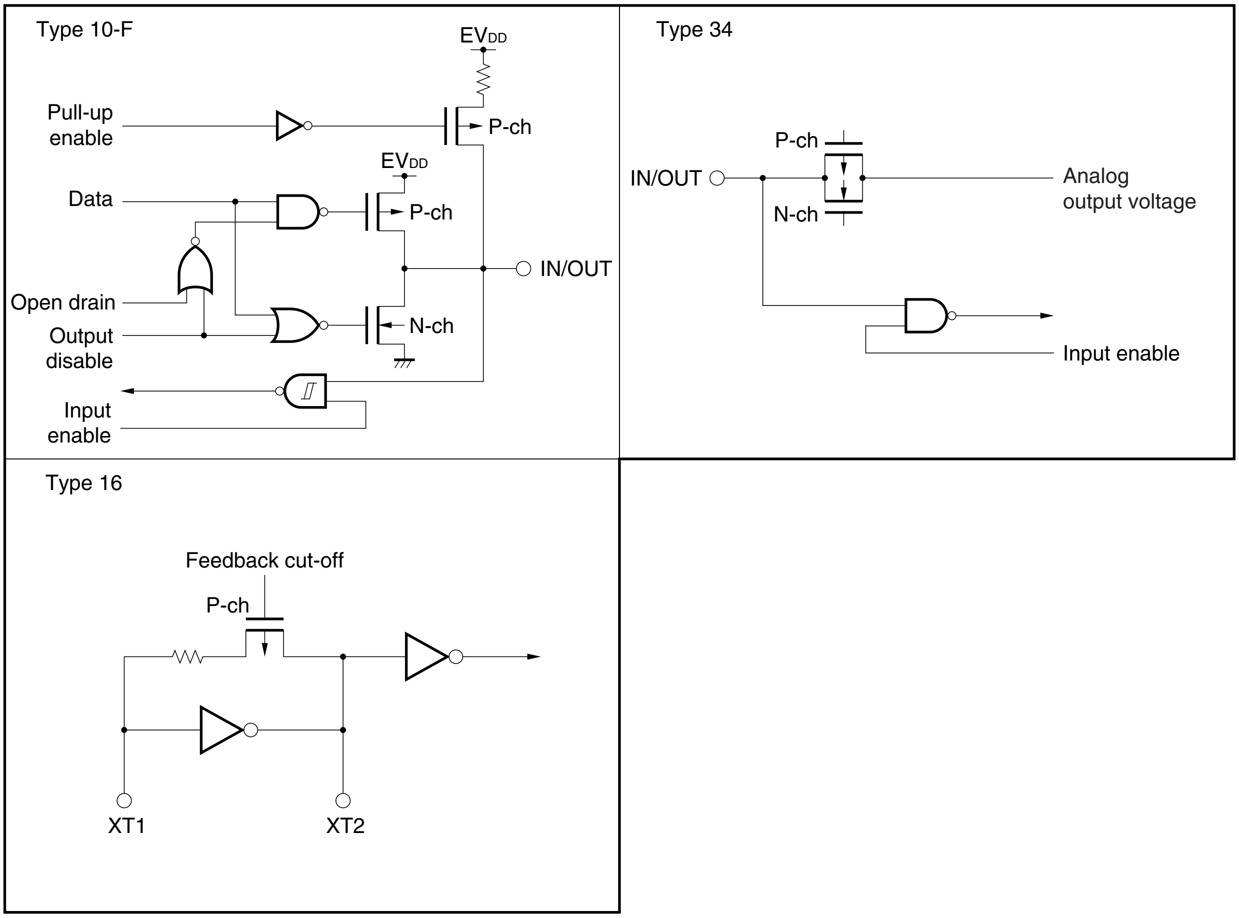
Type 5

Type 9

Type 5-A

Type 10-E

Figure 2-1. Pin I/O Circuits (2/2)



CHAPTER 3 CPU FUNCTION

The CPU of the V850ES/SA2 and V850ES/SA3 is based on RISC architecture and executes almost all instructions in one clock by using a 5-stage pipeline.

3.1 Features

★ ○ Minimum instruction execution time: 50 ns: Main clock = 20 MHz
(μPD703200, 703201, 703204, 70F3201, 70F3204)
59 ns: Main clock = 17 MHz
(μPD703200Y, 703201Y, 703204Y, 70F3201Y, 70F3204Y)
30.5 μs: Subclock = 32.768 kHz

○ Memory space Program space: 64 MB linear
Data space: 4 GB linear
• Memory block division function: 2, 2, 4, 8 MB/total: 4 blocks

○ General-purpose registers: 32 bits × 32 registers

○ Internal 32-bit architecture

○ 5-stage pipeline control

○ Multiplication/division instruction

○ Saturation operation instruction

○ 32-bit shift instruction: 1 clock

○ Load/store instruction with long/short format

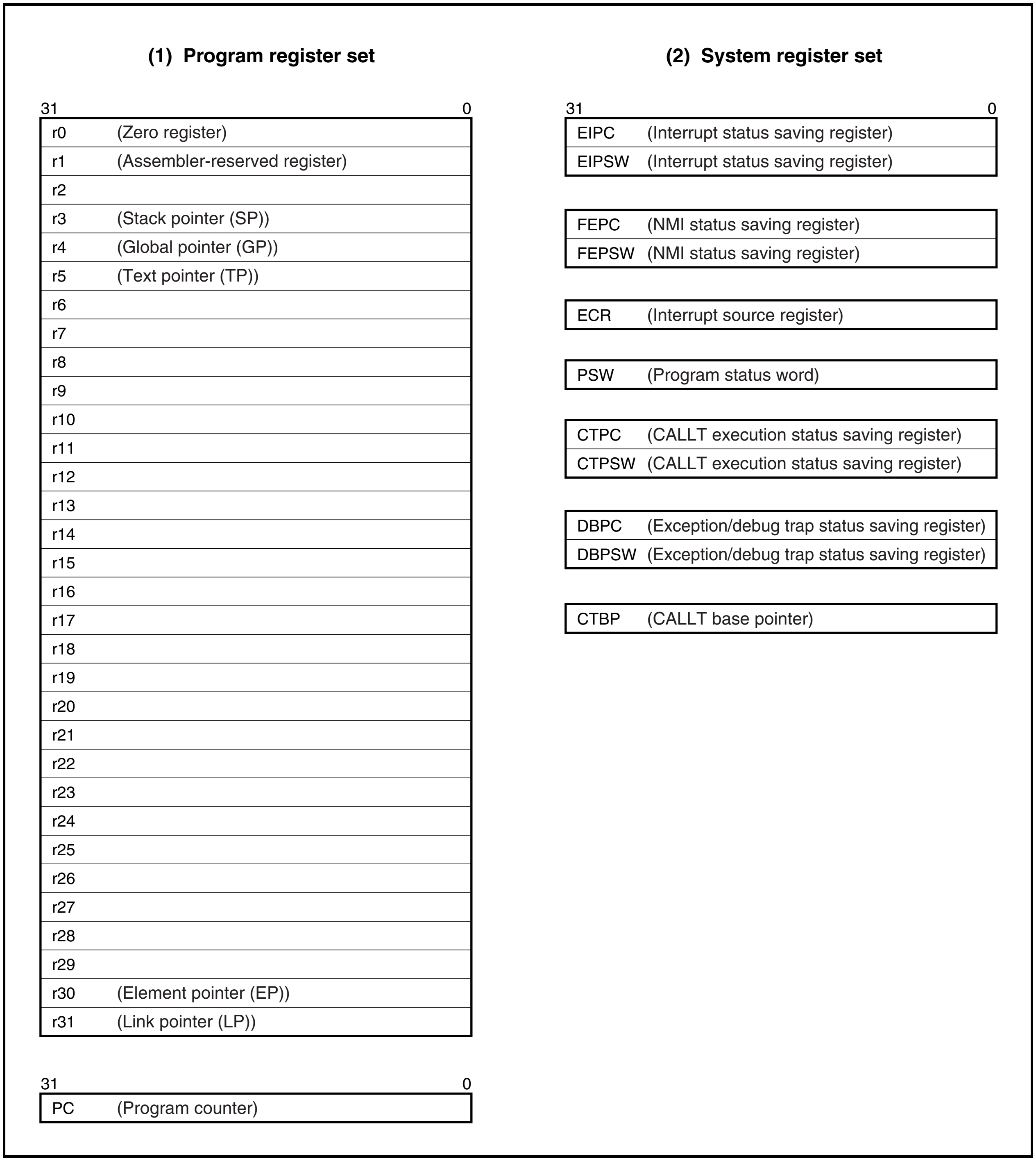
○ Four types of bit manipulation instructions

- SET1
- CLR1
- NOT1
- TST1

3.2 CPU Register Set

The registers of the V850ES/SA2 and V850ES/SA3 can be classified into two types: general-purpose program registers and dedicated system registers. All the registers are 32 bits wide.

For details, refer to the **V850ES Architecture User's Manual**.



(1) Program register set

31 0

r0	(Zero register)
r1	(Assembler-reserved register)
r2	
r3	(Stack pointer (SP))
r4	(Global pointer (GP))
r5	(Text pointer (TP))
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	(Element pointer (EP))
r31	(Link pointer (LP))

31 0

PC	(Program counter)

(2) System register set

31 0

EIPC	(Interrupt status saving register)
EIPSW	(Interrupt status saving register)

FEPC	(NMI status saving register)
FEPSW	(NMI status saving register)

ECR	(Interrupt source register)

PSW	(Program status word)

CTPC	(CALLT execution status saving register)
CTPSW	(CALLT execution status saving register)

DBPC	(Exception/debug trap status saving register)
DBPSW	(Exception/debug trap status saving register)

CTBP	(CALLT base pointer)

3.2.1 Program register set

The program registers include general-purpose registers and a program counter.

(1) General-purpose registers (r0 to r31)

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used to store a data variable or an address variable.

However, r0 and r30 are implicitly used by instructions and care must be exercised when these registers are used. r0 always holds 0 and is used for an operation that uses 0 or addressing of offset 0. r30 is used by the SLD and SST instructions as a base pointer when these instructions access the memory. r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. When using these registers, save their contents for protection, and then restore the contents after using the registers. r2 is sometimes used by the real-time OS. If the real-time OS does not use r2, it can be used as a register for variables.

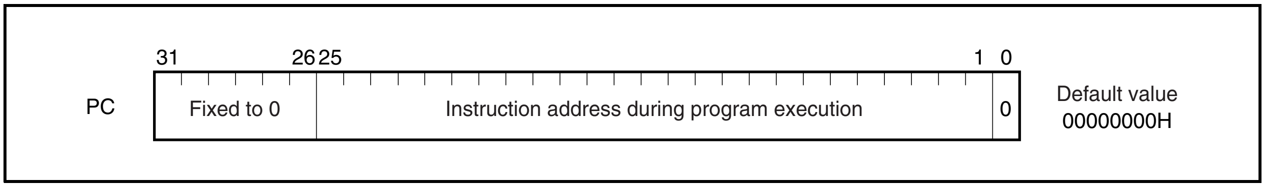
Table 3-1. Program Registers

Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as working register to create 32-bit immediate data
r2	Register for address/data variable (if real-time OS does not use r2)	
r3	Stack pointer	Used to create a stack frame when a function is called
r4	Global pointer	Used to access a global variable in the data area
r5	Text pointer	Used as register that indicates the beginning of a text area (area where program codes are located)
r6 to r29	Register for address/data variable	
r30	Element pointer	Used as base pointer to access memory
r31	Link pointer	Used when the compiler calls a function
PC	Program counter	Holds the instruction address during program execution

(2) Program counter (PC)

The program counter holds the instruction address during program execution. The lower 32 bits of this register are valid. Bits 31 to 26 are fixed to 0. A carry from bit 25 to 26 is ignored even if it occurs.

Bit 0 is fixed to 0. This means that execution cannot branch to an odd address.



3.2.2 System register set

The system registers control the status of the CPU and hold interrupt information.

These registers can be read or written by using system register load/store instructions (LDSR and STSR), using the system register numbers listed below.

Table 3-2. System Register Numbers

Register Number	System Register Name	Operand Specification	
		LDSR Instruction	STSR Instruction
0	Interrupt status saving register (EIPC)[Note 1]	√	√
1	Interrupt status saving register (EIPSW)[Note 1]	√	√
2	NMI status saving register (FEPC)[Note 1]	√	√
3	NMI status saving register (FEPSW)[Note 1]	√	√
4	Interrupt source register (ECR)	×	√
5	Program status word (PSW)	√	√
6 to 15	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×
16	CALLT execution status saving register (CTPC)	√	√
17	CALLT execution status saving register (CTPSW)	√	√
18	Exception/debug trap status saving register (DBPC)	√[Note 2]	√
19	Exception/debug trap status saving register (DBPSW)	√[Note 2]	√
20	CALLT base pointer (CTBP)	√	√
21 to 31	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×

Notes 1. Because only one set of this register is available, the contents of this register must be saved by program if multiple interrupts are enabled.

2. These registers can be accessed only when the DBTRAP instruction is executed.

Caution **Even if EIPC or FEPC, or bit 0 of CTPC is set to 1 by the LDSR instruction, bit 0 is ignored when execution is returned to the main routine by the RETI instruction after interrupt servicing (this is because bit 0 of the PC is fixed to 0). Set an even value to EIPC, FEPC, and CTPC (bit 0 = 0).**

Remark √: Can be accessed
×: Access prohibited

(1) Interrupt status saving registers (EIPC and EIPSW)

EIPC and EIPSW are used to save the status when an interrupt occurs.

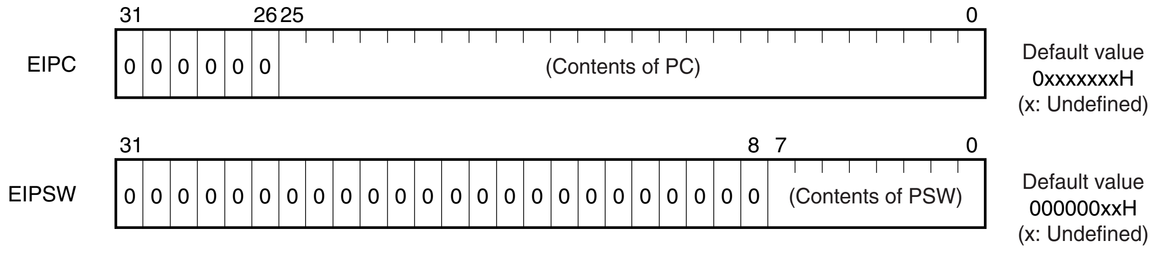
If a software exception or a maskable interrupt occurs, the contents of the program counter (PC) are saved to EIPC, and the contents of the program status word (PSW) are saved to EIPSW (these contents are saved to the NMI status saving registers (FEPC and FEPSW) if a non-maskable interrupt occurs).

The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to EIPC when a software exception or a maskable interrupt occurs.

The current contents of the PSW are saved to EIPSW.

Because only one set of interrupt status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled.

Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved for future function expansion (these bits are always fixed to 0).



(2) NMI status saving registers (FEPC and FEPSW)

FEPC and FEPSW are used to save the status when a non-maskable interrupt (NMI) occurs.

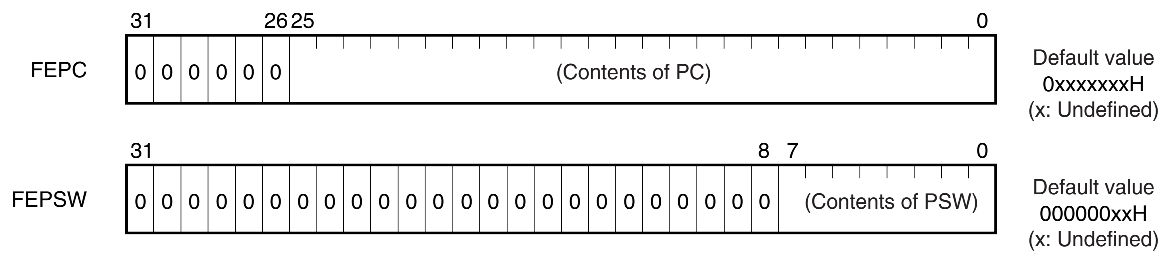
If an NMI occurs, the contents of the program counter (PC) are saved to FEPC, and those of the program status word (PSW) are saved to FEPSW.

The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

The current contents of the PSW are saved to FEPSW.

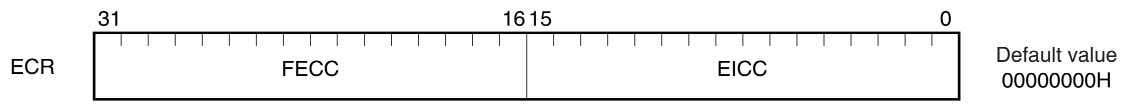
Because only one set of NMI status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled.

Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved for future function expansion (these bits are always fixed to 0).



(3) Interrupt source register (ECR)

The interrupt source register (ECR) holds the source of an exception or interrupt if an exception or interrupt occurs. This register holds the exception code of each interrupt source. Because this register is a read-only register, data cannot be written to this register using the LDSR instruction.



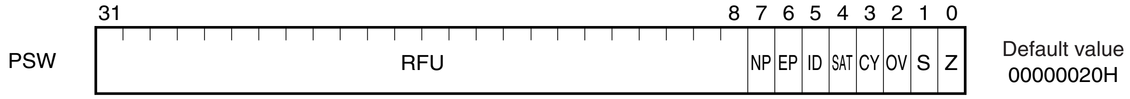
Bit position	Bit name	Meaning
31 to 16	FECC	Exception code of non-maskable interrupt (NMI)
15 to 0	EICC	Exception code of exception or maskable interrupt

(4) Program status word (PSW)

The program status word (PSW) is a collection of flags that indicate the status of the program (result of instruction execution) and the status of the CPU.

If the contents of a bit of this register are changed by using the LDSR instruction, the new contents are validated immediately after completion of LDSR instruction execution. If the ID flag is set to 1, however, interrupt request acknowledgment is disabled even while the LDSR instruction is being executed.

Bits 31 to 8 of this register are reserved for future function expansion (these bits are fixed to 0).



Bit position	Flag name	Meaning
31 to 8	RFU	Reserved field. Fixed to 0.
7	NP	Indicates that a non-maskable interrupt (NMI) is being serviced. This bit is set to 1 when an NMI request is acknowledged, disabling multiple interrupts. 0: NMI is not being serviced. 1: NMI is being serviced.
6	EP	Indicates that an exception is being processed. This bit is set to 1 when an exception occurs. Even if this bit is set, interrupt requests are acknowledged. 0: Exception is not being processed. 1: Exception is being processed.
5	ID	Indicates whether a maskable interrupt can be acknowledged. 0: Interrupt enabled (EI) 1: Interrupt disabled (DI)
4	SAT[Note]	Indicates that the result of a saturation operation has overflowed and is saturated. Because this is a cumulative flag, it is set to 1 when the result of a saturation operation instruction is saturated, and is not cleared to 0 even if the subsequent operation result is not saturated. Use the LDSR instruction to clear this bit. This flag is neither set to 1 nor cleared to 0 by execution of an arithmetic operation instruction. 0: Not saturated 1: Saturated
3	CY	Indicates whether a carry or a borrow occurs as a result of an operation. 0: Carry or borrow does not occur. 1: Carry or borrow occurs.
2	OV[Note]	Indicates whether an overflow occurs during operation. 0: Overflow does not occur. 1: Overflow occurs.
1	S[Note]	Indicates whether the result of an operation is negative. 0: The result is positive or 0. 1: The result is negative.
0	Z	Indicates whether the result of an operation is 0. 0: The result is not 0. 1: The result is 0.

Remark Also read **Note** on the next page.

Note The result of the operation that has performed saturation processing is determined by the contents of the OV and S flags. The SAT flag is set to 1 only when the OV flag is set to 1 when a saturation operation is performed.

Status of Operation Result	Flag Status			Result of Operation of Saturation Processing
	SAT	OV	S	
Maximum positive value is exceeded.	1	1	0	7FFFFFFFH
Maximum negative value is exceeded.	1	1	1	80000000H
Positive (maximum value is not exceeded)	Holds value before operation	0	0	Operation result itself
Negative (maximum value is not exceeded)			1	

(5) CALLT execution status saving registers (CTPC and CTPSW)

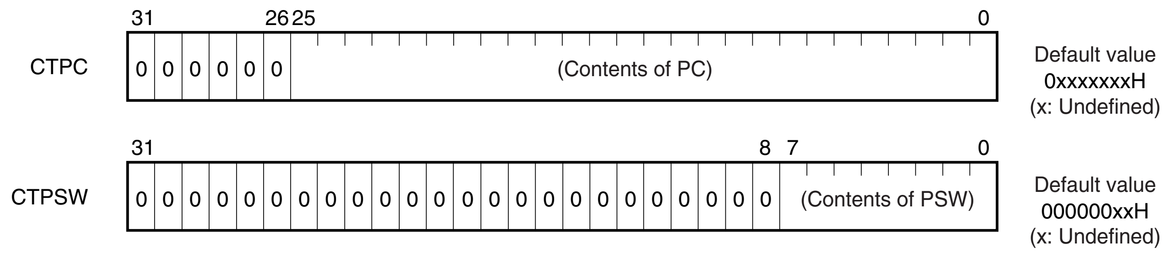
CTPC and CTPSW are CALLT execution status saving registers.

When the CALLT instruction is executed, the contents of the program counter (PC) are saved to CTPC, and those of the program status word (PSW) are saved to CTPSW.

The contents saved to CTPC are the address of the instruction next to CALLT.

The current contents of the PSW are saved to CTPSW.

Bits 31 to 26 of CTPC and bits 31 to 8 of CTPSW are reserved for future function expansion (fixed to 0).



(6) Exception/debug trap status saving registers (DBPC and DBPSW)

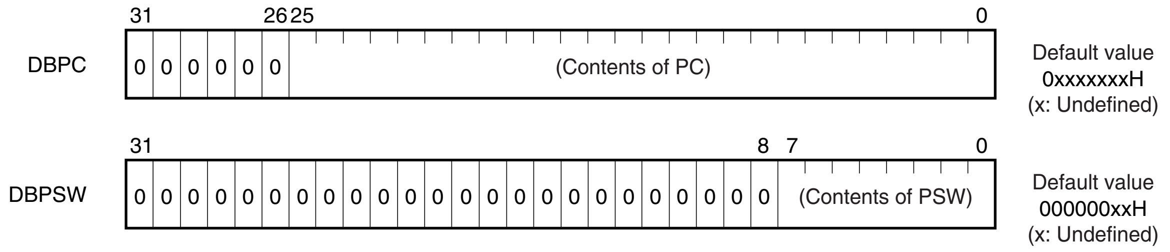
DBPC and DBPSW are exception/debug trap status registers.

If an exception trap or debug trap occurs, the contents of the program counter (PC) are saved to DBPC, and those of the program status word (PSW) are saved to DBPSW.

The contents to be saved to DBPC are the address of the instruction next to the one that is executed when an exception trap or debug trap occurs.

The current contents of the PSW are saved to DBPSW.

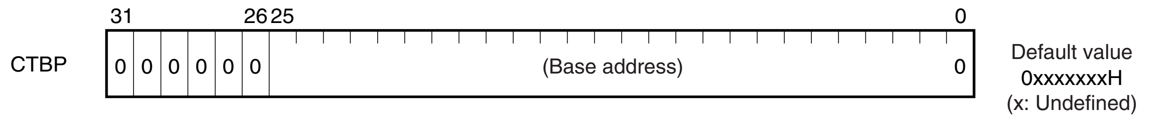
Bits 31 to 26 of DBPC and bits 31 to 8 of DBPSW are reserved for future function expansion (fixed to 0).



(7) CALLT base pointer (CTBP)

The CALLT base pointer (CTBP) is used to specify a table address or generate a target address (bit 0 is fixed to 0).

Bits 31 to 26 of this register are reserved for future function expansion (fixed to 0).



3.3 Operation Modes

The V850ES/SA2 and V850ES/SA3 have the following operation modes.

(1) Single-chip mode

In this mode, each pin related to the bus interface is set to the port mode after system reset has been released. Execution branches to the reset entry address of the internal ROM, and then instruction processing is started. By setting the PMCDH, PMCDL, PMCCM, PMCCS, and PMCCT registers to the control mode by software, an external device can be connected to the external memory area.

(2) Flash memory programming mode (μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y)

In this mode, the internal flash memory can be programmed by using a flash programmer.

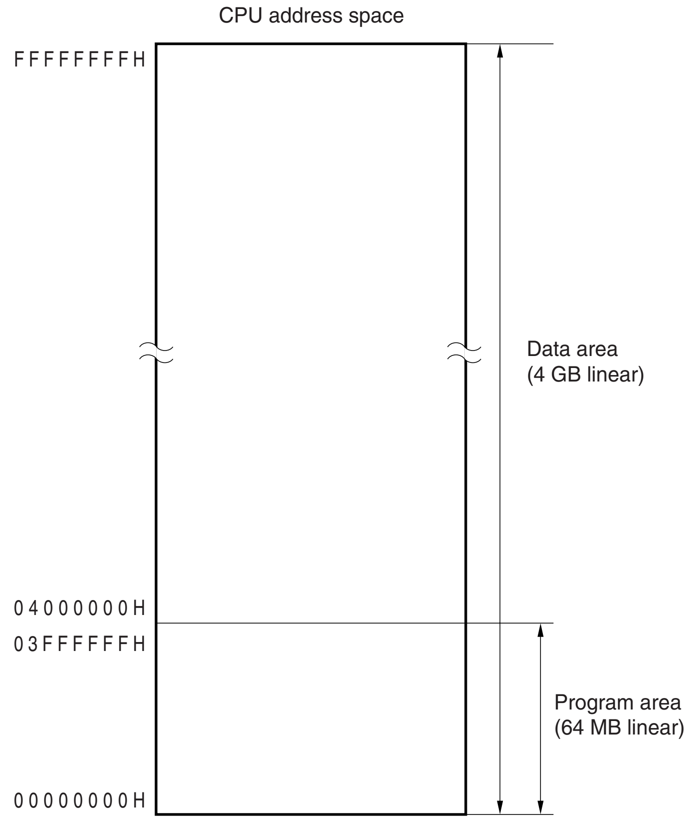
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850ES/SA2 and V850ES/SA3 has 32-bit architecture and supports up to 4 GB of linear address space (data space) for operand addressing (data access). It also supports up to 64 MB of linear address space (program space) for instruction addressing. Note, however, that both the program and data spaces have areas that are prohibited from being used. For details, refer to **Figure 3-2**.

Figure 3-1 shows the CPU address space.

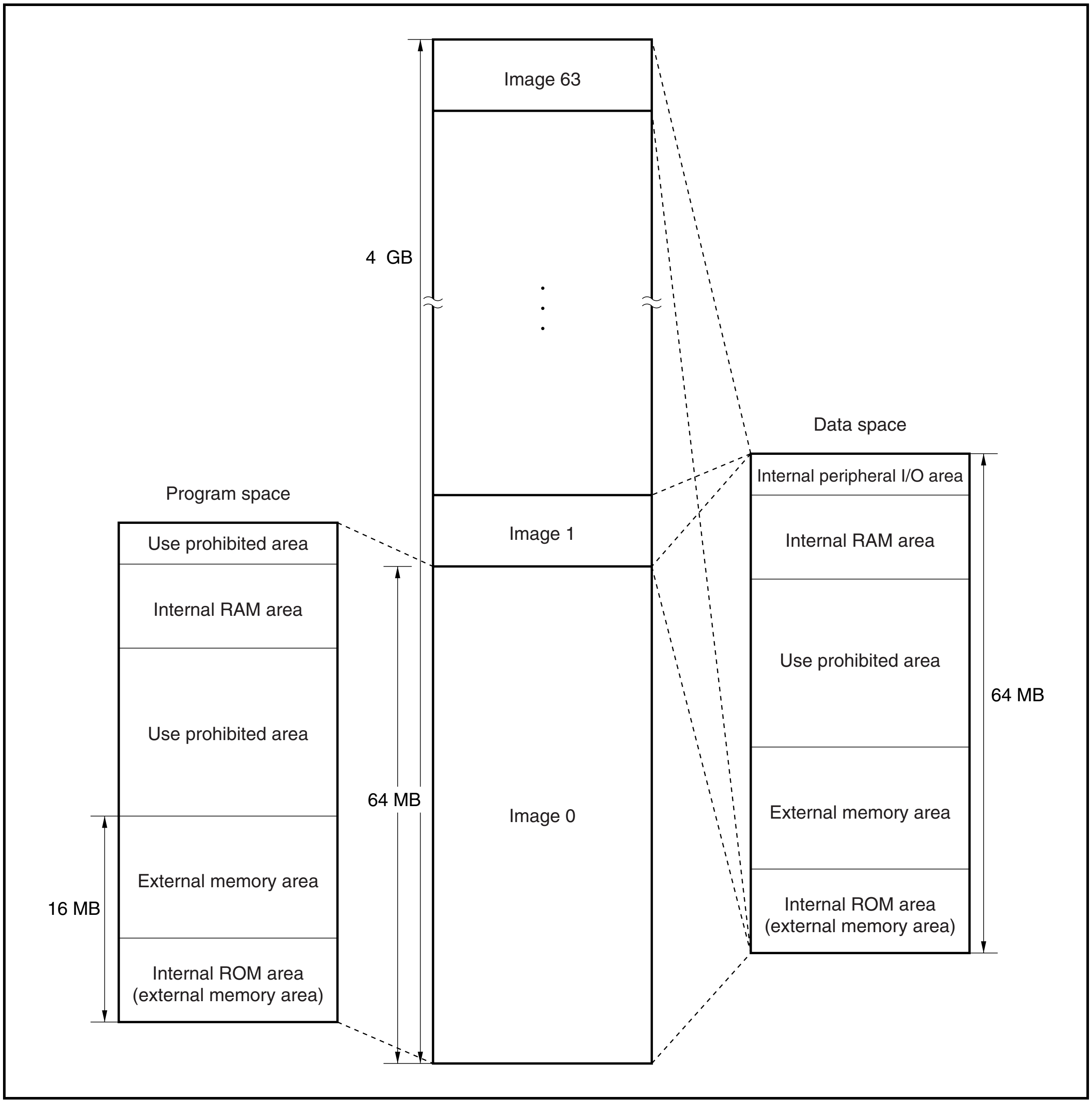
Figure 3-1. CPU Address Space



3.4.2 Image

For instruction addressing, up to 16 MB of linear address space (program space) and an internal RAM area are supported. Up to 4 GB of linear address space (data space) is supported for operand addressing (data access). In the 4 GB address space, it seems that there are sixty-four 64 MB physical address spaces. This means that the same 64 MB physical address space is accessed, regardless of the values of bits 31 to 26.

Figure 3-2. Image on Address Space



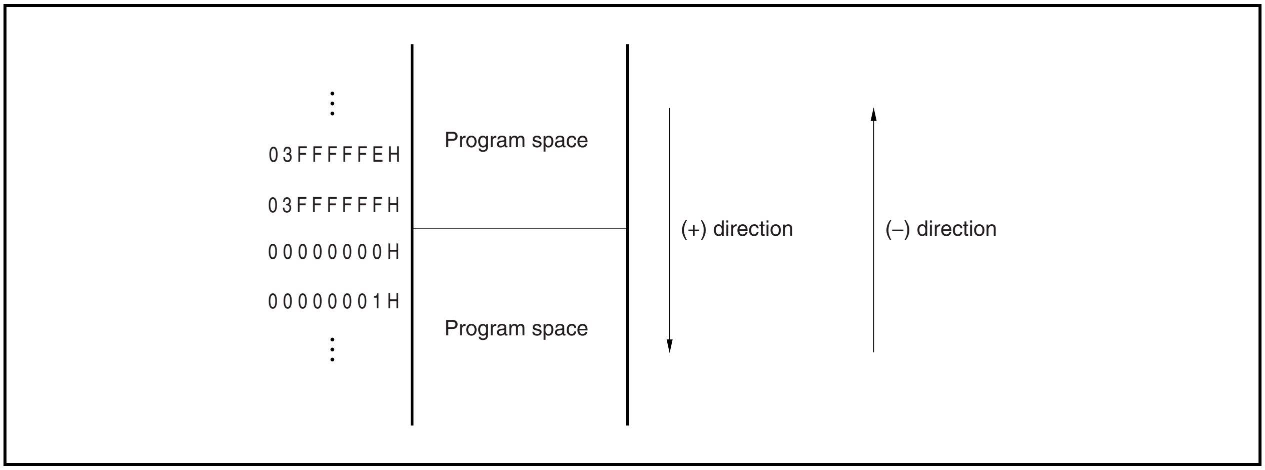
3.4.3 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0 and only the lower 26 bits are valid. The higher 6 bits ignore a carry or borrow from bit 25 to 26 during branch address calculation.

Therefore, the lowest address of the program space, 00000000H, and the highest address, 03FFFFFFH, are contiguous addresses. That the lowest address and the highest address of the program space are contiguous in this way is called wrap-around.

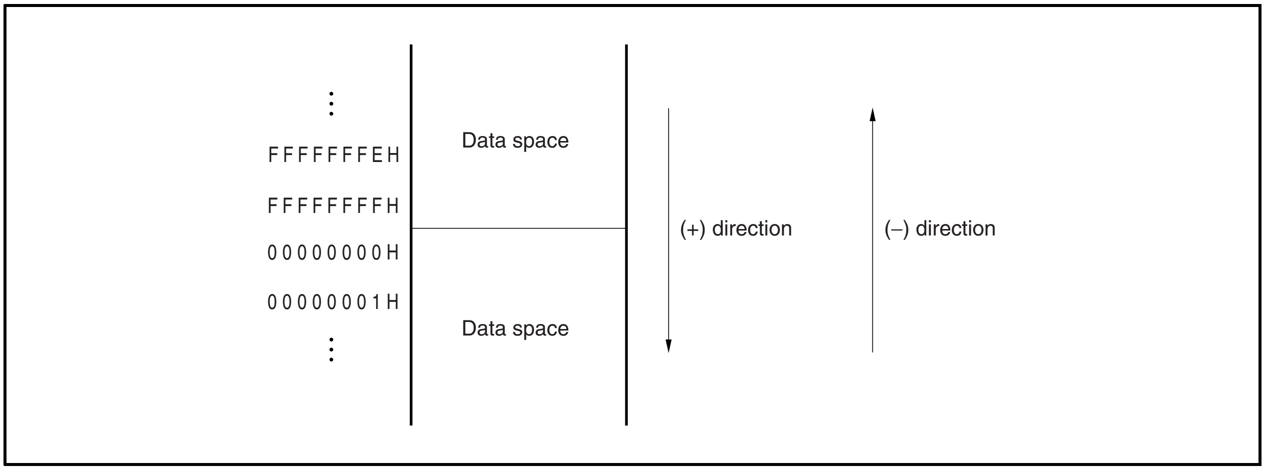
Caution Because the 4 KB area of addresses 03FFF000H to 03FFFFFFH is an internal peripheral I/O area, instructions cannot be fetched from this area. Therefore, do not execute an operation in which the result of a branch address calculation affects this area.



(2) Data space

The result of an operand address calculation operation that exceeds 32 bits is ignored.

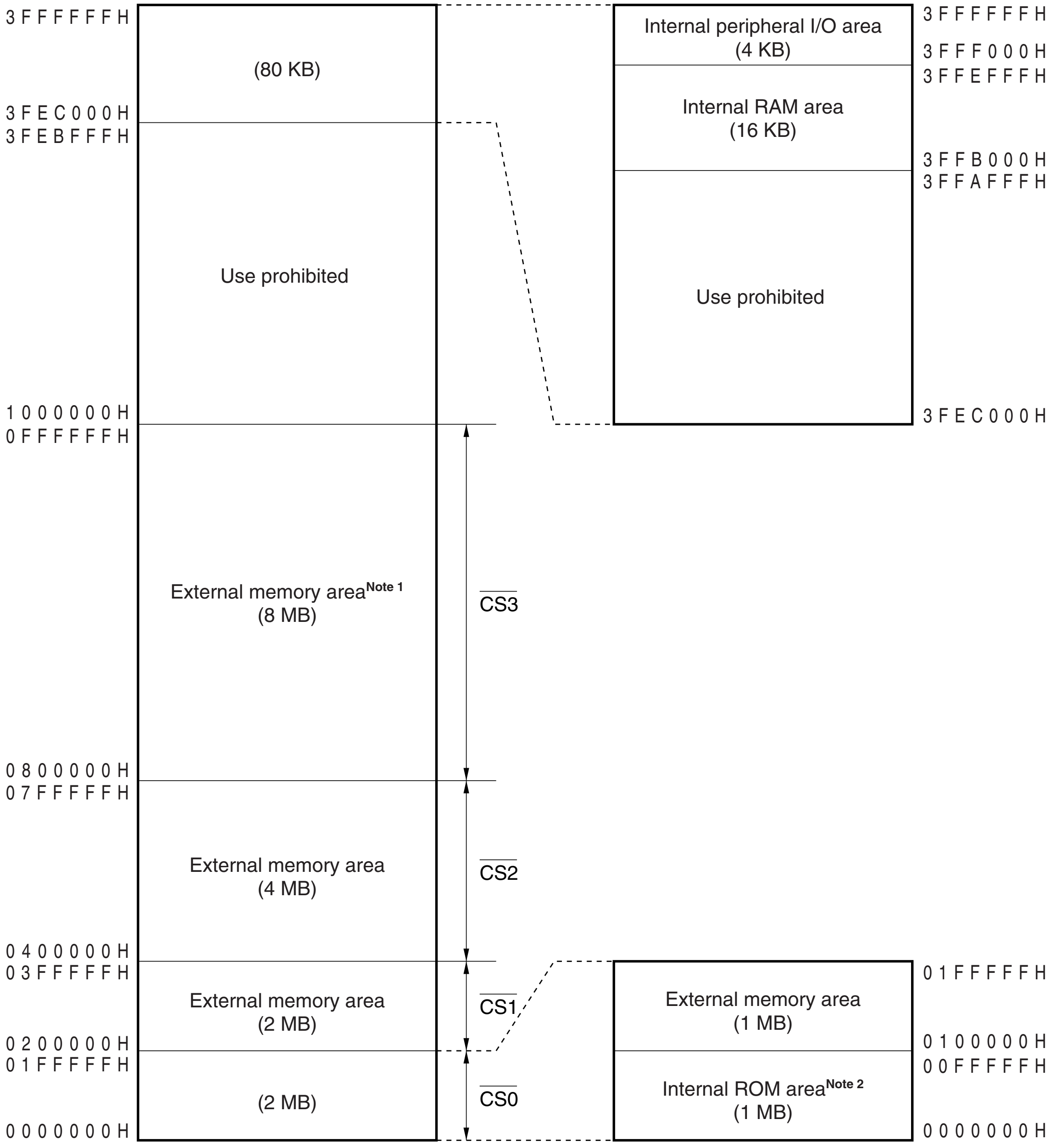
Therefore, the lowest address of the data space, 00000000H, and the highest address, FFFFFFFFH, are contiguous, and wrap-around occurs at the boundary of these addresses.



3.4.4 Memory map

The V850ES/SA2 and V850ES/SA3 reserve the areas shown in Figure 3-3.

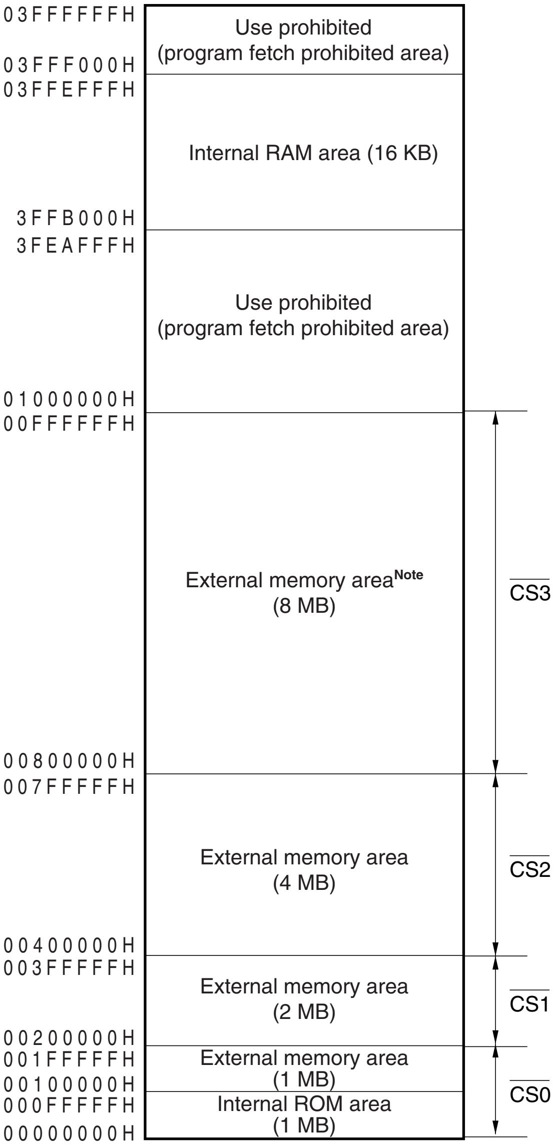
Figure 3-3. Data Memory Map (Physical Addresses)



Notes 1. This is the 4 MB space of 0800000H to 0BFFFFFH in the V850ES/SA2 (the area of 0C00000H to 0FFFFFFH is the image of 0800000H to 0BFFFFFH).

2. Fetch access and read access to addresses 0000000H to 00FFFFFH is made to the internal ROM area. However, data write access to these addresses is made to the external memory area.

Figure 3-4. Program Memory Map



Note This is the 4 MB space of 0800000H to 0BFFFFFH in the V850ES/SA2 (the area of 0C00000H to 0FFFFFFH is the image of 0800000H to 0BFFFFFH).

Remark Instructions can be executed to the external memory area without execution branching from the internal ROM area to the external memory area.

3.4.5 Areas

(1) Internal ROM area

(a) Memory map

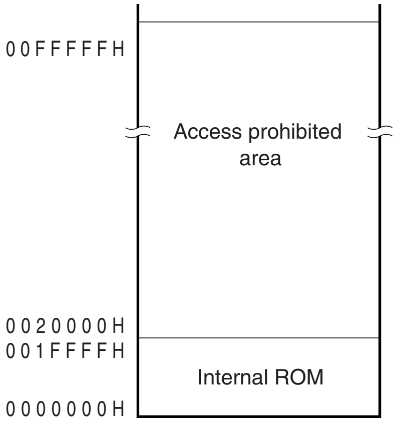
1 MB of addresses 0000000H to 00FFFFFH is reserved as the internal ROM area.

<1> *μ*PD703200 and 703200Y

128 KB are allocated to the following addresses as the internal physical ROM (mask ROM).

- Addresses 0000000H to 001FFFFH

Figure 3-5. Internal ROM Area (128 KB)



<2> *μ*PD703201, 703201Y, 703204, and 703204Y

256 KB are allocated to the following addresses as the internal physical ROM (mask ROM).

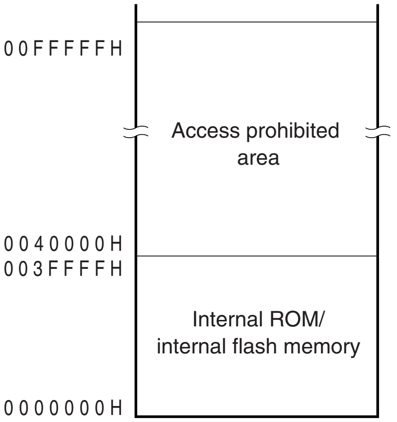
- 0000000H to 003FFFFH

<3> *μ*PD70F3201, 70F3201Y, 70F3204, and 70F3204Y

256 KB are allocated to the following addresses as the internal physical ROM (flash memory).

- 0000000H to 003FFFFH

Figure 3-6. Internal ROM/Internal Flash Memory Area (256 KB)



- Interrupt/exception table

  The V850ES/SA2 and V850ES/SA3 speed up the interrupt response time by fixing handler addresses corresponding to interrupts/exceptions.

  A collection of these handler addresses is called an interrupt/exception table, which is mapped to the internal ROM area. When an interrupt/exception is acknowledged, execution jumps to a handler address and the program in the area starting from that address is executed. Table 3-3 shows the interrupt/exception sources and corresponding addresses.

Table 3-3. Interrupt/Exception Table

First Address of Interrupt/Exception Table	Interrupt/Exception Source	First Address of Interrupt/Exception Table	Interrupt/Exception Source
00000000H	RESET	00000180H	INTTM3
00000010H	NMI	00000190H	INTTM4
00000020H	INTWDT	000001A0H	INTTM5
00000040H	TRAP0n (n = 0 to F)	000001B0H	INTCSI0
00000050H	TRAP1n (n = 0 to F)	000001C0H	INTIIC[Note 1]
00000060H	ILGOP/DBG0	000001D0H	INTCSI1
00000080H	INTWDTM	000001E0H	INTSRE0
00000090H	INTP0	000001F0H	INTSR0
000000A0H	INTP1	00000200H	INTST0
000000B0H	INTP2	00000210H	INTCSI2
000000C0H	INTP3	00000220H	INTSRE1
000000D0H	INTP4	00000230H	INTSR1
000000E0H	INTP5	00000240H	INTST1
000000F0H	INTP6	00000250H	INTCSI3
00000100H	INTRTC	00000260H	INTCSI4[Note 2]
00000110H	INTCC00	00000270H	INTAD
00000120H	INTCC01	00000280H	INTDMA0
00000130H	INTOVF0	00000290H	INTDMA1
00000140H	INTCC10	000002A0H	INTDMA2
00000150H	INTCC11	000002B0H	INTDMA3
00000160H	INTOVF1	000002C0H	INTROV
00000170H	INTTM2	000002D0H	INTBRG

Notes 1. μPD703200Y, 703201, 703204, 70F3201Y, and 70F3204Y only

2. V850ES/SA3 only

(2) Internal RAM area

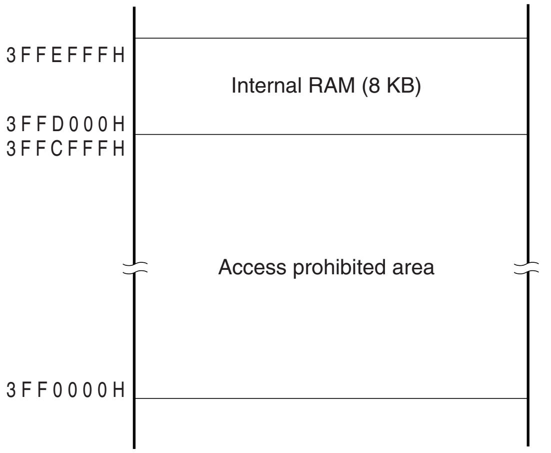
60 KB of addresses 3FF0000H to 3FFEFFFH are reserved as the internal RAM area.

<1> μPD703200 and 703200Y

8 KB are allocated to the following addresses as the internal physical RAM.

- 3FFD000H to 3FFEFFFH

Figure 3-7. Internal RAM Area (8 KB)

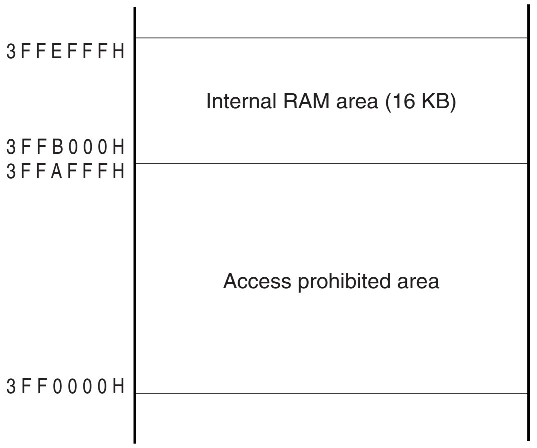


<2> μPD703201, 703201Y, 703204, 703204Y, 70F3201, 70F3201Y, 70F3204, and 70F3204Y

16 KB are allocated to the following addresses as the internal physical RAM.

- 3FFB000H to 3FFEFFFH

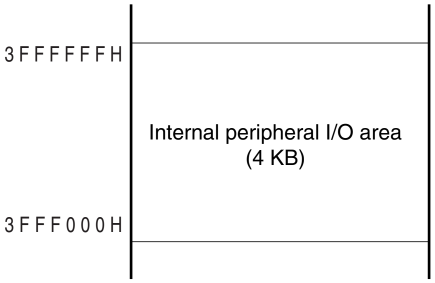
Figure 3-8. Internal RAM Area (16 KB)



(3) Internal peripheral I/O area

4 KB of addresses 3FFF000H to 3FFFFFFH are allocated as the internal peripheral I/O area.

Figure 3-9. Internal Peripheral I/O Area



Peripheral I/O registers that have functions to specify the operation mode for and monitor the status of the internal peripheral I/O are mapped to the internal peripheral I/O area. Program cannot be fetched from this area.

Cautions 1. When a register is accessed in word units, a word area is accessed twice in halfword units in the order of lower area and higher area, with the lower 2 bits of the address ignored.

2. If a register that can be accessed in byte units is accessed in halfword units, the higher 8 bits are undefined when the register is read, and data is written to the lower 8 bits.

3. Addresses not defined as registers are reserved for future expansion. The operation is undefined and not guaranteed when these addresses are accessed.

(4) External memory area

15 MB (0100000H to 0FFFFFFH) are allocated as the external memory area. For details, refer to **CHAPTER 5 BUS CONTROL FUNCTION**.

3.4.6 Recommended use of address space

The architecture of the V850ES/SA2 and V850ES/SA3 requires that a register that serves as a pointer be secured for address generation when operand data in the data space is accessed. The address stored in this pointer ±32 KB can be directly accessed by an instruction for operand data. Because the number of general-purpose registers that can be used as a pointer is limited, however, by keeping the performance from dropping during address calculation when a pointer value is changed, as many general-purpose registers as possible can be secured for variables, and the program size can be reduced.

(1) Program space

Of the 32 bits of the program counter (PC), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Regarding the program space, therefore, a 64 MB space of contiguous addresses starting from 00000000H unconditionally corresponds to the memory map.

To use the internal RAM area as the program space, access addresses 3FFC000H to 3FFEFFFH.

(2) Data space

With the V850ES/SA2 and V850ES/SA3, it seems that there are sixty-four 64 MB address spaces on the 4 GB CPU address space. Therefore, the least significant bit (bit 25) of a 26-bit address is sign-extended to 32 bits and allocated as an address.

(a) Application example of wrap-around

If R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, a range of addresses 00000000H ±32 KB can be addressed by sign-extended disp16. All the resources of the internal hardware can be addressed by one pointer.

The zero register (r0) is a register fixed to 0 by hardware, and practically eliminates the need for registers dedicated to pointers.

Figure 3-10. Application Example of Wrap-Around (V850E/SA3)

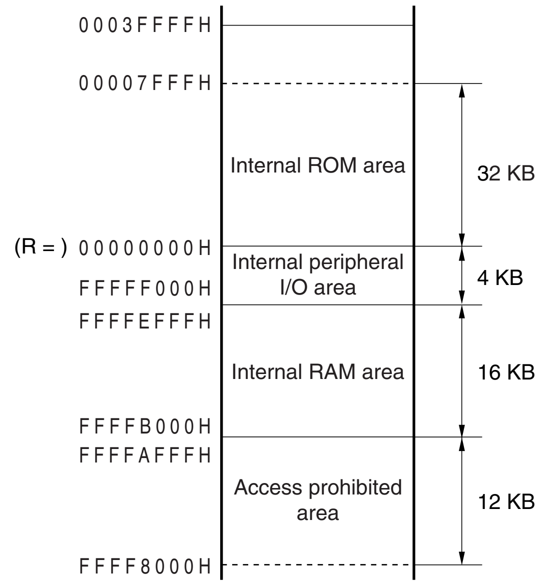
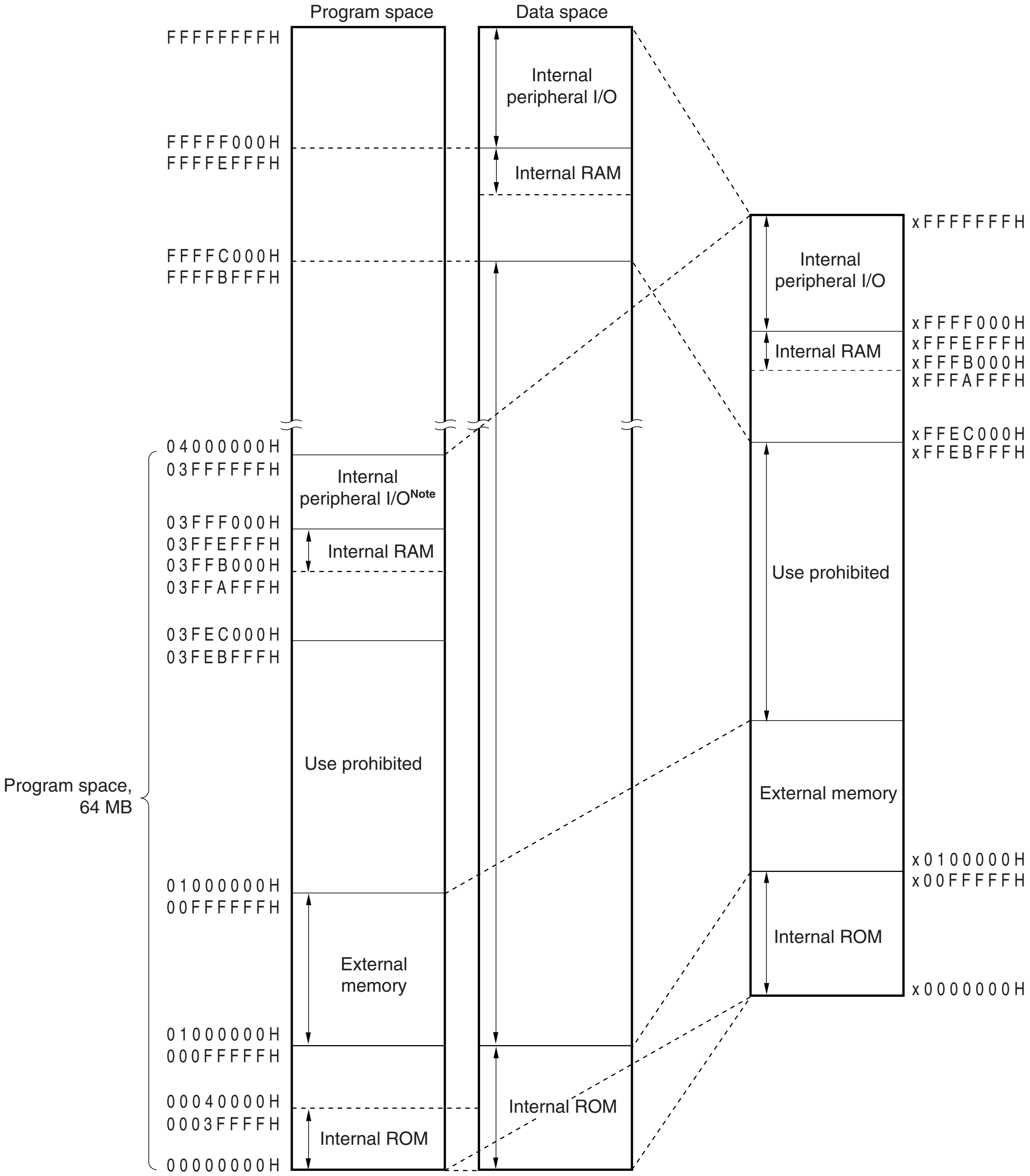


Figure 3-11. Recommended Memory Map



Note Access to this area is prohibited. To access the internal I/O in this area, specify addresses FFFF000H to FFFFFFFH.

Remarks 1. ↕ indicates the recommended area.

2. This figure is the recommended memory map of the μPD703204.

3.4.7 Peripheral I/O registers

(1/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port register DL	PDL	R/W			√	Undefined
FFFFF004H	Port register DLL	PDLL		√	√		
FFFFF005H	Port register DLH	PDLH		√	√		
FFFFF006H	Port register DH	PDH		√	√		
FFFFF008H	Port register CS	PCS		√	√		
FFFFF00AH	Port register CT	PCT		√	√		
FFFFF00CH	Port register CM	PCM		√	√		
FFFFF00EH	Port register CD[Note]	PCD		√	√		
FFFFF024H	Port mode register DL	PMDL				√	FFFFH
FFFFF024H	Port mode register DLL	PMDLL		√	√		FFH
FFFFF025H	Port mode register DLH	PMDLH		√	√		
FFFFF026H	Port mode register DH	PMDH		√	√		
FFFFF028H	Port mode register CS	PMCS		√	√		
FFFFF02AH	Port mode register CT	PMCT		√	√		
FFFFF02CH	Port mode register CM	PMCM		√	√		
FFFFF02EH	Port mode register CD[Note]	PMCD		√	√		
FFFFF044H	Port mode control register DL	PMCDL				√	0000H
FFFFF044H	Port mode control register DLL	PMCDLL		√	√		00H
FFFFF045H	Port mode control register DLH	PMCDLH		√	√		
FFFFF046H	Port mode control register DH	PMCDH		√	√		
FFFFF048H	Port mode control register CS	PMCCS		√	√		
FFFFF04AH	Port mode control register CT	PMCCT		√	√		
FFFFF04CH	Port mode control register CM	PMCCM		√	√		
FFFFF066H	Bus size configuration register	BSC				√	5555H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H				√	
FFFFF084H	DMA destination address register 0L	DDA0L				√	
FFFFF086H	DMA destination address register 0H	DDA0H				√	
FFFFF088H	DMA source address register 1L	DSA1L				√	
FFFFF08AH	DMA source address register 1H	DSA1H				√	
FFFFF08CH	DMA destination address register 1L	DDA1L				√	
FFFFF08EH	DMA destination address register 1H	DDA1H				√	
FFFFF090H	DMA source address register 2L	DSA2L				√	
FFFFF092H	DMA source address register 2H	DSA2H				√	
FFFFF094H	DMA destination address register 2L	DDA2L				√	
FFFFF096H	DMA destination address register 2H	DDA2H				√	
FFFFF098H	DMA source address register 3L	DSA3L				√	
FFFFF09AH	DMA source address register 3H	DSA3H				√	
FFFFF09CH	DMA destination address register 3L	DDA3L				√	
FFFFF09EH	DMA destination address register 3H	DDA3H				√	

Note V850ES/SA3 only

(2/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF0C0H	DMA transfer count register 0	DBC0	R/W			√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1				√	
FFFFF0C4H	DMA transfer count register 2	DBC2				√	
FFFFF0C6H	DMA transfer count register 3	DBC3				√	
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	
FFFFF0D4H	DMA addressing control register 2	DADC2				√	
FFFFF0D6H	DMA addressing control register 3	DADC3				√	
FFFFF0E0H	DMA channel control register 0	DCHC0		√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		
FFFFF0E4H	DMA channel control register 2	DCHC2		√	√		
FFFFF0E6H	DMA channel control register 3	DCHC3		√	√		
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF110H	Interrupt control register	WDTIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		
FFFFF114H	Interrupt control register	PIC1		√	√		
FFFFF116H	Interrupt control register	PIC2		√	√		
FFFFF118H	Interrupt control register	PIC3		√	√		
FFFFF11AH	Interrupt control register	PIC4		√	√		
FFFFF11CH	Interrupt control register	PIC5		√	√		
FFFFF11EH	Interrupt control register	PIC6		√	√		
FFFFF120H	Interrupt control register	RTCIC		√	√		
FFFFF122H	Interrupt control register	CCIC00		√	√		
FFFFF124H	Interrupt control register	CCIC01		√	√		
FFFFF126H	Interrupt control register	OVFIC0		√	√		
FFFFF128H	Interrupt control register	CCIC10		√	√		
FFFFF12AH	Interrupt control register	CCIC11		√	√		
FFFFF12CH	Interrupt control register	OVFIC1		√	√		
FFFFF12EH	Interrupt control register	TMIC2		√	√		
FFFFF130H	Interrupt control register	TMIC3		√	√		
FFFFF132H	Interrupt control register	TMIC4		√	√		
FFFFF134H	Interrupt control register	TMIC5		√	√		
FFFFF136H	Interrupt control register	CSIIC0		√	√		
FFFFF138H	Interrupt control register[Note]	IICIC		√	√		

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

(3/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF13AH	Interrupt control register	CSIIC1	R/W	√	√		47H
FFFFF13CH	Interrupt control register	SREIC0		√	√		
FFFFF13EH	Interrupt control register	SRIC0		√	√		
FFFFF140H	Interrupt control register	STIC0		√	√		
FFFFF142H	Interrupt control register	CSIIC2		√	√		
FFFFF144H	Interrupt control register	SREIC1		√	√		
FFFFF146H	Interrupt control register	SRIC1		√	√		
FFFFF148H	Interrupt control register	STIC1		√	√		
FFFFF14AH	Interrupt control register	CSIIC3		√	√		
FFFFF14CH	Interrupt control register[Note]	CSIIC4		√	√		
FFFFF14EH	Interrupt control register	ADIC		√	√		
FFFFF150H	Interrupt control register	DMAIC0		√	√		
FFFFF152H	Interrupt control register	DMAIC1		√	√		
FFFFF154H	Interrupt control register	DMAIC2		√	√		
FFFFF156H	Interrupt control register	DMAIC3		√	√		
FFFFF158H	Interrupt control register	ROVIC		√	√		
FFFFF15AH	Interrupt control register	BRGIC		√	√		
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D converter mode register	ADM		√	√		
FFFFF201H	Analog input channel specification register	ADS			√		
FFFFF202H	Power fail comparison mode register	PFM		√	√		
FFFFF203H	Power fail comparison threshold value register	PFT			√		
FFFFF204H	A/D conversion result register	ADCR	R			√	Undefined
FFFFF205H	A/D conversion result register H	ADCRH			√		
FFFFF280H	D/A converter conversion value setting register 0	DACS0	R/W		√		00H
FFFFF282H	D/A converter conversion value setting register 1	DACS1			√		
FFFFF288H	D/A converter mode register	DAM		√	√		
FFFFF400H	Port register 0	P0		√	√		Undefined
FFFFF404H	Port register 2[Note]	P2		√	√		
FFFFF406H	Port register 3	P3		√	√		
FFFFF408H	Port register 4	P4		√	√		
FFFFF40EH	Port register 7	P7	R			√	
FFFFF40EH	Port register 7L	P7L		√	√		
FFFFF40FH	Port register 7H	P7H		√	√		
FFFFF410H	Port register 8	P8		√	√		
FFFFF412H	Port register 9	P9	R/W			√	
FFFFF412H	Port register 9L	P9L		√	√		
FFFFF413H	Port register 9H	P9H		√	√		

Note V850ES/SA3 only

(4/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF420H	Port mode register 0	PM0	R/W	√	√		FFH
FFFFF424H	Port mode register 2[Note]	PM2		√	√		
FFFFF426H	Port mode register 3	PM3		√	√		
FFFFF428H	Port mode register 4	PM4		√	√		
FFFFF432H	Port mode register 9	PM9				√	FFFFH
FFFFF432H	Port mode register 9L	PM9L		√	√		FFH
FFFFF433H	Port mode register 9H	PM9H		√	√		
FFFFF440H	Port mode control register 0	PMC0		√	√		00H
FFFFF444H	Port mode control register 2[Note]	PMC2		√	√		
FFFFF446H	Port mode control register 3	PMC3		√	√		
FFFFF448H	Port mode control register 4	PMC4		√	√		
FFFFF452H	Port mode control register 9	PMC9				√	0000H
FFFFF452H	Port mode control register 9L	PMC9L		√	√		00H
FFFFF453H	Port mode control register 9H	PMC9H		√	√		
FFFFF466H	Port function control register 3	PFC3		√	√		
FFFFF468H	Port function control register 4	PFC4		√	√		
FFFFF472H	Port function control register 9	PFC9				√	0000H
FFFFF472H	Port function control register 9L	PFC9L		√	√		00H
FFFFF473H	Port function control register 9H	PFC9H		√	√		
FFFFF484H	Data wait control register 0	DWC0				√	7777H
FFFFF488H	Address wait control register	AWC				√	FFFFH
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH
FFFFF600H	Timer 0	TM0	R			√	0000H
FFFFF602H	Capture/compare register 00	CC00	R/W			√	
FFFFF604H	Capture/compare register 01	CC01				√	
FFFFF606H	Timer control register 00	TMC00		√	√		00H
FFFFF608H	Timer control register 01	TMC01		√	√		20H
FFFFF609H	Valid edge select register 0	SES0			√		00H
FFFFF610H	Timer 1	TM1	R			√	0000H
FFFFF612H	Capture/compare register 10	CC10	R/W			√	
FFFFF614H	Capture/compare register 11	CC11				√	
FFFFF616H	Timer control register 10	TMC10		√	√		00H
FFFFF618H	Timer control register 11	TMC11		√	√		20H
FFFFF619H	Valid edge select register 1	SES1			√		00H
FFFFF640H	Timer counter 23	TM23	R			√	0000H
FFFFF640H	Timer counter 2	TM2			√		00H
FFFFF641H	Timer counter 3	TM3			√		
FFFFF642H	Compare register 23	CR23	R/W			√	0000H
FFFFF642H	Compare register 2	CR2			√		00H
FFFFF643H	Compare register 3	CR3			√		

Note V850ES/SA3 only

(5/8)

Address		Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
					1	8	16	
FFFFF644H		Timer clock selection register 23	TCL23	R/W			√	0000H
	FFFFF644H	Timer clock selection register 2	TCL2			√		00H
	FFFFF645H	Timer clock selection register 3	TCL3			√		
FFFFF646H		Timer mode control register 23	TMC23				√	0000H
	FFFFF646H	Timer mode control register 2	TMC2		√	√		00H
	FFFFF647H	Timer mode control register 3	TMC3		√	√		
FFFFF650H		Timer counter 45	TM45	R			√	0000H
	FFFFF650H	Timer counter 4	TM4			√		00H
	FFFFF651H	Timer counter 5	TM5			√		
FFFFF652H		Compare register 45	CR45	R/W			√	0000H
	FFFFF652H	Compare register 4	CR4			√		00H
	FFFFF653H	Compare register 5	CR5			√		
FFFFF654H		Timer clock selection register 45	TCL45				√	0000H
	FFFFF654H	Timer clock selection register 4	TCL4			√		00H
	FFFFF655H	Timer clock selection register 5	TCL5			√		
FFFFF656H		Timer mode control register 45	TMC45				√	0000H
	FFFFF656H	Timer mode control register 4	TMC4		√	√		00H
	FFFFF657H	Timer mode control register 5	TMC5		√	√		
FFFFF6C0H		Oscillation stabilization time selection register	OSTS			√		04H
FFFFF6C1H		Watchdog timer clock selection register	WDCS			√		00H
FFFFF6C2H		Watchdog timer mode register	WDTM		√	√		
FFFFF6E0H		RTC operation control register	RTCC				√	808XH
	FFFFF6E0H	RTC operation control register 0	RTCC0		√	√		80H
	FFFFF6E1H	RTC operation control register 1	RTCC1		√	√		8XH
FFFFF6E2H		Sub-count register	SUBC	R			√	XXXXH
	FFFFF6E2H	Sub-count register L	SUBCL			√		XXH
	FFFFF6E3H	Sub-count register H	SUBCH			√		
FFFFF6E4H		Minute/second count register	SECMIN				√	XXXXH
	FFFFF6E4H	Second count register	SEC			√		XXH
	FFFFF6E5H	Minute count register	MIN			√		
FFFFF6E6H		Day/hour count register	HOURDAY				√	0XXXH
	FFFFF6E6H	Hour count register	HOUR			√		XXH
	FFFFF6E7H	Day count register	DAY			√		0XH
FFFFF6E8H		Week count register	WEEK				√	0XXXH
	FFFFF6E8H	Week count register L	WEEKL			√		XXH
	FFFFF6E9H	Week count register H	WEEKH			√		0XH
FFFFF6EAH		Minute/second count setting register	SECMINB	W			√	0000H
	FFFFF6EAH	Second count setting register	SECB			√		00H
	FFFFF6EBH	Minute count setting register	MINB			√		
FFFFF6ECH		Day/hour count setting register	HOURDAYB				√	0000H
	FFFFF6ECH	Hour count setting register	HOURB			√		00H
	FFFFF6EDH	Day count setting register	DAYB			√		

(6/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits				Default Value
				1	8	16	32	
FFFFF6EEH	Week count setting register	WEEKB	W			√		0000H
FFFFF6EEH	Week count setting register L	WEEKBL			√			00H
FFFFF6EFH	Week count setting register H	WEEKBH			√			
FFFFF802H	System status register	SYS	R/W	√	√			
FFFFF810H	DMA trigger factor register 0	DTFR0		√	√			00H
FFFFF812H	DMA trigger factor register 1	DTFR1		√	√			
FFFFF814H	DMA trigger factor register 2	DTFR2		√	√			
FFFFF816H	DMA trigger factor register 3	DTFR3		√	√			
FFFFF820H	Power save mode register	PSMR		√	√			
FFFFF828H	Processor clock control register	PCC		√	√			03H
FFFFF840H	Correction address register 0	CORAD0					√	00000000H
FFFFF840H	Correction address register 0L	CORAD0L				√		0000H
FFFFF842H	Correction address register 0H	CORAD0H				√		
FFFFF844H	Correction address register 1	CORAD1					√	00000000H
FFFFF844H	Correction address register 1L	CORAD1L				√		0000H
FFFFF846H	Correction address register 1H	CORAD1H				√		
FFFFF848H	Correction address register 2	CORAD2					√	00000000H
FFFFF848H	Correction address register 2L	CORAD2L				√		0000H
FFFFF84AH	Correction address register 2H	CORAD2H				√		
FFFFF84CH	Correction address register 3	CORAD3					√	00000000H
FFFFF84CH	Correction address register 3L	CORAD3L				√		0000H
FFFFF84EH	Correction address register 3H	CORAD3H				√		
FFFFF880H	Correction control register	CORCN		√	√			00H
FFFFF8B0H	Prescaler mode register	PRSM			√			
FFFFF8B1H	Prescaler compare register	PRSCM			√			
FFFFFA00H	UART0 operation mode register	ASIM0		√	√			01H
FFFFFA02H	Receive buffer register 0	RXB0	R		√			FFH
FFFFFA03H	UART0 reception error status register	ASIS0			√			00H
FFFFFA04H	Transmit buffer register 0	TXB0	R/W		√			FFH
FFFFFA05H	UART0 transmit status register	ASIF0	R	√	√			00H
FFFFFA06H	Clock select register 0	CKSR0	R/W		√			
FFFFFA07H	Baud rate generator compare register 0	BRGC0			√			FFH
FFFFFA10H	UART1 operation mode register	ASIM1		√	√			01H
FFFFFA12H	Receive buffer register 1	RXB1	R		√			FFH
FFFFFA13H	UART1 reception error status register	ASIS1			√			00H
FFFFFA14H	Transmit buffer register 1	TXB1	R/W		√			FFH
FFFFFA15H	UART1 transmit status register	ASIF1	R	√	√			00H
FFFFFA16H	Clock select register 1	CKSR1	R/W		√			
FFFFFA17H	Baud rate generator compare register 1	BRGC1			√			FFH

(7/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFC00H	External interrupt falling edge specification register 0	INTF0	R/W	√	√		00H
FFFFFC12H	External interrupt falling edge specification register 9	INTF9				√	0000H
FFFFFC12H	External interrupt falling edge specification register 9L	INTF9L		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		
FFFFFC32H	External interrupt rising edge specification register 9	INTR9				√	0000H
FFFFFC32H	External interrupt rising edge specification register 9L	INTR9L		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		
FFFFFC44H	Pull-up resistor option register 2Note	PU2		√	√		
FFFFFC46H	Pull-up resistor option register 3	PU3		√	√		
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		
FFFFFC64H	Port function register 2Note	PF2		√	√		
FFFFFC66H	Port function register 3	PF3		√	√		
FFFFFC68H	Port function register 4	PF4		√	√		
FFFFFC72H	Port function register 9	PF9				√	0000H
FFFFFC73H	Port function register 9H	PF9H		√	√		00H
FFFFFD00H	Clocked serial interface mode register 0	CSIM0		√	√		
FFFFFD01H	Clocked serial interface clock selection register 0	CSIC0			√		
FFFFFD02H	Serial I/O shift register 0	SIO0	R		√		
FFFFFD03H	Receive-only serial I/O shift register 0	SIOE0			√		
FFFFFD04H	Clocked serial interface transmit buffer register 0	SOTB0	R/W		√		
FFFFFD10H	Clocked serial interface mode register 1	CSIM1		√	√		
FFFFFD11H	Clocked serial interface clock selection register 1	CSIC1			√		
FFFFFD12H	Serial I/O shift register 1	SIO1	R		√		
FFFFFD13H	Receive-only serial I/O shift register 1	SIOE1			√		
FFFFFD14H	Clocked serial interface transmit buffer register 1	SOTB1	R/W		√		
FFFFFD20H	Clocked serial interface mode register 2	CSIM2		√	√		
FFFFFD21H	Clocked serial interface clock selection register 2	CSIC2			√		
FFFFFD22H	Serial I/O shift register 2	SIO2	R		√		
FFFFFD23H	Receive-only serial I/O shift register 2	SIOE2			√		
FFFFFD24H	Clocked serial interface transmit buffer register 2	SOTB2	R/W		√		
FFFFFD30H	Clocked serial interface mode register 3	CSIM3		√	√		
FFFFFD31H	Clocked serial interface clock selection register 3	CSIC3			√		
FFFFFD32H	Serial I/O shift register 3	SIO3	R		√		
FFFFFD33H	Receive-only serial I/O shift register 3	SIOE3			√		
FFFFFD34H	Clocked serial interface transmit buffer register 3	SOTB3	R/W		√		
FFFFFD40H	Clocked serial interface mode register 4Note	CSIM4		√	√		
FFFFFD41H	Clocked serial interface clock selection register 4Note	CSIC4			√		
FFFFFD42H	Serial I/O shift register 4Note	SIO4	R		√		
FFFFFD43H	Receive-only serial I/O shift register 4Note	SIOE4			√		

Note V850ES/SA3 only

(8/8)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFD44H	Clocked serial interface transmit buffer register 4[Note 1]	SOTB4	R/W		√		00H
FFFFFD80H	IIC shift register[Note 2]	IIC			√		
FFFFFD82H	IIC control register[Note 2]	IICC		√	√		
FFFFFD83H	Slave address register[Note 2]	SVA			√		
FFFFFD84H	IIC clock select register[Note 2]	IICCL		√	√		
FFFFFD85H	IIC function expansion register[Note 2]	IICX		√	√		
FFFFFD86H	IIC status register[Note 2]	IICS	R	√	√		
FFFFFFBEH	External bus interface mode control register	EXIMC	R/W	√	√		

Notes 1. V850ES/SA3 only

2. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

3.4.8 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up.

The V850ES/SA2 and V850ES/SA3 have the following three special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- Watchdog timer mode register (WDTM)

In addition, a command register (PRCDM) is provided to protect against a write access to the special registers so that the application system does not inadvertently stop due to a program hang-up. A write access to the special registers is made in a specific sequence, and an illegal store operation is reported to the system status register (SYS).

★ (1) Setting data to special registers

Set data to the special registers in the following sequence:

<1> Store the DMA transfer state.
<2> Disable DMA operation.
<3> Prepare data to be set to the special register in a general-purpose register.
<4> Write the data prepared in <3> to the command register (PRCMD).
<5> Write the setting data to the special register (by using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<6> Insert NOP instructions (5 instructions).
<7> Check if DMA transfer has ended between <1> and <2> above.
<8> If DMA transfer has not ended and DMA operation is required, enable DMA operation.

[Example] With PSC register

```
    ST.B   r11, PSMR[r0]          ; Set PSMR register.
<1> LD.B   DCHCn[r0], r12         ; Store DMA transfer state.
    ANDI   0xfe, r12, r13         ;
<2> ST.B   r13 , DCHCn [r0]       ; Disable DMA operation.
<3> MOV    0x02 , r10
<4> ST.B   r10 , PRCMD [r0]       ; Write PRCMD register.
<5> ST.B   r10 , PSC [r0]         ; Set PSC register.
<6> NOP                           ; Dummy instruction
    NOP                           ; Dummy instruction
    NOP                           ; Dummy instruction
    NOP                           ; Dummy instruction
    NOP                           ; Dummy instruction
<7> TST1   7, DCHCn[r0]           ; Check if DMA transfer has ended between <1> and <2>.
    BNE    next                   ;
<8> ST.B   r12 , DCHCn [r0]       ; Return DMA to the original state.
    next :
    (next instruction)
```

There is no special sequence to read a special register.

Cautions 1. When a store instruction is executed to store data in the command register, an interrupt is not acknowledged. This is because it is assumed that steps <4> and <5> above are performed by successive store instructions. If another instruction is placed between <4> and <5>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction.

2. Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<5> in Example) to write data to the PRCMD register (<4> in Example). The same applies when a general-purpose register is used for addressing.

Cautions 3. Five NOP instructions or more must be inserted immediately after setting the IDLE mode or software STOP mode (by setting the STP bit of the PSC register to 1). NOP instructions are not necessary in other cases.

4. Finish all DMA transfers before performing the above processing.

(2) Command register (PRCMD)

The command register (PRCMD) is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program hang-up. The first write access to a special register (power save control register (PSC)) is valid after data has been written in advance to the PRCMD register. In this way, the value of the special register can be rewritten only in a specific sequence, so as to protect the register from an illegal write access.

The PRCMD register is write-only, in 8-bit units (undefined data is read when this register is read).

After reset: Undefined W Address: FFFFF1FCH

	7	6	5	4	3	2	1	0
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) System status register (SYS)

Status flags that indicate the operation status of the overall system are allocated to this register.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF802H

	7	6	5	4	3	2	1	<0>
SYS	0	0	0	0	0	0	0	PRERR

PRERR	Detects protection error
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Set condition (PRERR = 1)

(i) When data is written to a special register without writing anything to the PRCMD register (when <5> is executed without executing <4> in **3.4.8 (1) Setting special register**)

(ii) When data is written to a peripheral I/O register other than a special register (including execution of a bit manipulation instruction) after writing data to the PRCMD register (if <5> in **3.4.8 (1) Setting special register** is not the setting of a special register)

Remark If a peripheral I/O register is read (excluding the bit manipulation instructions) or the internal RAM is accessed after the PRCMD register is written and before the special register is written, the PRERR flag is not set and setting data can be written to the special registers other than the WDTM register (i.e., the PCC and PSC registers).

(b) Clear condition (PRERR = 0)

(i) When 0 is written to the PRERR flag of the SYS register

(ii) When the system is reset

Cautions 1. If 0 is written to the PRERR bit of the SYS register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is cleared to 0 (the write access takes precedence).

2. If data is written to the PRCMD register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is set to 1.

3.4.9 Notes

Be sure to set the following register first when using the V850ES/SA2 and V850ES/SA3:

- System wait control register (VSWC)

After setting the VSWC register, set the other registers as necessary.
When using the external bus, initialize each register in the following order after setting the above register.

<1> Set each pin to the control mode by using the port-related registers.

(1) System wait control register (VSWC)

The system wait control register (VSWC) controls wait of bus access to the internal peripheral I/O registers.
Three clocks are required to access an internal peripheral I/O register (without a wait cycle). The V850ES/SA2 and V850ES/SA3 require wait cycles according to the operating frequency. Set the following value to the VSWC register in accordance with the frequency used.
The VSWC register can be read or written in 8-bit units (address: FFFFF06EH, default value: 77H).

Operating Frequency (f_{CLK})	Set Value of VSWC
2 MHz ≤ f_{CLK} < 16.6 MHz	00H
16.6 MHz ≤ f_{CLK} ≤ 20 MHz	01H

★ #### (2) Access to special internal peripheral I/O registers

When accessing the following registers, if the CPU accesses the register at the same time the register changes due to hardware processing, a wait operation is generated for the register access. In this case, it may take longer to access the internal peripheral I/O registers compared with ordinary access.

Peripheral Function	Target Register Name
DMA	DTFR0 to DTFR3
Timer n (n = 0, 1)	TMn, CCn0, CCn1, TMCn0
Watchdog timer	WDTM
UARTn (n = 0, 1)	ASISn
I^{2}C bus	IICS

Caution When the CPU operates on the subclock and main oscillation is stopped, access to a register for which a wait is generated is prohibited. If a wait is generated, only reset can release the wait.

CHAPTER 4 PORT FUNCTIONS

4.1 Features

4.1.1 V850ES/SA2

○ Input ports: 14 pins
○ I/O ports: 68 pins
○ I/O pins function alternately as other peripheral functions
○ Can be set to input or output mode in 1-bit units.

4.1.2 V850ES/SA3

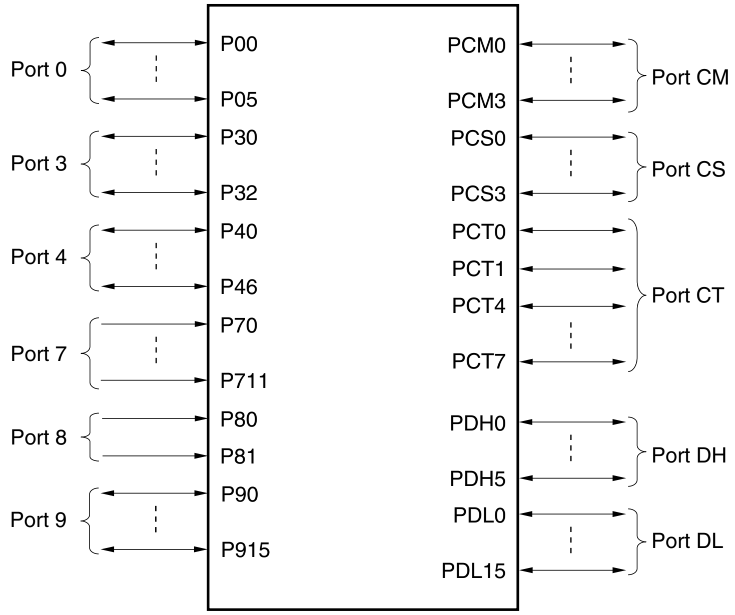
○ Input ports: 18 pins
○ I/O ports: 84 pins
○ I/O pins function alternately as other peripheral functions
○ Can be set to input or output mode in 1-bit units.

4.2 Basic Configuration of Port

4.2.1 V850ES/SA2

The V850ES/SA2 has a total of 82 input/output port pins (of which 14 are input-only port pins): ports 0, 3, 4, 7 to 9, CM, CS, CT, DH, and DL. The port configuration is shown below.

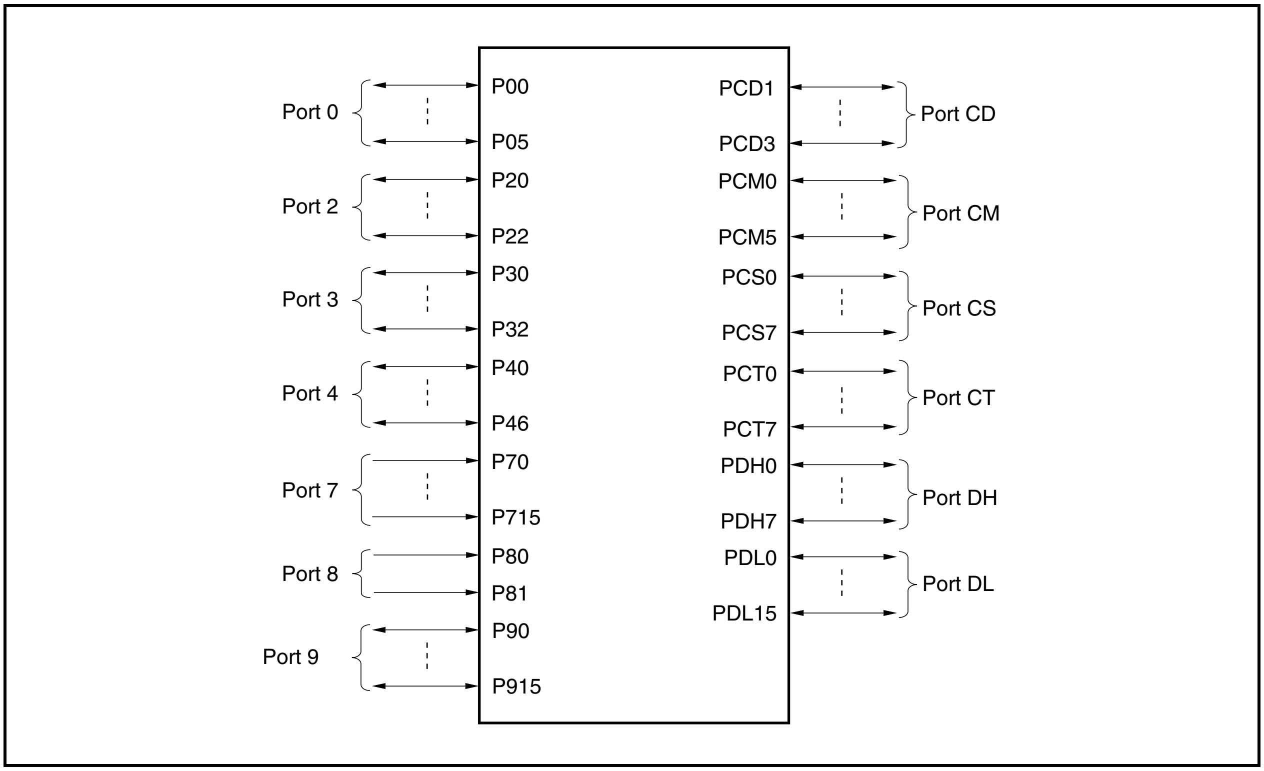
Figure 4-1. Port Configuration (V850ES/SA2)



4.2.2 V850ES/SA3

The V850ES/SA3 has a total of 102 input/output port pins (of which 18 are input-only port pins): ports 0, 2 to 4, 7 to 9, CD, CM, CS, CT, DH, and DL. The port configuration is shown below.

Figure 4-2. Port Configuration (V850ES/SA3)



4.3 Port Configuration

Table 4-1. Port Configuration (V850ES/SA2)

Item	Configuration
Control registers	Port mode registers (PMn: n = 0, 3, 4, 9, CD, CM, CS, CT, DH, DL) Port mode control registers (PMCn: n = 0, 3, 4, 9, CM, CS, CT, DH, DL) Port function control registers (PFCn: n = 3, 4, 9) Port function registers (PFn: n= 3, 4, 9) Pull-up resistor option registers (PUn: n = 0, 3, 4, 9) External interrupt rising edge specification registers (INTRn: n = 0, 9) External interrupt falling edge specification registers (INTFn: n = 0, 9)
Ports	I/O: 68 pins, Input: 14 pins
Pull-up resistor	Software-controlled: 32 resistors

Table 4-2. Port Configuration (V850ES/SA3)

Item	Configuration
Control registers	Port mode registers (PMn: n = 0, 2 to 4, 9, CD, CM ,CS, CT, DH, DL) Port mode control registers (PMCn: n = 0, 2 to 4, 9, CM, CS, CT, DH, DL) Port function control registers (PFCn: n = 3, 4, 9) Port function registers (PFn: n = 2, 3, 4, 9) Pull-up resistor option registers (PUn: n = 0, 2, 3, 4, 9) External interrupt rising edge specification registers (INTRn: n = 0, 9) External interrupt falling edge specification registers (INTFn: n = 0, 9)
Ports	I/O: 84 pins, Input: 18 pins
Pull-up resistor	Software-controlled: 35 resistors

4.3.1 Port 0

Port 0 can be set to the input or output mode in 1-bit units.

The number of I/O port bits of each product is the same.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	6-bit I/O port
V850ES/SA3	6-bit I/O port

(1) Function of port 0

○ Input/output data can be specified in 1-bit units by using port register 0 (P0).

○ Can be set to the input or output mode in 1-bit units by using port mode register 0 (PM0).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register 0 (PMC0).

○ An internal pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 0 (PU0).

○ The valid edge of the external interrupt (alternate function) can be set in 1-bit units by using external interrupt falling edge specification register 0 (INTF0) and external interrupt rising edge specification register 0 (INTR0).

Port 0 has an alternate function as the following pins

Table 4-3. Alternate-Function Pins of Port 0

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark
Port 0	P00	NMI	I/O	Provided	–
	P01	INTP0/TI2			
	P02	INTP1/TI3			
	P03	INTP2/TI4			
	P04	INTP3/TI5			
	P05	INTP4			

Note Software pull-up function

(2) Registers

(a) Port register 0 (P0)

Port register 0 (P0) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	0	P05	P04	P03	P02	P01	P00

P0n	Controls output data (in output mode) (n = 0 to 5)
0	Outputs 0.
1	Outputs 1.

Remarks 1. In input mode: When port 0 (P0) is read, the pin level at that time is read. When written, the data written to P0 is written. The input pin is not affected.

In output mode: When port 0 (P0) is read, the value of P0 is read. When a value is written to P0, it is immediately output.

2. After reset, an undefined value (pin input level) is read from P0 in the input mode. When P0 is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register 0 (PM0)

This is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	1	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	Controls input/output mode (n = 0 to 5)
0	Output mode
1	Input mode

(c) Port mode control register 0 (PMC0)

This is an 8-bit register that specifies the port mode or control mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	0	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00

PMC05	Specifies operation mode of P05 pin
0	I/O port
1	INTP4 input

PMC04	Specifies operation mode of P04 pin
0	I/O port
1	INTP3/TI5 input

PMC03	Specifies operation mode of P03 pin
0	I/O port
1	INTP2/TI4 input

PMC02	Specifies operation mode of P02 pin
0	I/O port
1	INTP1/TI3 input

PMC01	Specifies operation mode of P01 pin
0	I/O port
1	INTP0/TI2 input

PMC00	Specifies operation mode of P00 pin
0	I/O port
1	NMI input

Caution A register for selects external interrupts (INTP0 to INTP3) and timer inputs (TI2 to TI5) is not provided.

When using port 0 to input an external interrupt, specify the valid edge of the interrupt request by using the INTR0/INTF0 register. When using the port for timer input, specify the valid edge of TIn by using the TCLn register.

- **INTR0: External interrupt rising edge specification register 0 (Refer to 4.3.1 (2) (f).)**
- **INTF0: External interrupt falling edge specification register 0 (Refer to 4.3.1 (2) (e).)**
- **TCLn: Timer n clock select register (refer to CHAPTER 8 8-BIT TIEMR/EVENT COUNTERS 2 TO 5)**

(d) Pull-up resistor option register 0 (PU0)

This is an 8-bit register that specifies connection of an internal pull-up resistor.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	0	0	PU05	PU04	PU03	PU02	PU01	PU00

PU0n	Controls connection of internal pull-up resistor (n = 0 to 5)
0	Not connected
1	Connected

(e) External interrupt falling edge specification register 0 (INTF0)

This 8-bit register specifies detection of the falling edge of the external interrupt pins.

It can be read or written in 8-bit or 1-bit units.

Caution Set the port mode after clearing the INTF0n and INTR0n bits to 0 when switching from the external interrupt function (alternate function) to the port function because an edge may be detected.

After reset: 00H R/W Address: FFFFFC00H

	7	6	5	4	3	2	1	0
INTF0	0	0	INTF05	INTF04	INTF03	INTF02	INTR01	INTR00
			INTP4	INTP3	INTP2	INTP1	INTP0	NMI

Remark For how to specify a valid edge, refer to **Table 4-4**.

(f) External interrupt rising edge specification register 0 (INTR0)

This 8-bit register specifies detection of the rising edge of the external interrupt pins.

It can be read or written in 8-bit or 1-bit units.

Caution Set the port mode after clearing the INTF0n and INTR0n bits to 0 when switching from the external interrupt function (alternate function) to the port function because an edge may be detected.

After reset: 00H   R/W   Address: FFFFFC20H

	7	6	5	4	3	2	1	0
INTR0	0	0	INTR05	INTR04	INTR03	INTR02	INTR01	INTR00
			INTP4	INTP3	INTP2	INTP1	INTP0	NMI

Remark For how to specify a valid edge, refer to **Table 4-4**.

Table 4-4. Specifying Valid Edge

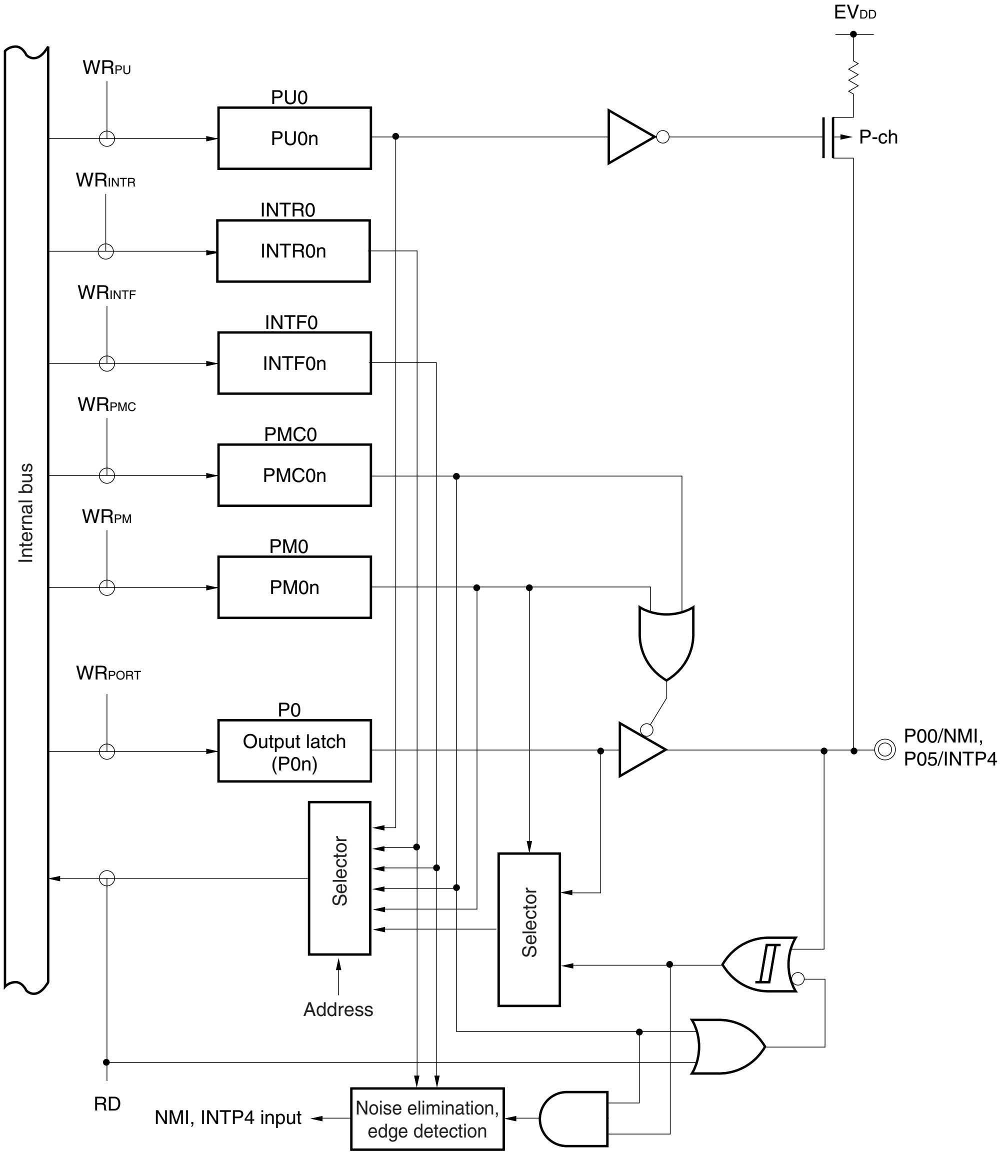
INTF0n	INTR0n	Specifies valid edge (n = 0 to 5)
0	0	Detects no edge.
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 0: Control of NMI pin

n = 1 to 5: Control of INTP0 to INTP4 pins

(3) Block diagram

Figure 4-3. Block Diagram of P00 and P05



Caution These pins do not have hysteresis characteristics in the port mode. They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P0: Port register 0

PM0: Port mode register 0

PMC0: Port mode control register 0

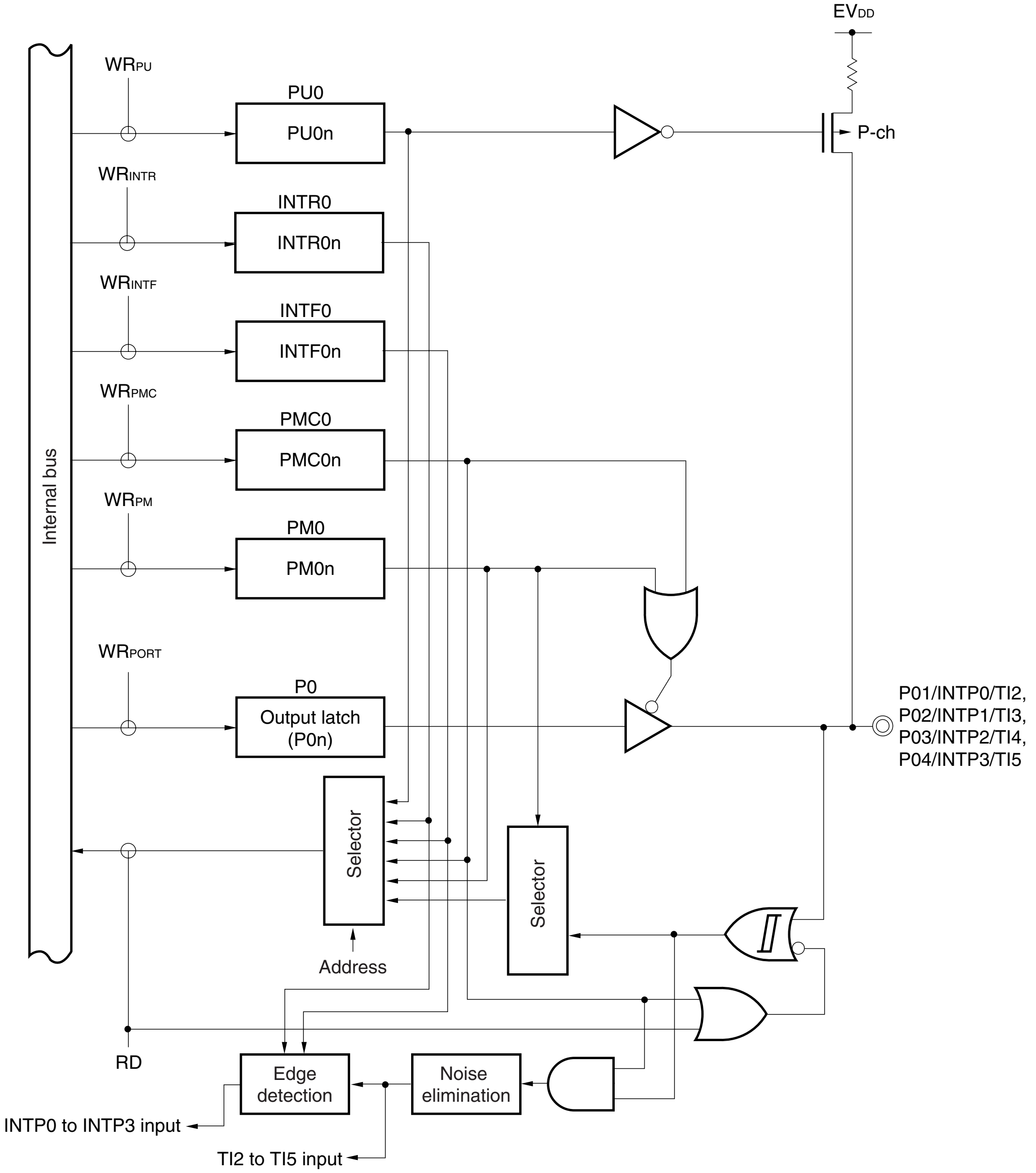
PU0: Pull-up resistor option register 0

INTR0: External interrupt rising edge specification register 0

INTF0: External interrupt falling edge specification register 0

2. n = 0, 5

Figure 4-4. Block Diagram of P01 to P04



Caution These pins do not have hysteresis characteristics in the port mode.
They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P0: Port register 0

PM0: Port mode register 0

PMC0: Port mode control register 0

PU0: Pull-up resistor option register 0

INTR0: External interrupt rising edge specification register 0

INTF0: External interrupt falling edge specification register 0

2. n = 1 to 4

4.3.2 Port 2

Port 2 can be set to the input or output mode in 1-bit units.

The number of I/O port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	–
V850ES/SA3	3-bit I/O port

(1) Functions of port 2 (V850ES/SA3)

○ Input/output data can be specified in 1-bit units by using port register 2 (P2).

○ Can be set to the input or output mode in 1-bit units by using port mode register 2 (PM2).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register 2 (PMC2).

○ N-ch open-drain output can be set in 1-bit units by using port function register 2 (PF2).

○ An internal pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 2 (PU2).

Port 2 has an alternate function as the following pins.

Table 4-5. Alternate-Function Pins of Port 2 (V850ES/SA3)

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark
Port 2	P20	SI4	I/O	Provided	–
	P21	SO4			N-ch open-drain output selectable
	P22	$\overline{\text{SCK4}}$			

Note Software pull-up function

(2) Registers

(a) Port register 2 (P2)

Port register 2 (P2) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF404H

	7	6	5	4	3	2	1	0
P2	0	0	0	0	0	P22	P21	P20

P2n	Controls output data (in output mode) (n = 0 to 2)
0	Outputs 0.
1	Outputs 1.

Remarks 1. In input mode: When port 2 (P2) is read, the pin level at that time is read. When written, the data written to P2 is written. The input pin is not affected.

In output mode: When port 2 (P2) is read, the value of P2 is read. When a value is written to P2, it is immediately output.

2. After reset, an undefined value (pin input level) is read from P2 in the input mode. When P2 is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register 2 (PM2)

This is an 8-bit register that specifies the input or output mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF424H

	7	6	5	4	3	2	1	0
PM2	1	1	1	1	1	PM22	PM21	PM20

PM2n	Controls input/output mode (n = 0 to 2)
0	Output mode
1	Input mode

(c) Port mode control register 2 (PMC2)

This is an 8-bit register that specifies the port mode or control mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	0	PMC22	PMC21	PMC20

PMC22	Specifies operation mode of P22 pin
0	I/O port
1	$\overline{\text{SCK4}}$ I/O

PMC21	Specifies operation mode of P21 pin
0	I/O port
1	SO4 output

PMC20	Specifies operation mode of P20 pin
0	I/O port
1	SI4 input

(d) Port function register 2 (PF2)

This 8-bit register specifies normal output or N-ch open-drain output.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC64H

	7	6	5	4	3	2	1	0
PF2	0	0	0	0	0	PF22	PF21	0

PF2n	Controls normal output or N-ch open-drain output (n = 1, 2)
0	Normal output
1	N-ch open-drain output

Cautions 1. The N-ch open-drain output voltage is the normal voltage, not the medium voltage.

2. PF2n = 1 is enabled only in the following cases. Otherwise, the setting is prohibited.

n = 1: SO4

n = 2: $\overline{\text{SCK4}}$

(e) Pull-up resistor option register 2 (PU2)

This is an 8-bit register that specifies connection of an internal pull-up resistor.

This register can be read or written in 8-bit or 1-bit units.

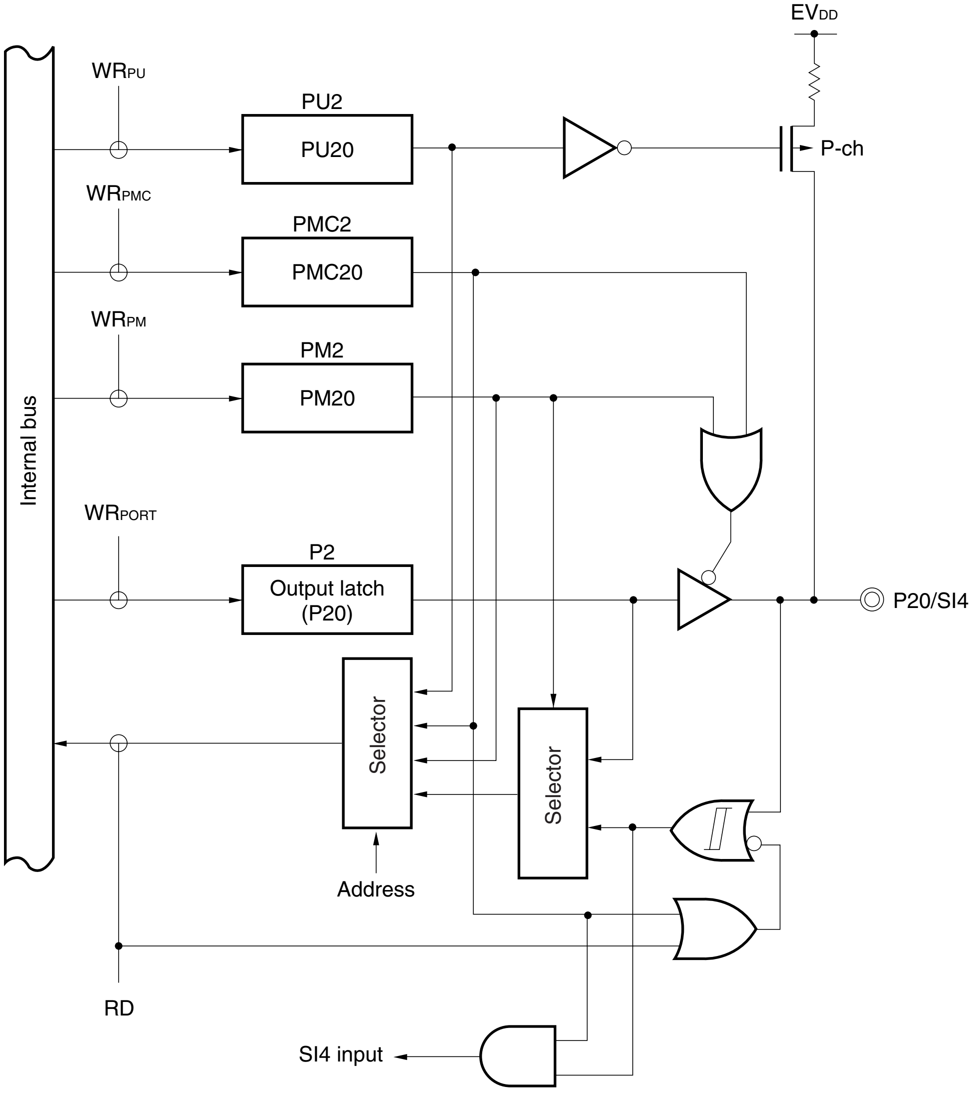
After reset: 00H R/W Address: FFFFFC44H

	7	6	5	4	3	2	1	0
PU2	0	0	0	0	0	PU22	PU21	PU20

PU2n	Controls connection of internal pull-up resistor (n = 0 to 2)
0	Not connected
1	Connected

(3) Block diagram

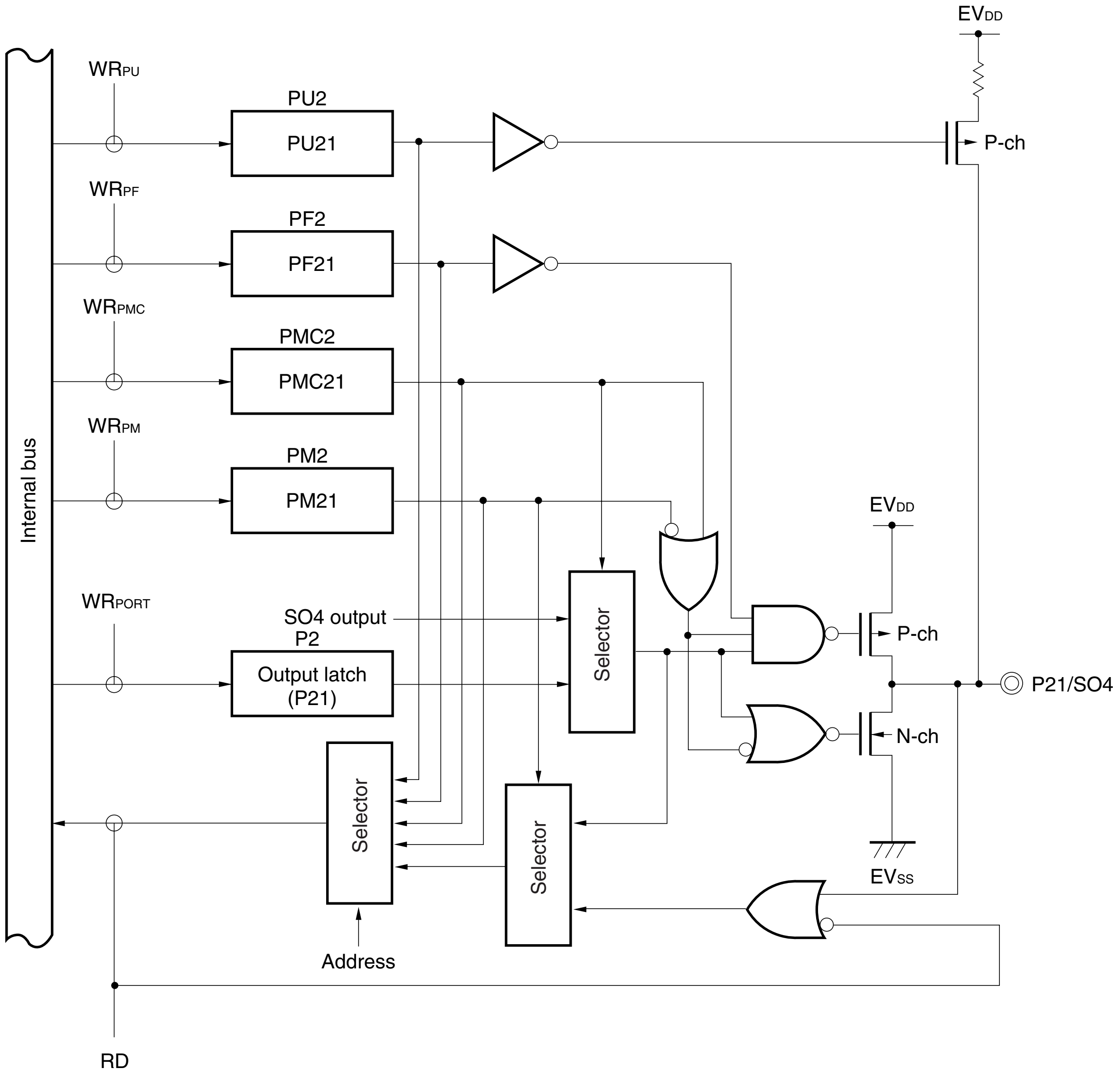
Figure 4-5. Block Diagram of P20



Caution This pin does not have hysteresis characteristics in the port mode.
It has hysteresis characteristics only when an input-pin alternate function is used.

Remark		
	P2:	Port register 2
	PM2:	Port mode register 2
	PMC2:	Port mode control register 2
	PU2:	Pull-up resistor option register 2

Figure 4-6. Block Diagram of P21



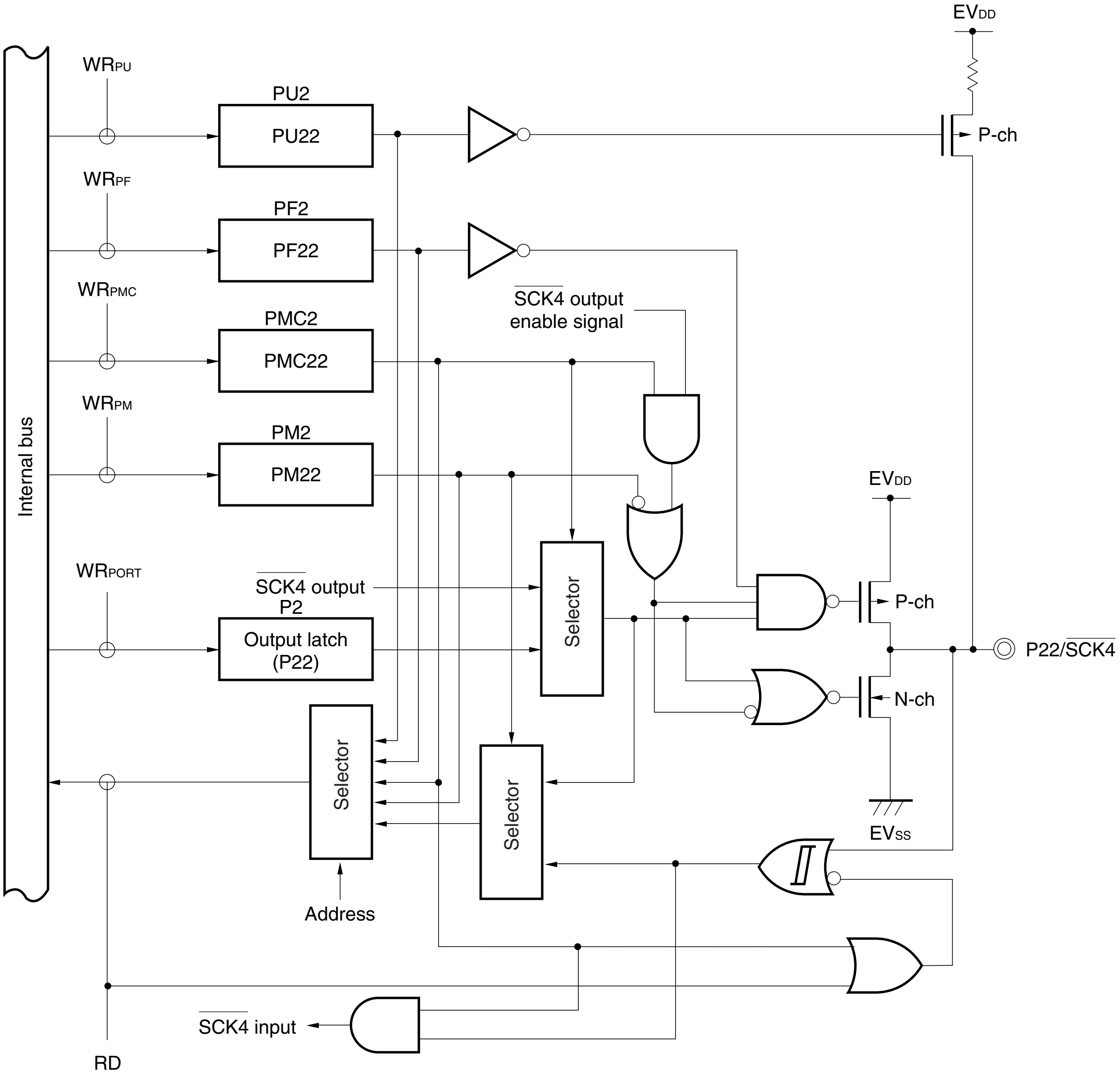
Remark P2: Port register 2

PM2: Port mode register 2

PMC2: Port mode control register 2

PU2: Pull-up resistor option register 2

Figure 4-7. Block Diagram of P22



Caution **This pin does not have hysteresis characteristics in the port mode. It has hysteresis characteristics only when an input-pin alternate function is used.**

Remark

P2:	Port register 2
PM2:	Port mode register 2
PMC2:	Port mode control register 2
PF2:	Port function register 2
PU2:	Pull-up resistor option register 2

4.3.3 Port 3

Port 3 can be set to the input or output mode in 1-bit units.

The number of I/O port bits of each product is the same.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	3-bit I/O port
V850ES/SA3	3-bit I/O port

(1) Functions of port 3

○ Input/output data can be specified in 1-bit units by using port register 3 (P3).

○ Can be set to the input or output mode in 1-bit units by using port mode register 3 (PM3).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register 3 (PMC3).

○ Control mode 1 or control mode 2 can be specified in 1-bit units by using port function control register 3 (PFC3).

○ N-ch open-drain output can be set in 1-bit units by using port function register 3 (PF3).

○ An internal pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 3 (PU3).

Port 3 has an alternate function as the following pins.

Table 4-6. Alternate-Function Pins of Port 3

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark
Port 3	P30	SI1/RXD0	I/O	Provided	–
	P31	SO1/TXD0			N-ch open-drain output selectable
	P32	$\overline{\text{SCK0}}$			

Note Software pull-up function

(2) Registers

(a) Port register 3 (P3)

Port register 3 (P3) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF406H

	7	6	5	4	3	2	1	0
P3	0	0	0	0	0	P32	P31	P30

P3n	Controls output data (in output mode) (n = 0 to 2)
0	Outputs 0.
1	Outputs 1.

Remarks 1. In input mode: When port 3 (P3) is read, the pin level at that time is read. When written, the data written to P3 is written. The input pin is not affected.

In output mode: When port 3 (P3) is read, the value of P3 is read. When a value is written to P3, it is immediately output.

2. After reset, an undefined value (pin input level) is read from P3 in the input mode. When P3 is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register 3 (PM3)

This is an 8-bit register that specifies the input or output mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF426H

	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

PM3n	Controls input/output mode (n = 0 to 2)
0	Output mode
1	Input mode

(c) Port mode control register 3 (PMC3)

This is an 8-bit register that specifies the port mode or control mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	0	0	0	0	0	PMC32	PMC31	PMC30

PMC32	Specifies operation mode of P32 pin
0	I/O port
1	$\overline{\text{SCK1}}$ I/O

PMC31	Specifies operation mode of P31 pin
0	I/O port
1	SO1/TXD0 output

PMC30	Specifies operation mode of P30 pin
0	I/O port
1	SI1/RXD0 input

(d) Port function control register 3 (PFC3)

This 8-bit register specifies control mode 1 or control mode 2.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	0	0	0	0	PFC31	PFC30

PFC31	Specifies operation mode of P31 pin in control mode
0	SO1 output
1	TXD0 output

PFC30	Specifies operation mode of P30 pin in control mode
0	SI1 input
1	RXD0 input

(e) Port function register 3 (PF3)

This 8-bit register specifies normal output or N-ch open-drain output.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC66H

	7	6	5	4	3	2	1	0
PF3	0	0	0	0	0	PF32	PF31	0

PF3n	Controls normal output or N-ch open-drain output (n = 1, 2)
0	Normal output
1	N-ch open-drain output

Cautions 1. The N-ch open-drain output voltage is the normal voltage, not the medium voltage.

2. PF3n = 1 is enabled only in the following cases. Otherwise, the setting is prohibited.

n = 1: SO1

n = 2: $\overline{\text{SCK1}}$

(f) Pull-up resistor option register 3 (PU3)

This is an 8-bit register that specifies connection of an internal pull-up resistor.

This register can be read or written in 8-bit or 1-bit units.

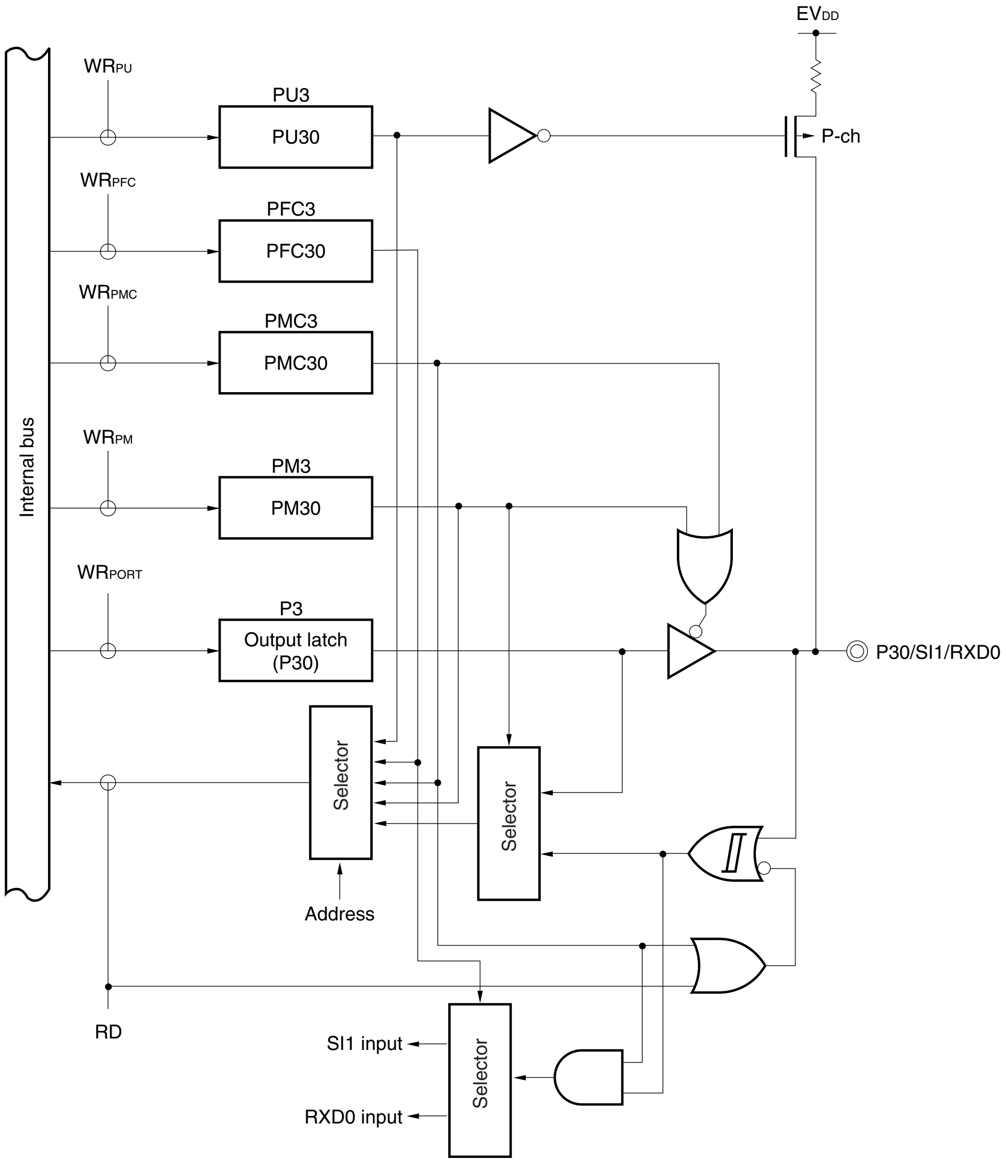
After reset: 00H R/W Address: FFFFFC46H

	7	6	5	4	3	2	1	0
PU3	0	0	0	0	0	PU32	PU31	PU30

PU3n	Controls connection of internal pull-up resistor (n = 0 to 2)
0	Not connected.
1	Connected.

(3) Block diagram

Figure 4-8. Block Diagram of P30

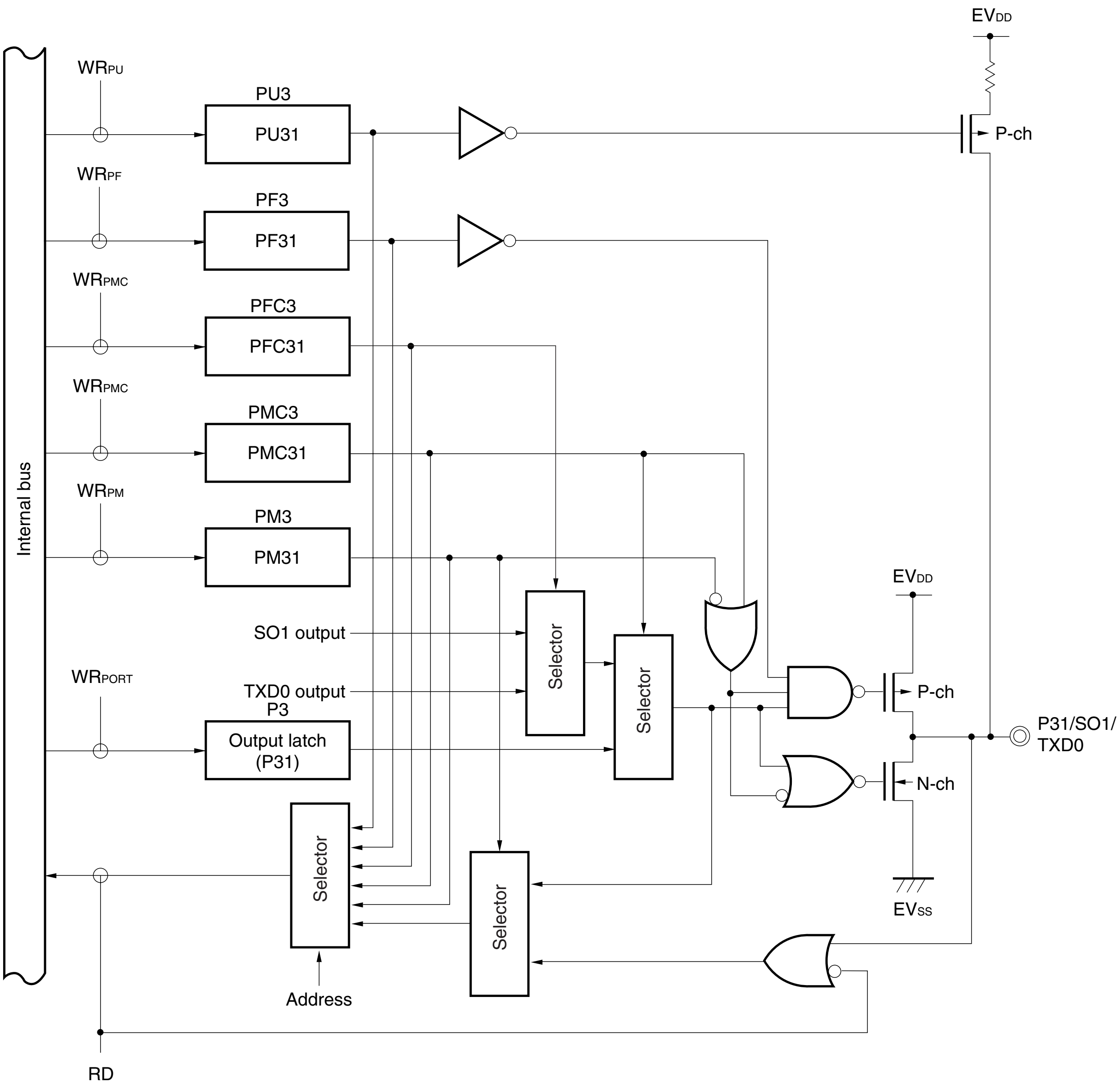


Caution **This pin does not have hysteresis characteristics in the port mode.**
It has hysteresis characteristics only when an input-pin alternate function is used.

Remark

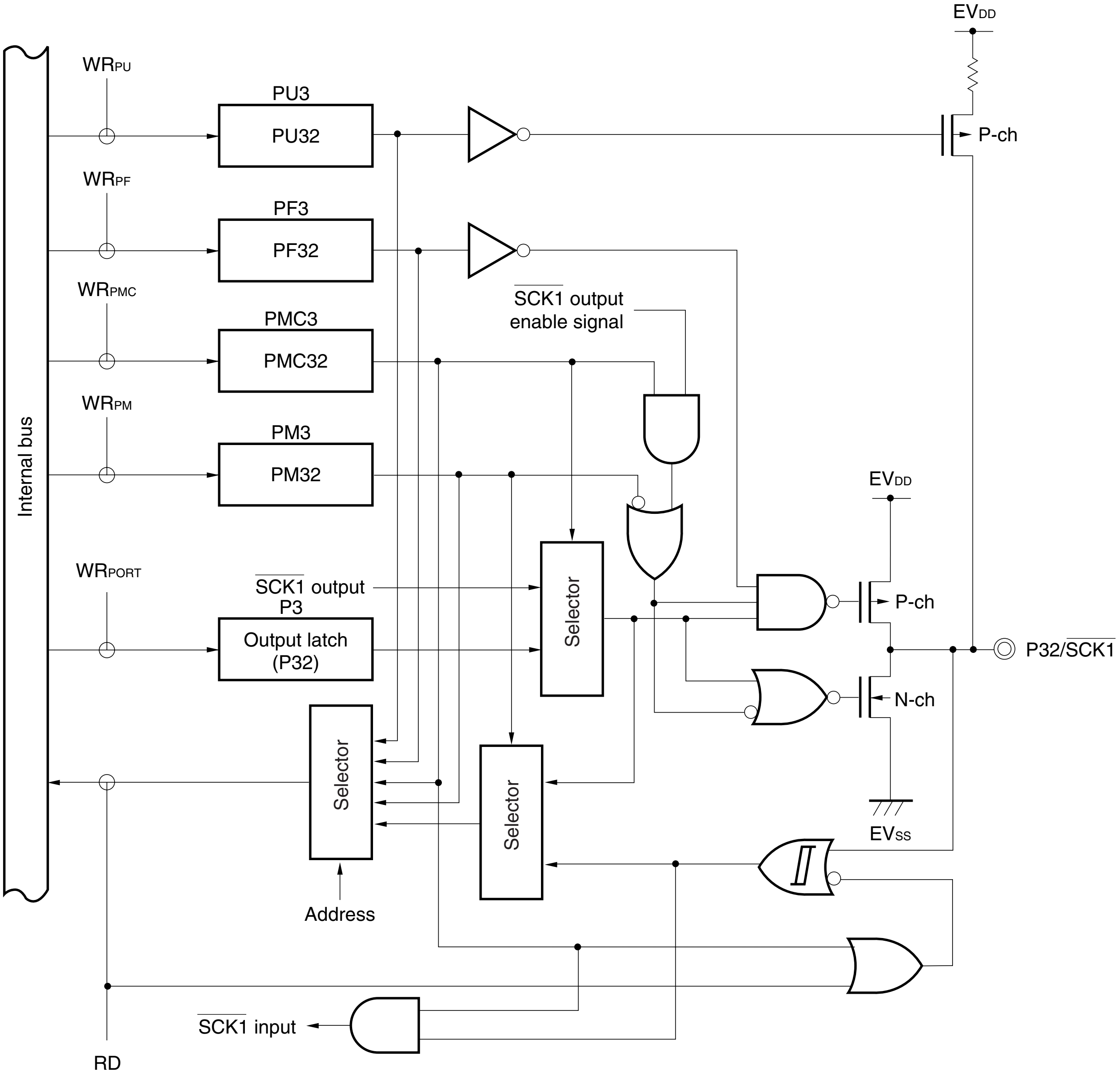
P3:	Port register 3
PM3:	Port mode register 3
PMC3:	Port mode control register 3
PFC3:	Port function control register 3
PU3:	Pull-up resistor option register 3

Figure 4-9. Block Diagram of P31



Remark P3: Port register 3
PM3: Port mode register 3
PMC3: Port mode control register 3
PFC3: Port function control register 3
PF3: Port function register 3
PU3: Pull-up resistor option register 3

Figure 4-10. Block Diagram of P32



Caution This pin does not have hysteresis characteristics in the port mode.
It has hysteresis characteristics only when an input-pin alternate function is used.

Remark		
	P3:	Port register 3
	PM3:	Port mode register 3
	PMC3:	Port mode control register 3
	PFC3:	Port function control register 3
	PU3:	Pull-up resistor option register 3

4.3.4 Port 4

Port 4 can be set to the input or output mode in 1-bit units.

The number of I/O port bits each product is the same.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	7-bit I/O port
V850ES/SA3	7-bit I/O port

(1) Functions of port 4

○ Input/output data can be specified in 1-bit units by using port register 4 (P4).

○ Can be set to the input or output mode in 1-bit units by using port mode register 4 (PM4).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register 4 (PMC4).

○ Control mode 1 or control mode 2 can be specified in 1-bit units by using port function control register 4 (PFC4).

○ N-ch open-drain output can be set to 1-bit units by using port function register 4 (PF4).

○ The internal pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 4 (PU4).

Port 4 has an alternate function as the following pins.

Table 4-7. Alternate-Function Pins of Port 4

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port 4	P40	SI0	I/O	Provided	–
	P41	SO0/SDA[Note 2]			
	P42	$\overline{\text{SCK0}}$/SCL[Note 2]			
	P43	INTP00/TI0/TCLR0			
	P44	INTP01/TO0			
	P45	INTP01/TI1/TCLR1			
	P46	INTP11/TO1			

Notes 1. Software pull-up function

2. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

(2) Registers

(a) Port register 4 (P4)

Port register 4 (P4) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF408H

	7	6	5	4	3	2	1	0
P4	0	P46	P45	P44	P43	P42	P41	P40

P4n	Controls output data (in output mode) (n = 0 to 6)
0	Outputs 0.
1	Outputs 1.

Remarks 1. In input mode: When port 4 (P4) is read, the pin level at that time is read. When written, the data written to P4 is written. The input pin is not affected.

In output mode: When port 4 (P4) is read, the value of P4 is read. When a value is written to P4, it is immediately output.

2. After reset, an undefined value (pin input level) is read from P4 in the input mode. When P4 is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register 4 (PM4)

This is an 8-bit register that specifies the input or output mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF428H

	7	6	5	4	3	2	1	0
PM4	1	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	Controls input/output mode (n = 0 to 6)
0	Output mode
1	Input mode

(c) Port mode control register 4 (PMC4)

This is an 8-bit register that specifies the port mode or control mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

PMC46	Specifies operation mode of P46 pin
0	I/O port
1	INTP11/TO1 I/O

PMC45	Specifies operation mode of P45 pin
0	I/O port
1	INTP10/TI1/TCLR1 input

PMC44	Specifies operation mode of P44 pin
0	I/O port
1	INTP01/TO0 I/O

PMC43	Specifies operation mode of P43 pin
0	I/O port
1	INTP00/TI0/TCLR0 input

PMC42	Specifies operation mode of P42 pin
0	I/O port
1	$\overline{\text{SCK0}}$/SCL[Note] I/O

PMC41	Specifies operation mode of P41 pin
0	I/O port
1	SO0 output/SDA[Note] I/O

PMC40	Specifies operation mode of P40 pin
0	I/O port
1	SI0 input

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Cautions 1. To use INTP0n, perform the following setting:

- **CMSn0 bit of TMCn1 register = 0**
- **ETIn bit of TMCn1 register = 1**
- **Setting of valid edge by SESn register**

2. To use TIn, perform the following setting:

- **CMSn0 bit of TMCn1 register = 1**
- **ETIn bit of TMCn1 register = 1**
- **Setting of valid edge by SESn register**

3. To use TCLRn, perform the following setting:

- **ECLRn bit of TMCn1 register = 1**
- **Setting of valid edge by SESn register**

Remark n = 0 or 1

(d) Port function control register 4 (PFC4)

This 8-bit register specifies control mode 1 or control mode 2.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	PFC46	0	PFC44	0	PFC42	PFC41	0

PFC46	Specifies operation mode of P46 pin in control mode
0	INTP11 input
1	TO1 output[Note 1]

PFC44	Specifies operation mode of P44 pin in control mode
0	INTP01 input
1	TO0 output[Note 1]

PFC42	Specifies operation mode of P42 pin in control mode
0	$\overline{\text{SCK0}}$ I/O
1	SCL I/O[Note 2]

PFC41	Specifies operation mode of P41 pin in control mode
0	SO0 output
1	SDA I/O[Note 2]

Notes 1. Setting of PCF44 and PCF46 = 1 is enabled only when TOn output is enabled (ENTOn of TMCn1 register = 1: n = 0 or 1). Otherwise, this setting is prohibited.

2. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Caution The PFC4n bit is valid only when the PMC4n bit = 1 (n = 1, 2, 4, or 6).

(e) Port function register 4 (PF4)

This 8-bit register specifies normal output or N-ch open-drain output.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFFC68H

	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	PF42	PF41	0

PF4n	Controls normal output or N-ch open-drain output (n = 1, 2)
0	Normal output
1	N-ch open-drain output

Cautions 1. The N-ch open-drain output voltage is the normal voltage, not the medium voltage.

2. PF4n = 1 is enabled only in the following cases. Otherwise, the setting is prohibited.
n = 1: SO0, SDA
n = 2: $\overline{\text{SCK0}}$, SCL

3. Be sure to set N-ch open-drain output when using I^2C (the SDA and SCL pins).

4. Follow the procedure below to set N-ch open drain.
<1> Set the P4n bit to 1.
<2> Set the PF4n bit to 1.
<3> Set the CSIE0 bit of the CSIM0 register or the IICE bit of the IICC register to 1.
<4> Set the PFC4n bit.
<5> Set the PMC4n bit to 1.

(f) Pull-up resistor option register 4 (PU4)

This is an 8-bit register that specifies connection of an internal pull-up resistor.

This register can be read or written in 8-bit or 1-bit units.

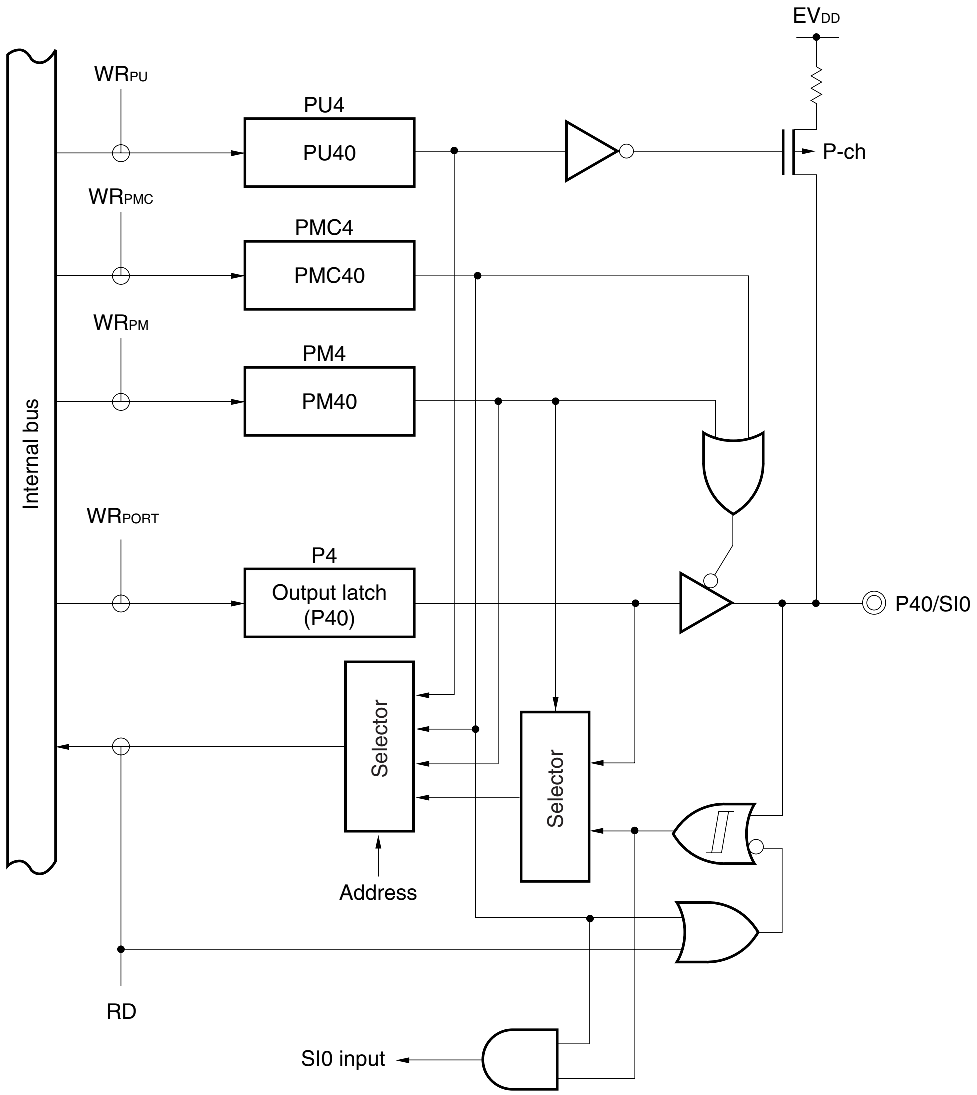
After reset: 00H R/W Address: FFFFFC48H

	7	6	5	4	3	2	1	0
PU4	0	PU46	PU45	PU44	PU43	PU42	PU41	PU40

PU4n	Controls connection of internal pull-up resistor (n = 0 to 6)
0	Not connected.
1	Connected.

(3) Block diagram

Figure 4-11. Block Diagram of P40



Caution This pin does not have hysteresis characteristics in the port mode.
It has hysteresis characteristics only when an input-pin alternate function is used.

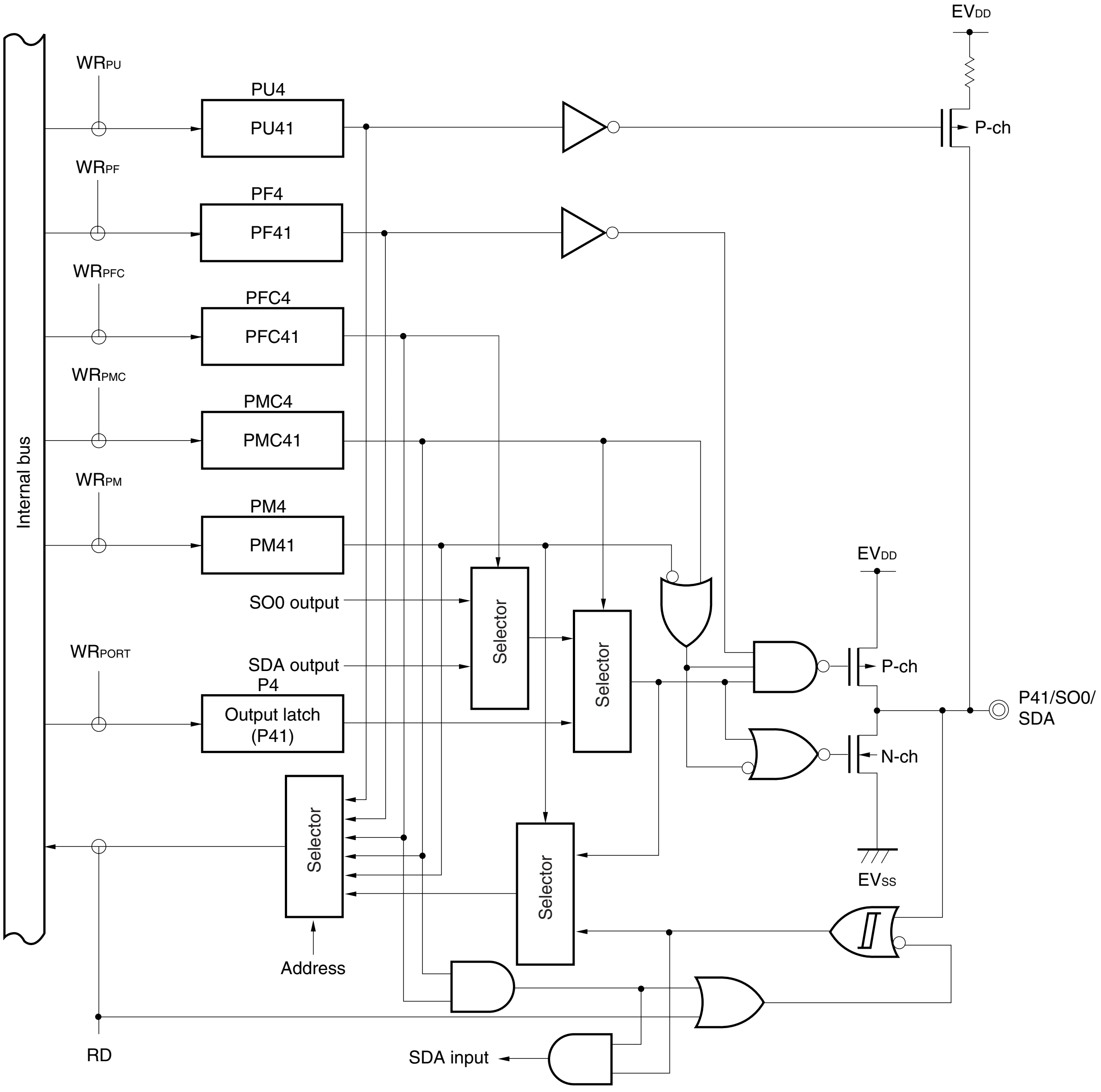
Remark P4: Port register 4

PM4: Port mode register 4

PMC4: Port mode control register 4

PU4: Pull-up resistor option register 4

Figure 4-12. Block Diagram of P41

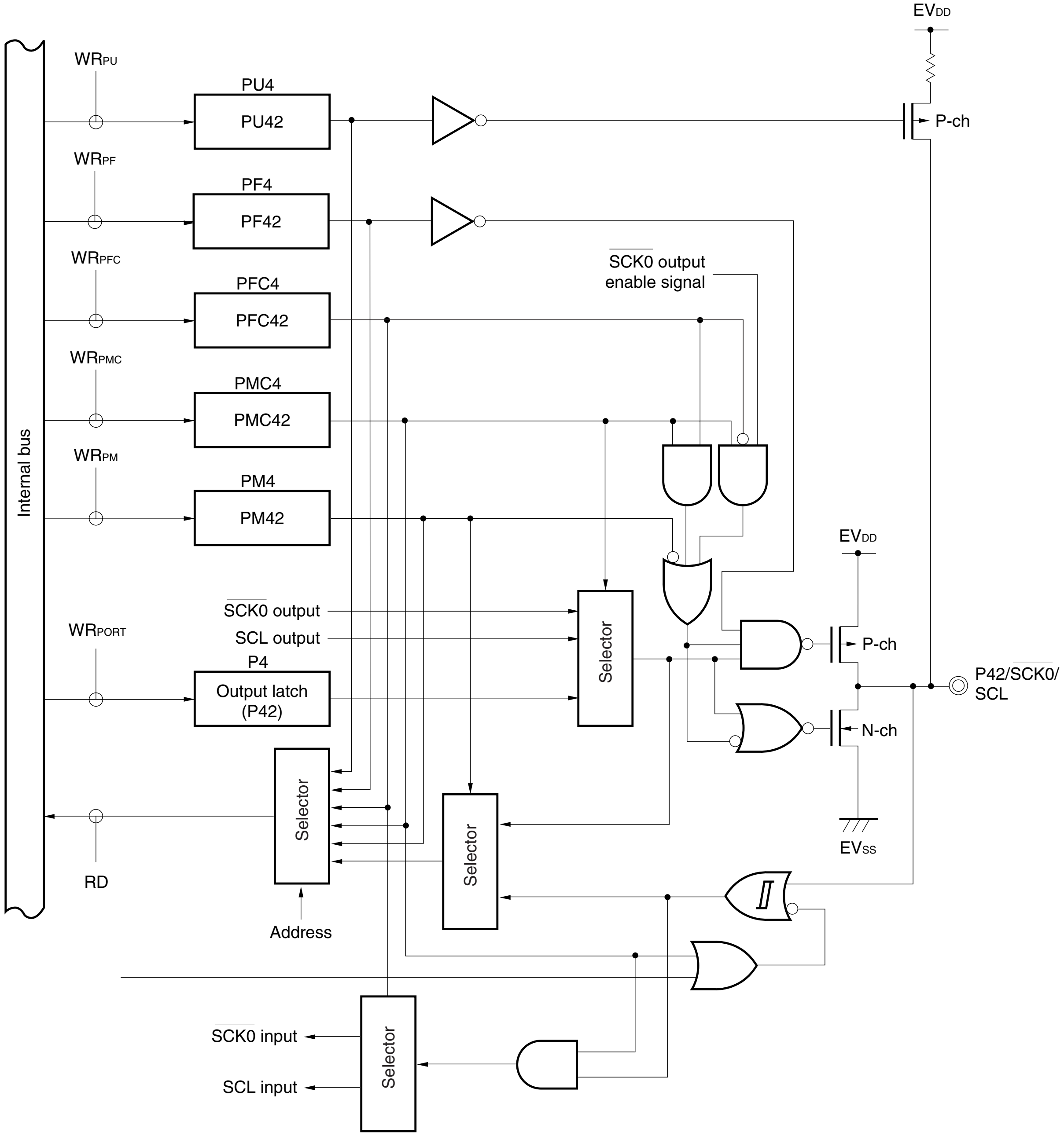


Caution **This pin does not have hysteresis characteristics in the port mode.**
It has hysteresis characteristics only when an input-pin alternate function is used.

Remark

P4:	Port register 4
PM4:	Port mode register 4
PMC4:	Port mode control register 4
PFC4:	Port function control register 4
PF4:	Port function register 4
PU4:	Pull-up resistor option register 4

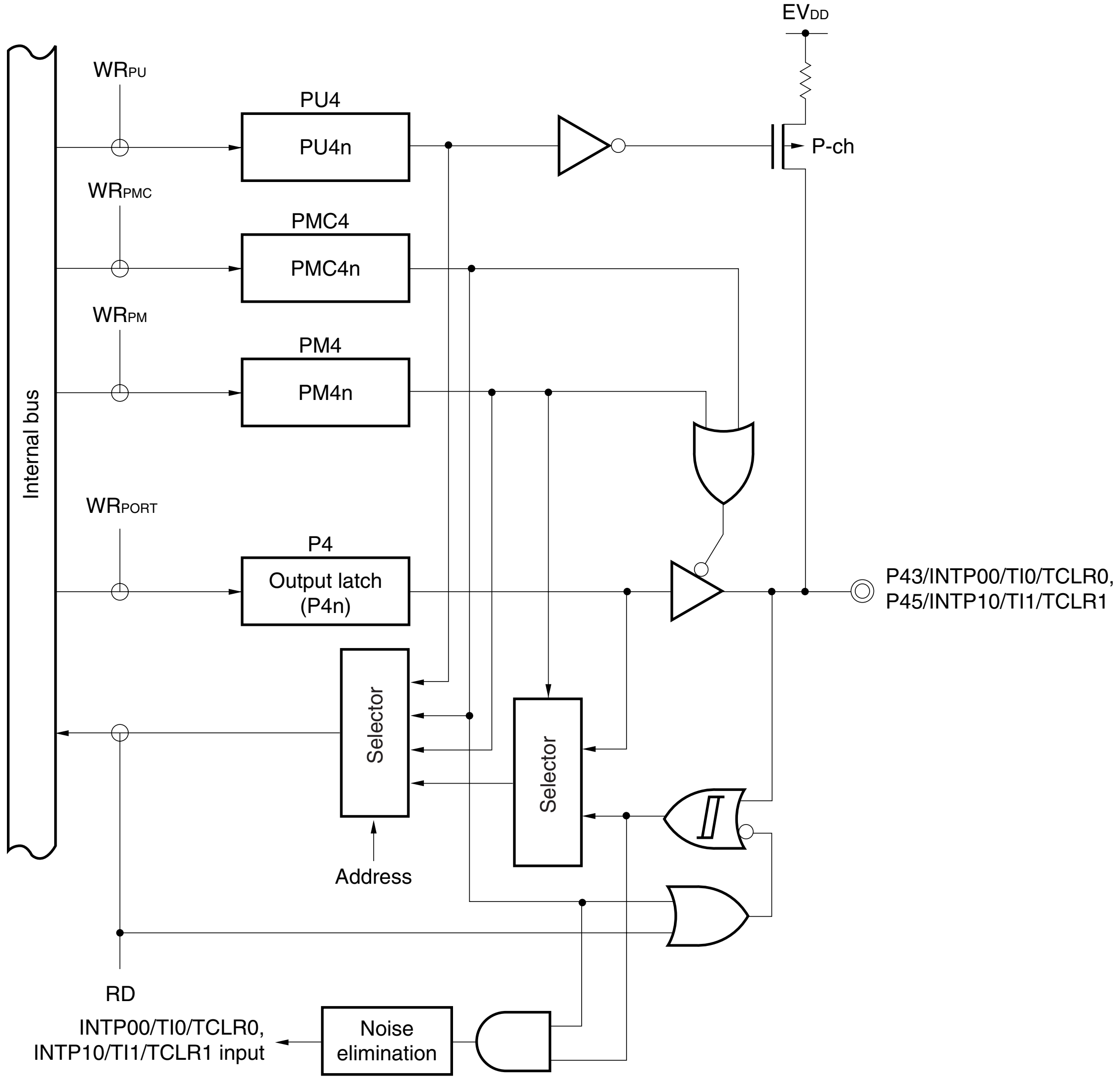
Figure 4-13. Block Diagram of P42



Caution This pin does not have hysteresis characteristics in the port mode.
It has hysteresis characteristics only when an input-pin alternate function is used.

Remark		
	P4:	Port register 4
	PM4:	Port mode register 4
	PMC4:	Port mode control register 4
	PFC4:	Port function control register 4
	PF4:	Port function register 4
	PU4:	Pull-up resistor option register 4

Figure 4-14. Block Diagram of P43 and P45



Caution These pins do not have hysteresis characteristics in the port mode.
They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P4: Port register 4

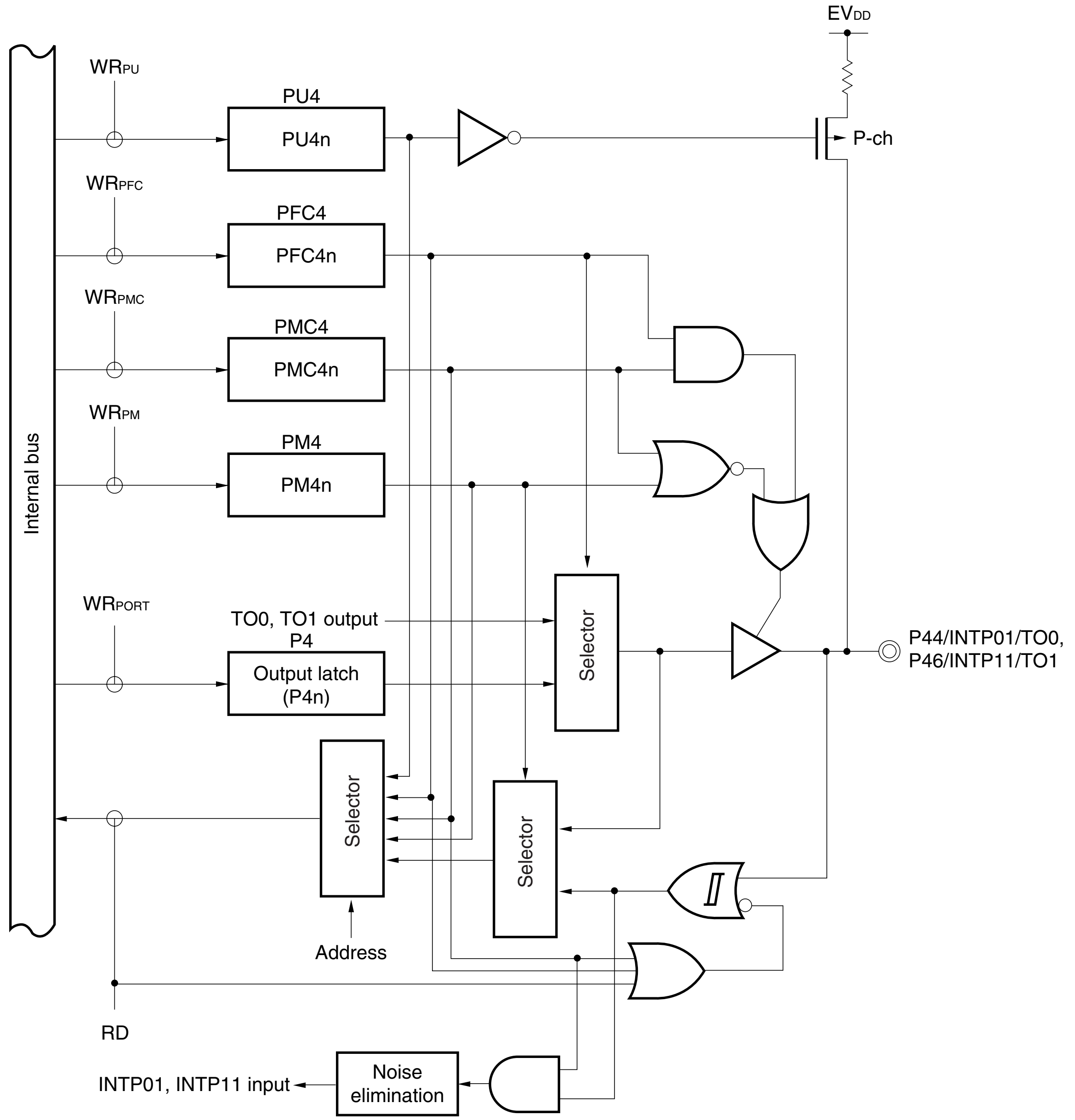
PM4: Port mode register 4

PMC4: Port mode control register 4

PU4: Pull-up resistor option register 4

2. n = 3 or 5

Figure 4-15. Block Diagram of P44 and P46



Caution These pins do not have hysteresis characteristics in the port mode.
They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P4: Port register 4
PM4: Port mode register 4
PMC4: Port mode control register 4
PFC4: Port function control register 4
PU4: Pull-up resistor option register 4

2. n = 4 or 6

4.3.5 Port 7

All the pins of port 7 are fixed to the input mode.

The number of input port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	12-bit input port
V850ES/SA3	16-bit input port

(1) Function of port 7

○ Input data can be specified in 1-bit units by using port register 7 (P7).

Port 7 has an alternate function as the following pins.

Table 4-8. Alternate-Function Pins of Port 7

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port 7	P70	ANI0	Input	None	–
	P71	ANI1			
	P72	ANI2			
	P73	ANI3			
	P74	ANI4			
	P77	ANI5			
	P76	ANI6			
	P77	ANI7			
	P78	ANI8			
	P79	ANI9			
	P710	ANI10			
	P711	ANI11			
	P712[Note 2]	ANI12[Note 2]			
	P713[Note 2]	ANI13[Note 2]			
	P714[Note 2]	ANI14[Note 2]			
	P715[Note 2]	ANI15[Note 2]			

Notes 1. Software pull-up function

2. V850ES/SA3 only

(2) Register

(a) Port register 7 (P7)

Port register 7 is a 16-bit register that is used to read the pin level.

This register is read-only, in 16-bit units.

If the higher 8 bits of the P7 register are used as P7H, and the lower 8 bits as P7L, however, this register can be read in 8-bit or 1-bit units.

After reset: Undefined R Address: P7 FFFFF40EH,
P7L FFFFF40EH, P7H FFFFF40FH

	15	14	13	12	11	10	9	8
P7 (P7H[Note 1])	P715[Note 2]	P714[Note 2]	P713[Note 2]	P712[Note 2]	P711	P710	P79	P78

	7	6	5	4	3	2	1	0
(P7L)	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Reads input data (V850ES/SA2: n = 0 to 12, V850ES/SA3: n = 0 to 15)
0	Inputs low level.
1	Inputs high level.

Notes 1. When reading or writing bits 8 to 15 of the P7 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the P7H register.

2. Bits 15 to 12 are valid only in the V850ES/SA3. These bits are undefined in the V850ES/SA2.

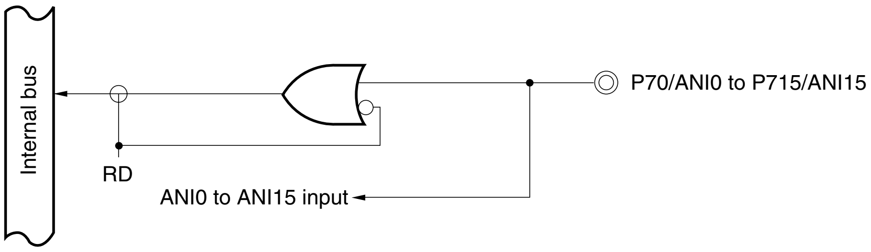
Caution Do not read the P7 register during A/D conversion.

Remarks 1. If port 7 (P7) is read, the pin levels at that time can be read.

2. After reset, an undefined value (pin input level) is read when P7 is read.

(3) Block diagram

Figure 4-16. Block Diagram of P70 to P715



4.3.6 Port 8

Port 8 can control input/output in 1-bit units.

The number of I/O port bits of each product is the same.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	2-bit I/O port
V850ES/SA3	2-bit I/O port

(1) Function of port 8

○ Input data can be specified in 1-bit units by using port register 8 (P8).

Port 8 has an alternate function as the following pins.

Table 4-9. Alternate-Function Pins of Port 8

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark
Port 8	P80	ANO0	Input	None	–
	P81	ANO1			

Note Software pull-up function

(2) Register

(a) Port register 8 (P8)

Port register 8 (P8) is an 8-bit register that is used to read the pin level.

This register is read-only, in 8-bit or 1-bit units.

After reset: Undefined R Address: FFFFF410H

	7	6	5	4	3	2	1	0
P8	0	0	0	0	0	0	P81	P80

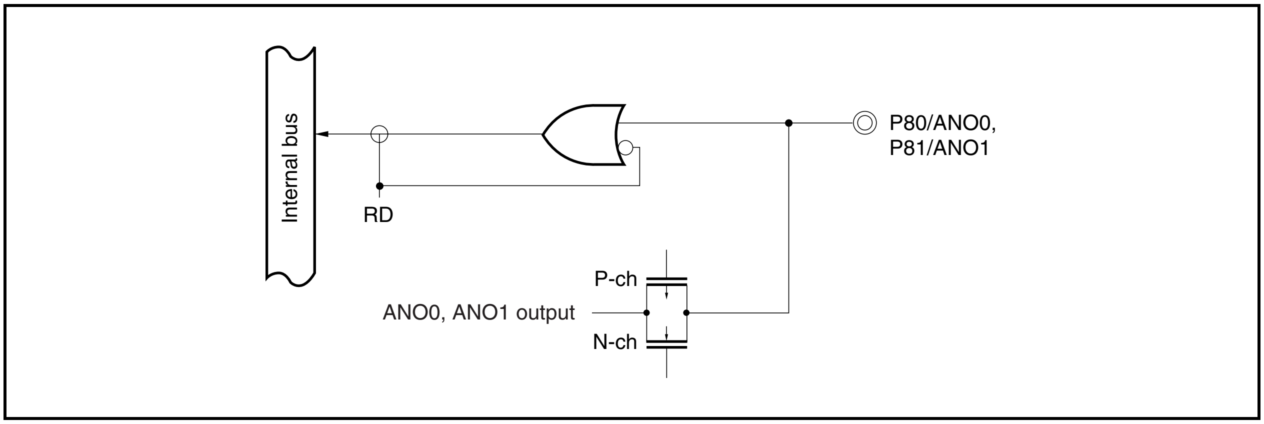
P8n	Reads input data (n = 0 or 1)
0	Inputs low level.
1	Inputs high level.

Remarks 1. If port 8 (P8) is read, the pin levels at that time can be read.

2. After reset, an undefined value (pin input level) is read when P8 is read.

(3) Block diagram

Figure 4-17. Block Diagram of P80 and P81



4.3.7 Port 9

Port 9 can be set to the input or output mode in 1-bit units.

The number of I/O port bits each product is the same.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	16-bit I/O port
V850ES/SA3	16-bit I/O port

(1) Functions of port 9

○ Input/output data can be specified in 1-bit units by using port register 9 (P9).

○ Can be set to the input or output mode in 1-bit units by using port mode register 9 (PM9).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register 9 (PMC9).

○ N-ch open-drain output can be set in 1-bit units by using port function register 9 (PF9).

○ Control mode 1 or control mode 2 can be specified in 1-bit units by using port function control register 9 (PFC9).

○ The internal pull-up resistor can be connected in 1-bit units by using pull-up resistor option register 9 (PU9).

○ The valid edge of the external interrupt (alternate function) can be set in 1-bit units by using external interrupt falling edge specification register 9 (INTF9) and external interrupt rising edge specification register 9 (INTR9).

Port 9 has an alternate function as the following pins.

Table 4-10. Alternate-Function Pins of Port 9

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark
Port 9	P90	A0	I/O	None	–
	P91	A1			
	P92	A2/INTP5			
	P93	A3/INTP6			
	P94	A4/TO2			
	P95	A5/TO3			
	P96	A6/TO4			
	P97	A7/TO5			
	P98	A8/RXD1			
	P99	A9/TXD1			
	P910	A10/SI2			
	P911	A11/SO2			
	P912	A12/$\overline{\text{SCK2}}$			
	P913	A13/SI3			
	P914	A14/SO3			
	P915	A15/$\overline{\text{SCK3}}$			

Note Software pull-up function

(2) Registers

(a) Port register 9 (P9)

Port register 9 (P9) is a 16-bit register that controls reading a pin level and writing an output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the P9 register is used as P9H and the lower 8 bits as P9L, however, P9H and P9L can be manipulated in 8-bit or 1-bit units.

After reset: Undefined R/W Address: P9 FFFFF412H,
P9L FFFFF412H, P9H FFFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H[Note])	P915	P914	P913	P912	P911	P910	P99	P98

	7	6	5	4	3	2	1	0
(P9L)	P97	P96	P95	P94	P93	P92	P91	P90

P9n	Controls output data (in output mode) (n = 0 to 15)
0	Outputs 0.
1	Outputs 1.

Note When reading or writing bits 8 to 15 of the P9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the P9H register.

Remarks 1. In input mode: When port 9 (P9) is read, the pin level at that time is read. When written, the data written to P9 is written. The input pin is not affected.

In output mode: When port 9 (P9) is read, the value of P9 is read. When a value is written to P9, it is immediately output.

2. After reset, an undefined value (pin input level) is read from P9 in the input mode. When P9 is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register 9 (PM9)

This is a 16-bit register that specifies the input or output mode.

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

If the higher 8 bits of the PM9 register is used as PM9H and the lower 8 bits as PM9L, however, PM9H and PM9L can be manipulated in 8-bit or 1-bit units.

After reset: FFFFH   R/W   Address: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H FFFFF433H

	15	14	13	12	11	10	9	8
(PM9H[Note])	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98

	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	Controls input/output mode (n = 0 to 15)
0	Output mode
1	Input mode

Note When reading or writing bits 8 to 15 of the PM9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PM9H register.

(c) Port mode control register 9 (PMC9)

This is a 16-bit register that specifies the port mode or control mode.

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

If the higher 8 bits of the PMC9 register is used as PMC9H and the lower 8 bits as PMC9L, however, PMC9H and PMC9L can be manipulated in 8-bit or 1-bit units.

(1/2)

After reset: 0000H R/W Address: PMC9 FFFFF452H,
PMC9L FFFFF452H, PMC9H FFFFF453H

	15	14	13	12	11	10	9	8
PMC9 (PMC9H[Note])	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC915	Specifies operation mode of P915 pin
0	I/O port
1	A15/$\overline{\text{SCK3}}$ I/O

PMC914	Specifies operation mode of P914 pin
0	I/O port
1	A14/SO3 output

PMC913	Specifies operation mode of P913 pin
0	I/O port
1	A13/SI3 I/O

PMC912	Specifies operation mode of P912 pin
0	I/O port
1	A12/$\overline{\text{SCK2}}$ I/O

PMC911	Specifies operation mode of P911 pin
0	I/O port
1	A11/SO2 output

PMC910	Specifies operation mode of P910 pin
0	I/O port
1	A10/SI2 I/O

PMC99	Specifies operation mode of P99 pin
0	I/O port
1	A9/TXD1 output

PMC98	Specifies operation mode of P98 pin
0	I/O port
1	A8/RXD1 I/O

Note When reading or writing bits 8 to 15 of the PMC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMC9H register.

(2/2)

PMC97	Specifies operation mode of P97 pin
0	I/O port
1	A7/TO5 output

PMC96	Specifies operation mode of P96 pin
0	I/O port
1	A6/TO4 output

PMC95	Specifies operation mode of P95 pin
0	I/O port
1	A5/TO3 output

PMC94	Specifies operation mode of P94 pin
0	I/O port
1	A4/TO2 output

PMC93	Specifies operation mode of P93 pin
0	I/O port
1	A3/INTP6 I/O

PMC92	Specifies operation mode of P92 pin
0	I/O port
1	A2/INTP5 I/O

PMC91	Specifies operation mode of P91 pin
0	I/O port
1	A1 output

PMC90	Specifies operation mode of P90 pin
0	I/O port
1	A0 output

(d) Port function control register 9 (PFC9)

This 16-bit register specifies control mode 1 or control mode 2.

It can be read or written only in 16-bit units.

If the higher 8 bits of the PFC9 register are used as PFC9H and the lower 8 bits as PFC9L, however, PFC9H and PFC9L can be manipulated in 8-bit or 1-bit units.

Caution To perform separate bus address output (A0 to A15), set the PFC9 register to 0000H, and then set the PMC9 register to FFFFH in 16-bit units.

(1/2)

After reset: 0000H R/W Address: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9 (PFC9H[Note])	PFC910	PFC910	PFC910	PFC910	PFC910	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	0	0

PFC915	Specifies operation mode of P915 pin in control mode
0	A15 output (with separate bus)
1	$\overline{\text{SCK3}}$ I/O

PFC914	Specifies operation mode of P914 pin in control mode
0	A14 output (with separate bus)
1	SO3 output

PFC913	Specifies operation mode of P913 pin in control mode
0	A13 output (with separate bus)
1	SI3 input

PFC912	Specifies operation mode of P912 pin in control mode
0	A12 output (with separate bus)
1	$\overline{\text{SCK2}}$ I/O

PFC911	Specifies operation mode of P911 pin in control mode
0	A11 output (with separate bus)
1	SO2 output

PFC910	Specifies operation mode of P910 pin in control mode
0	A10 output (with separate bus)
1	SI2 input

PFC99	Specifies operation mode of P99 pin in control mode
0	A9 output (with separate bus)
1	TXD1 output

Note When reading or writing bits 8 to 15 of the PFC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PFC9H register.

(2/2)

PFC98	Specifies operation mode of P98 pin in control mode
0	A8 output (with separate bus)
1	RXD1 input

PFC97	Specifies operation mode of P97 pin in control mode
0	A7 output (with separate bus)
1	TO5 output

PFC96	Specifies operation mode of P96 pin in control mode
0	A6 output (with separate bus)
1	TO4 output

PFC95	Specifies operation mode of P95 pin in control mode
0	A5 output (with separate bus)
1	TO3 output

PFC94	Specifies operation mode of P94 pin in control mode
0	A4 output (with separate bus)
1	TO2 output

PFC93	Specifies operation mode of P93 pin in control mode
0	A3 output (with separate bus)
1	INTP6 input

PFC92	Specifies operation mode of P92 pin in control mode
0	A2 output (with separate bus)
1	INTP5 input

(e) Port function register 9 (PF9)

This 16-bit register specifies normal output or N-ch open-drain output.

The PF9 register can be read or written only in 16-bit units. If the higher 8 bits of the PF9 register are used as PF9H and the lower 8 bits as PF9L, however, PF9H and PF9L can be manipulated in 8-bit or 1-bit units.

After reset: 0000H R/W Address: PF9 FFFFFC72H, PF9H FFFFFC73H

	15	14	13	12	11	10	9	8
PF9 (PF9H[Note])	PF915	PF914	0	PF912	PF911	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

PF9n	Controls normal output or N-ch open-drain output (n = 11, 12, 14, or 15)
0	Normal output
1	N-ch open-drain output

Note When reading or writing bits 8 to 15 of the PF9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PF9H register.

Cautions 1. The N-ch open-drain output voltage is the normal voltage, not the medium voltage.

2. PF9n = 1 is enabled only in the following cases. Otherwise, the setting is prohibited.

n = 1: SO2
n = 2: $\overline{\text{SCK2}}$
n = 4: SO3
n = 5: $\overline{\text{SCK3}}$

3. Follow the procedure below to set N-ch open drain.

<1> Set the P9n bit to 1.

<2> Set the PF9n bit to 1.

<3> Set the CSIE2 bit of the CSIM2 register or the CSIE3 bit of the CSIM3 register to 1.

<4> Set the PFC9n bit.

<5> Set the PMC9n bit to 1.

(f) Pull-up resistor option register 9 (PU9)

This is a 16-bit register that specifies connection of an internal pull-up resistor.

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

If the higher 8 bits of the PU9 register are used as PU9H and the lower 8 bits as PU9L, however, PU9H and PU9L can be manipulated in 8-bit or 1-bit units.

After reset: 0000H R/W Address: PU9 FFFFFC52H,
PU9L FFFFFC52H, PU9H FFFFFC53H

	15	14	13	12	11	10	9	8
PU9 (PU9H[Note])	PU915	PU914	PU913	PU912	PU911	PU910	PU99	PU98

	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90

PU9n	Controls connection of internal pull-up resistor (n = 0 to 15)
0	Not connected
1	Connected

Note When reading or writing bits 8 to 15 of the PU9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PU9H register.

(g) External interrupt falling edge specification register 9 (INTF9)

This 16-bit register specifies detection of the falling edge of the external interrupt pins.

It can be read or written only in 16-bit units. If the higher 8 bits of the INTF9 register are used as INTF9H and the lower 8 bits as INTF9L, however, INTF9H and INTF9L can be manipulated in 8-bit or 1-bit units.

Caution Set the port mode after clearing the INTF9n and INTR9n bits to 0 when switching from the external interrupt function (alternate function) to the port function because an edge may be detected.

After reset: 0000H R/W Address: INTF9 FFFFFC12H, INTF9L FFFFFC12H

	15	14	13	12	11	10	9	8
INTF9	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(INTF9L)	0	0	0	0	INTF93	INTF92	0	0

Remark For how to specify a valid edge, refer to **Table 4-11**.

(h) External interrupt rising edge specification register 9 (INTR9)

This 16-bit register specifies detection of the rising edge of the external interrupt pins.

It can be read or written only in 16-bit units. If the higher 8 bits of the INTR9 register are used as INTR9H and the lower 8 bits as INTR9L, however, INTR9H and INTR9L can be manipulated in 8-bit or 1-bit units.

Caution Set the port mode after clearing the INTF9n and INTR9n bits to 0 when switching from the external interrupt function (alternate function) to the port function because an edge may be detected.

After reset: 0000H R/W Address: INTR9 FFFFFC32H, INTR9L FFFFFC32H

	15	14	13	12	11	10	9	8
INTR9	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(INTR9L)	0	0	0	0	INTR93	INTR92	0	0

Remark For how to specify a valid edge, refer to **Table 4-11**.

Table 4-11. Specifying Valid Edge

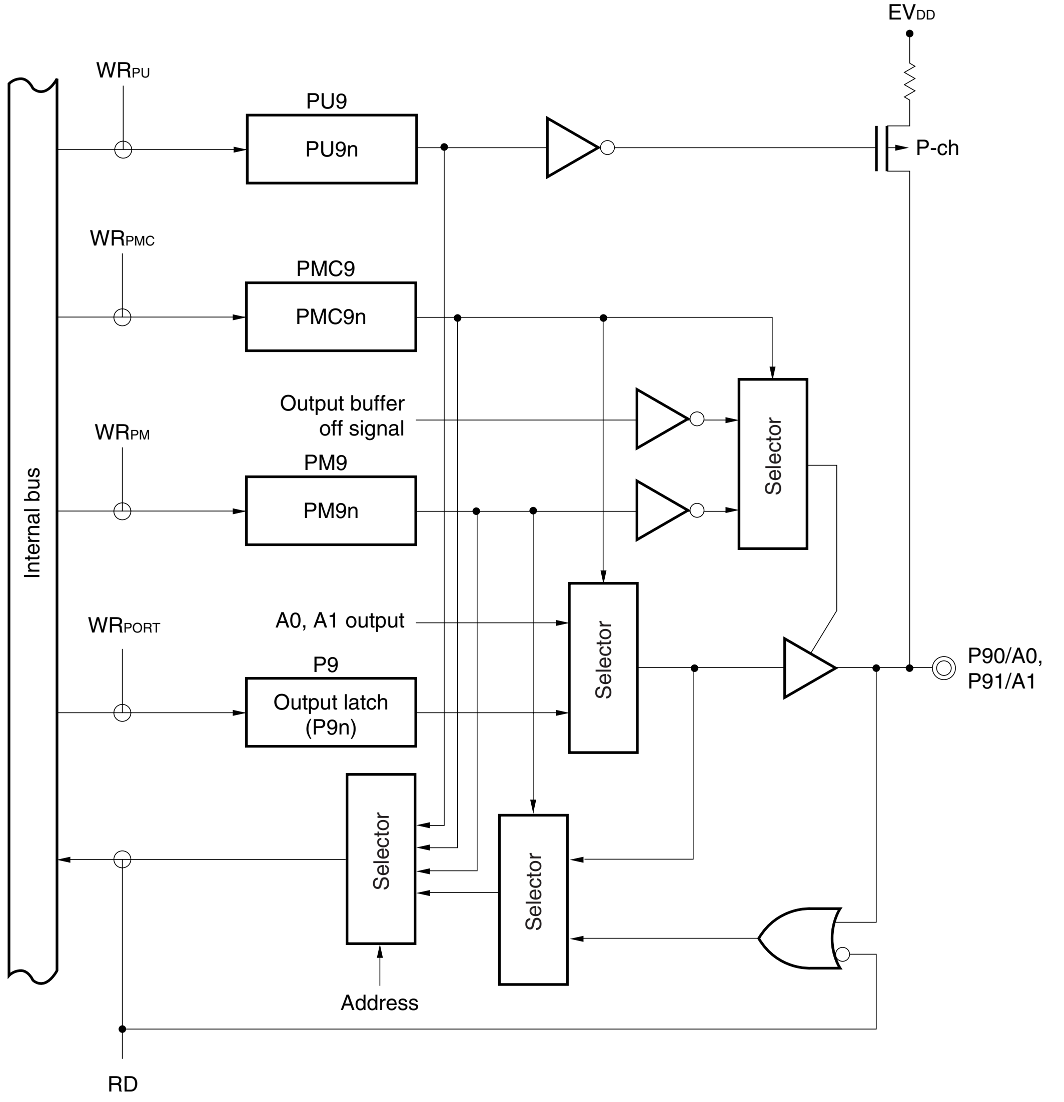
INTF9n	INTR9n	Specifies valid edge (n = 2 or 3).
0	0	Detects no edge.
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Caution When INTP5 and INTP6 are not used, be sure to clear INTF9n and INTR9n to “00”.

Remark n = 2 or 3: Control of INTP5 or INTP6 pin

(3) Block diagram

Figure 4-18. Block Diagram of P90 and P91



Remarks 1. P9: Port register 9

PM9: Port mode register 9

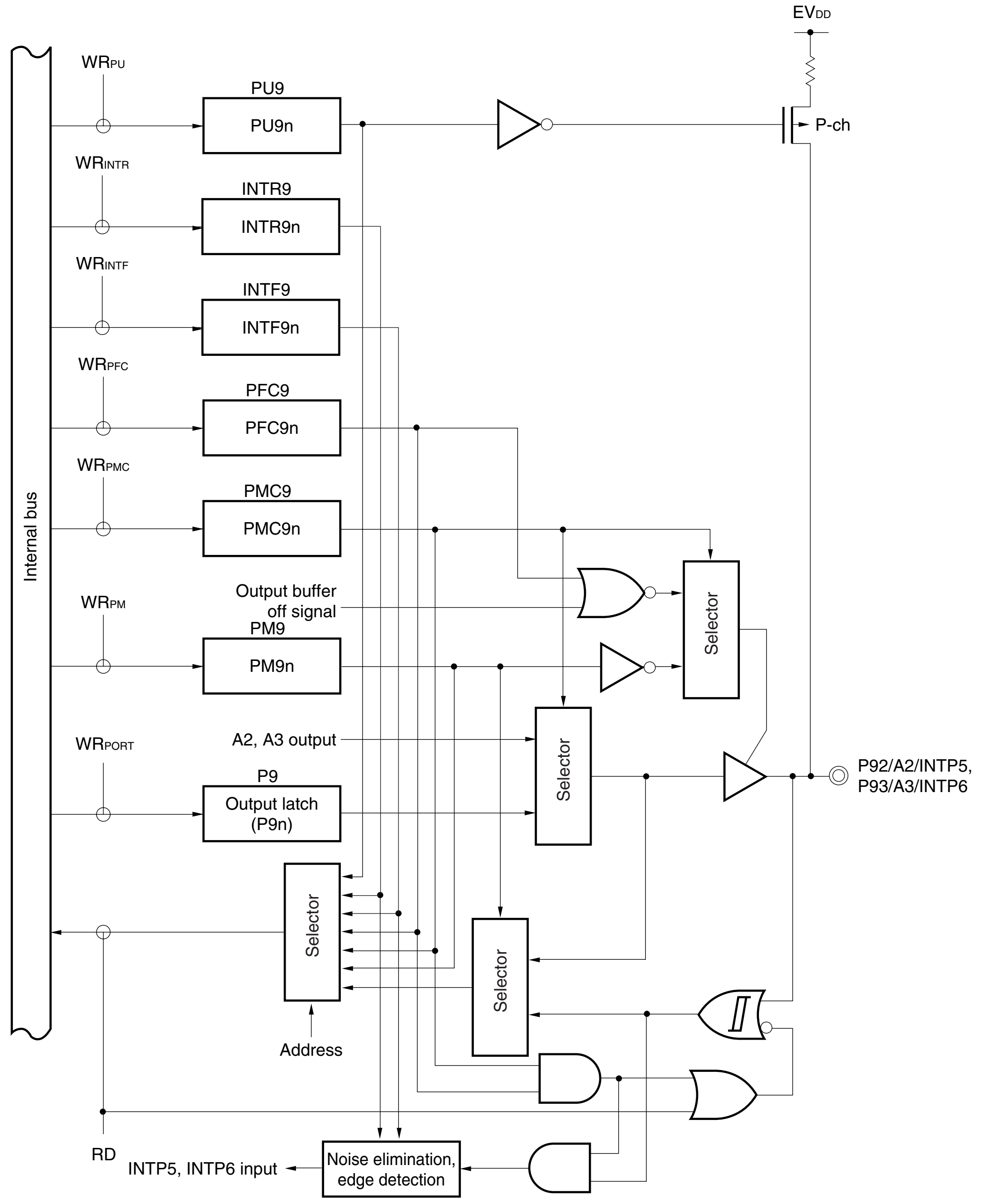
PMC9: Port mode control register 9

PU9: Pull-up resistor option register 9

Output buffer off signal: Signal that is active in the IDLE/STOP mode or during bus hold

2. n = 0 or 1

Figure 4-19. Block Diagram of P92 and P93

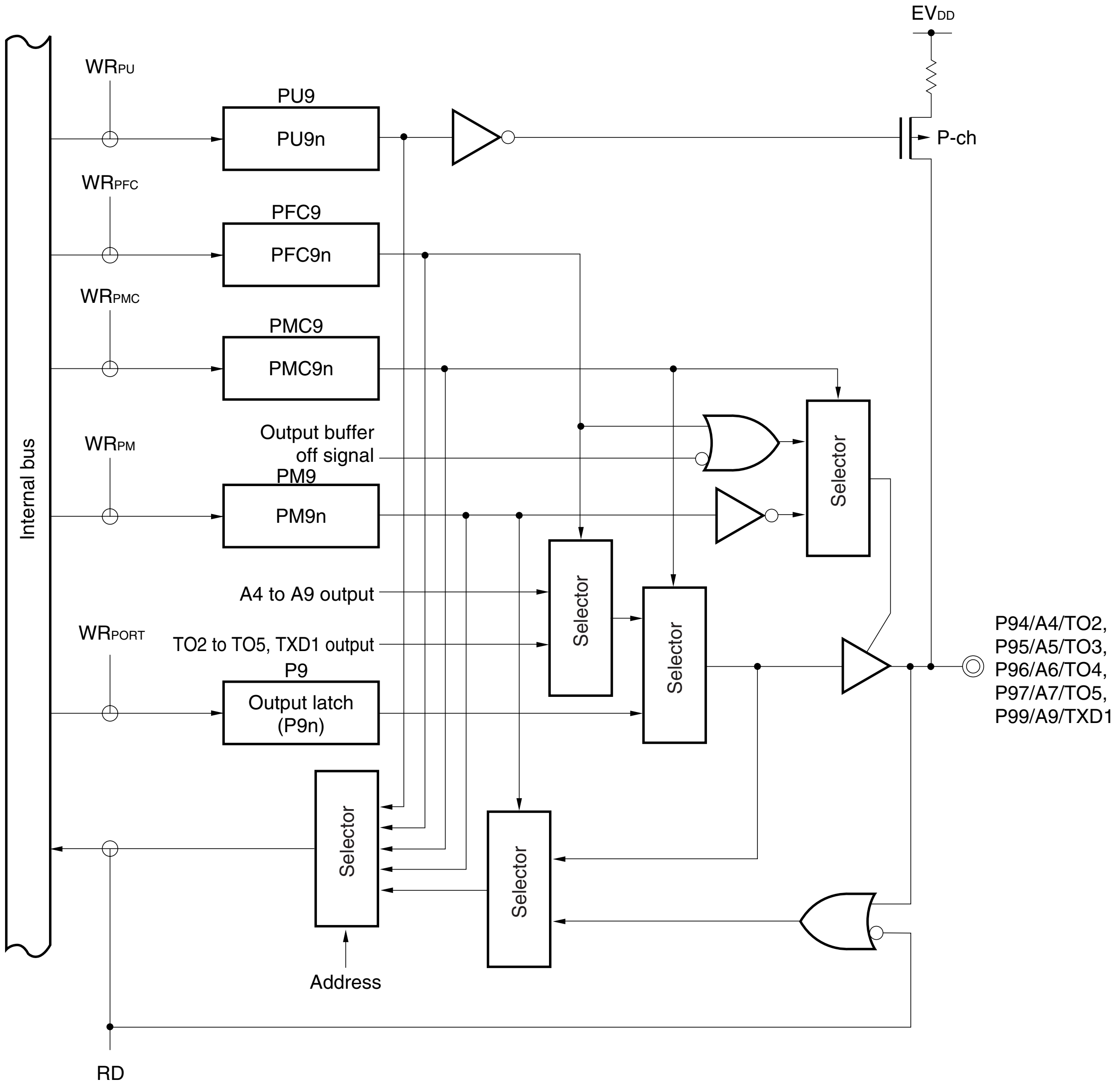


Caution These pins do not have hysteresis characteristics in the port mode. They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P9: Port register 9
PM9: Port mode register 9
PMC9: Port mode control register 9
PFC9: Port function control register 9
PU9: Pull-up resistor option register 9
INTR9: External interrupt rising edge specification register 9
INTF9: External interrupt falling edge specification register 9
Output buffer off signal: Signal that is active in the IDLE/STOP mode or during bus hold

2. n = 2 or 3

Figure 4-20. Block Diagram of P94 to P97 and P99



Remarks 1. P9: Port register 9

PM9: Port mode register 9

PMC9: Port mode control register 9

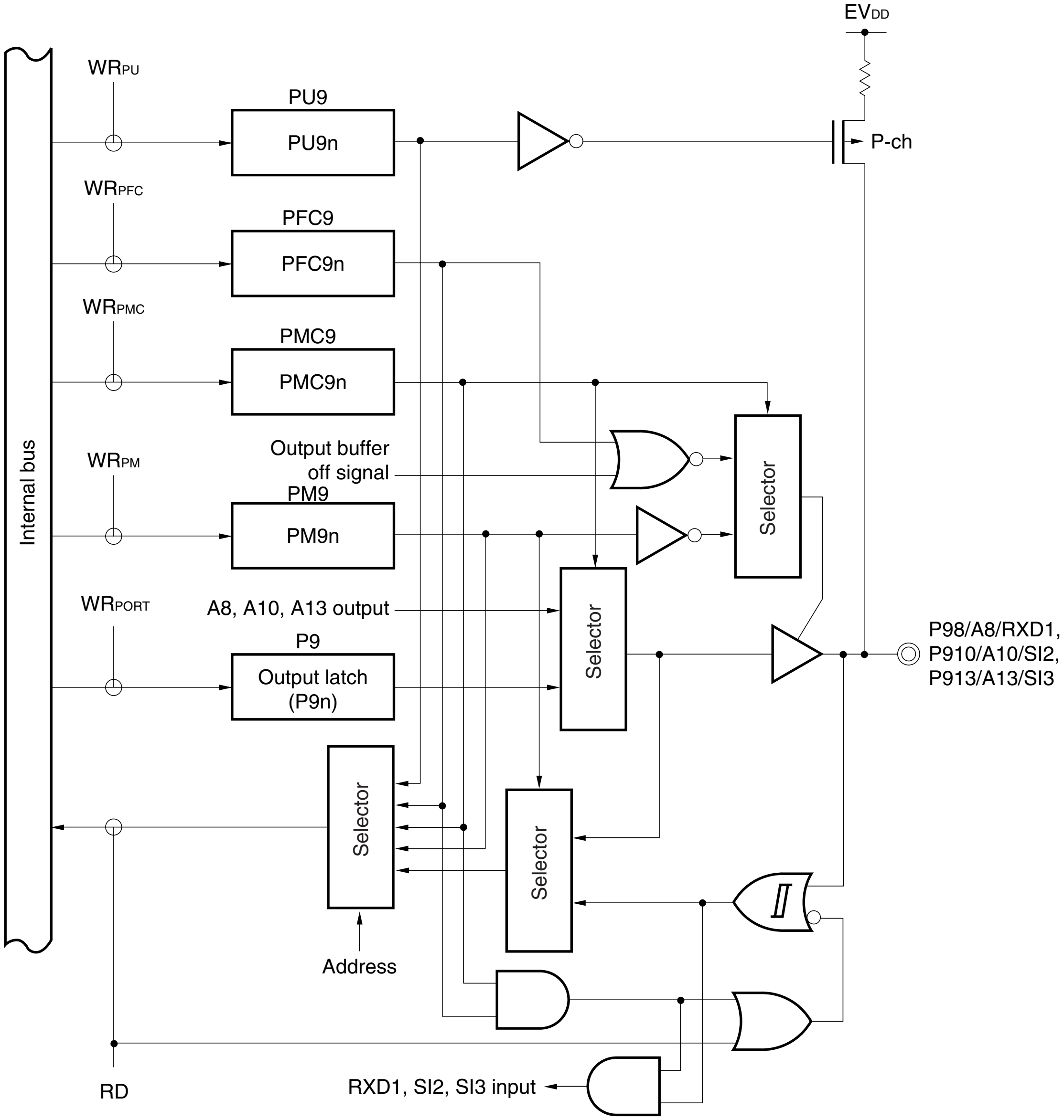
PF9: Port function register 9

PU9: Pull-up resistor option register 9

Output buffer off signal: Signal that is active in the IDLE/STOP mode or during bus hold

2. n = 4 to 7, 9

Figure 4-21. Block Diagram of P98, P910, and P913



Caution These pins do not have hysteresis characteristics in the port mode. They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P9: Port register 9

PM9: Port mode register 9

PMC9: Port mode control register 9

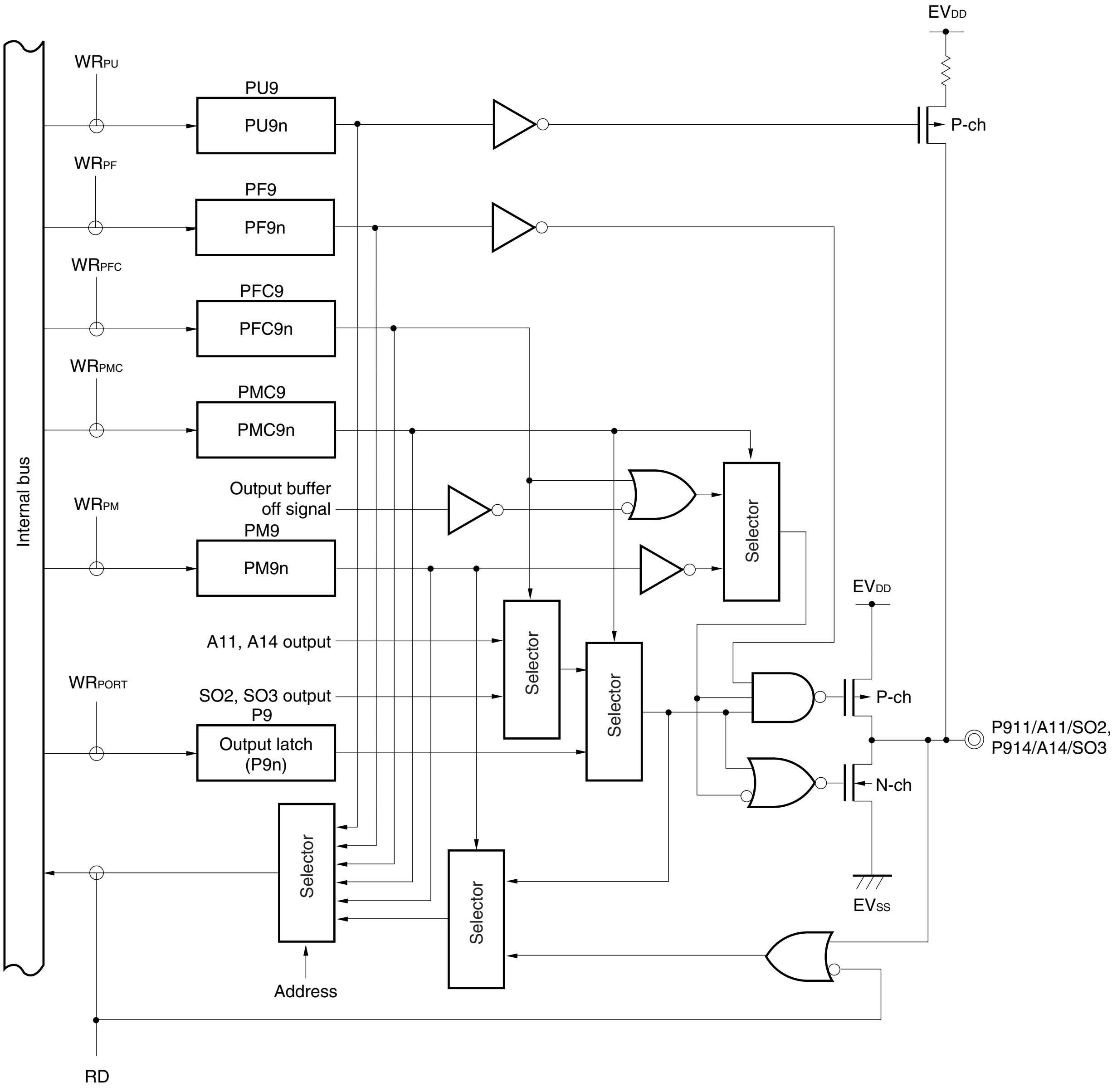
PF9: Port function register 9

PU9: Pull-up resistor option register 9

Output buffer off signal: Signal that is active in the IDLE/STOP mode or during bus hold

2. n = 8, 10, 13

Figure 4-22. Block Diagram of P911 and P914



Remarks 1. P9: Port register 9

PM9: Port mode register 9

PMC9: Port mode control register 9

PFC9: Port function control register 9

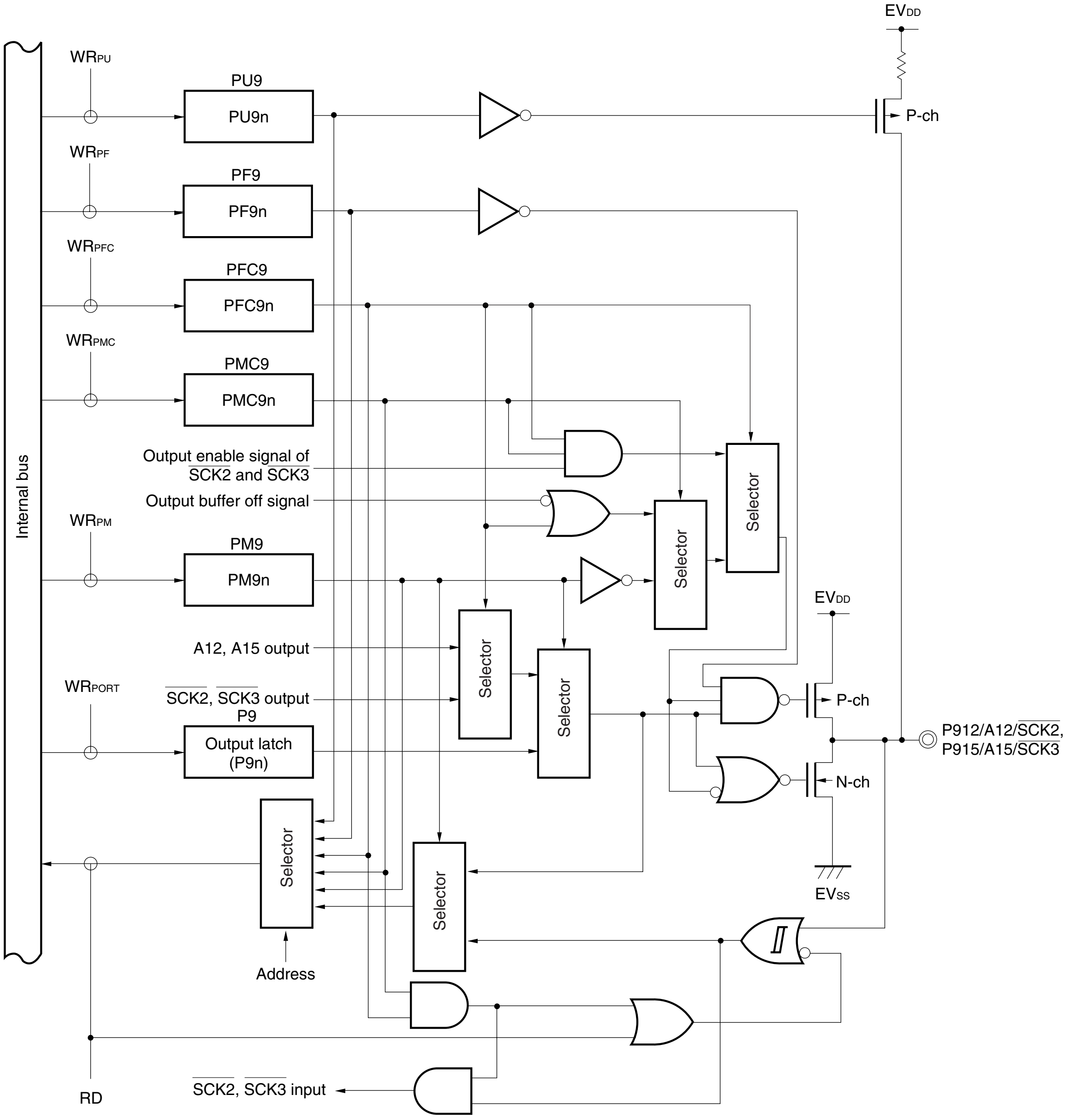
PF9: Port function register 9

PU9: Pull-up resistor option register 9

Output buffer off signal: Signal that is active in the IDLE/STOP mode or during bus hold

2. n = 11, 14

Figure 4-23. Block Diagram of P912 and P915



Caution These pins do not have hysteresis characteristics in the port mode.
They have hysteresis characteristics only when an input-pin alternate function is used.

Remarks 1. P9: Port register 9

PM9: Port mode register 9

PMC9: Port mode control register 9

PFC9: Port function control register 9

PF9: Port function register 9

PU9: Pull-up resistor option register 9

Output buffer off signal: Signal that is active in the IDLE/STOP mode or during bus hold

2. n = 12 or 15

4.3.8 Port CD

Port CD can be set to the input or output mode in 1-bit units.

The number of I/O port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	–
V850ES/SA3	3-bit I/O port

(1) Functions of port CD (V850ES/SA3)

○ Input/output data can be specified in 1-bit units by using port register CD (PCD).

○ Can be set to the input or output mode in 1-bit units by using port mode register CD (PMCD).

Port CD has no alternate-function pins.

Table 4-12. Alternate-Function Pins of Port CD (V850ES/SA3)

Pin Name		Alternate-Function Pin	I/O	PULL[Note]	Remark
Port CD	PCD1	–	I/O	None	–
	PCD2	–			
	PCD3	–			

Note Software pull-up function

(2) Registers

(a) Port register CD (PCD)

Port register CD (PCD) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF00EH

	7	6	5	4	3	2	1	0
PCD	0	0	0	0	PCD3	PCD2	PCD1	0

PCDn	Controls output data (in output mode) (n = 1 to 3)
0	Outputs 0.
1	Outputs 1.

Remarks 1. In input mode: When port CD (PCD) is read, the pin level at that time is read. When written, the data written to PCD is written. The input pin is not affected.

In output mode: When port CD (PCD) is read, the value of PCD is read. When a value is written to PCD, it is immediately output.

2. After reset, an undefined value (pin input level) is read from PCD in the input mode. When PCD is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register CD (PMCD)

This is an 8-bit register that specifies the input or output mode.

This register can be read or written in 8-bit or 1-bit units.

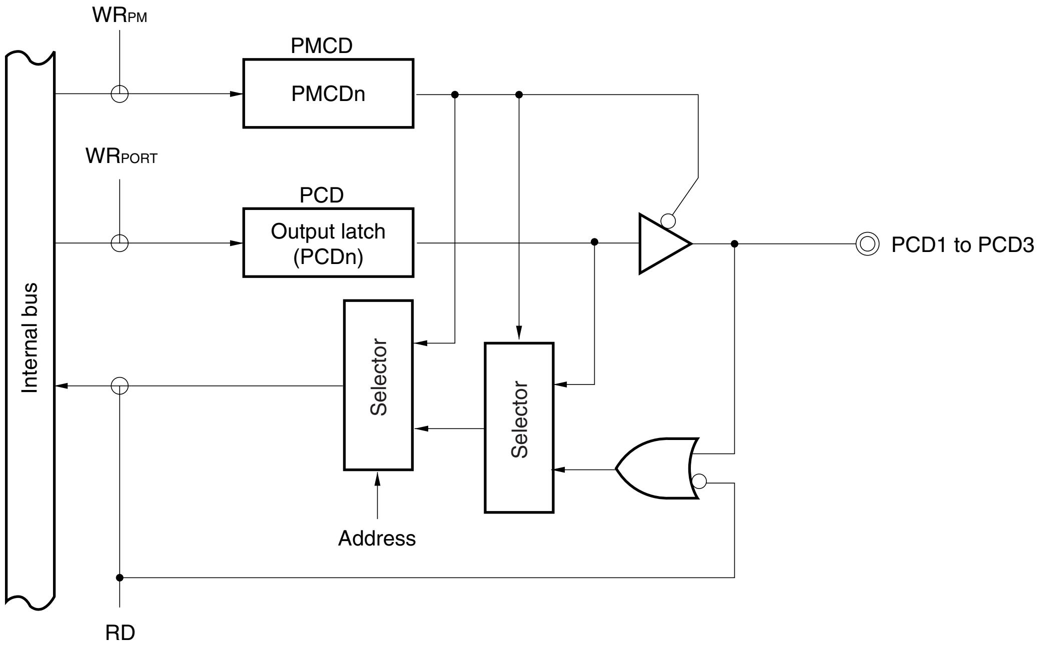
After reset: FFH R/W Address: FFFFF02EH

	7	6	5	4	3	2	1	0
PMCD	1	1	1	1	PMCD3	PMCD2	PMCD1	1

PMCDn	Controls input/output mode (n = 1 to 3)
0	Output mode
1	Input mode

(3) Block diagram

Figure 4-24. Block Diagram of PCD1 to PCD3



Remarks 1. PCD: Port register CD

PMCD: Port mode register CD

2. n = 1 to 3

4.3.9 Port CM

Port CM can be set to the input or output mode in 1-bit units.

The number of I/O port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	4-bit I/O port
V850ES/SA3	6-bit I/O port

(1) Functions of port CM

○ Input/output data can be specified in 1-bit units by using port register CM (PCM).

○ Can be set to the input or output mode in 1-bit units by using port mode register CM (PMCM).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register CM (PMCCM).

Port CM has an alternate function as the following pins.

Table 4-13. Alternate-Function Pins of Port CM

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port CM	PCM0	$\overline{\text{WAIT}}$	I/O	None	–
	PCM1	CLKOUT			
	PCM2	$\overline{\text{HLDAK}}$			
	PCM3	$\overline{\text{HLDRQ}}$			
	PCM4[Note 2]	–			
	PCM5[Note 2]	–			

Notes 1. Software pull-up function

2. V850ES/SA3 only

(2) Registers

(a) Port register CM (PCM)

Port register PCM (PCM) is an 8-bit register that controls reading the pin level and writing the output level.
This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	PCM5[Note]	PCM4[Note]	PCM3	PCM2	PCM1	PCM0

PCMn	Controls output data (in output mode) (V850ES/SA2: n = 0 to 3, V850ES/SA3: n = 0 to 5)
0	Outputs 0.
1	Outputs 1.

Note Bits 5 and 4 are provided in the V850ES/SA3 only. Be sure to clear these bits to 0 in the V850ES/SA2.

Remarks 1. In input mode: When port CM (PCM) is read, the pin level at that time is read. When written, the data written to PCM is written. The input pin is not affected.

In output mode: When port CM (PCM) is read, the value of PCM is read. When a value is written to PCM, it is immediately output.

2. After reset, an undefined value (pin input level) is read from PCM in the input mode. When PCM is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register CM (PMCM)

This is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	PMCM5[Note]	PMCM4[Note]	PMCM3	PMCM2	PMCM1	PMCM0

PMCMn	Controls input/output mode (V850ES/SA2: n = 0 to 3, V850ES/SA3: n = 0 to 5)
0	Output mode
1	Input mode

Note Bits 5 and 4 are provided in the V850ES/SA3 only. Be sure to set these bits to 1 in the V850ES/SA2.

(c) Port mode control register CM (PMCCM)

This is an 8-bit register that specifies the port mode or control mode.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	PMCCM3	PMCCM2	PMCCM1	PMCCM0

PMCCM3	Specifies operation mode of PCM3 pin
0	I/O port
1	$\overline{\text{HLDRQ}}$ input

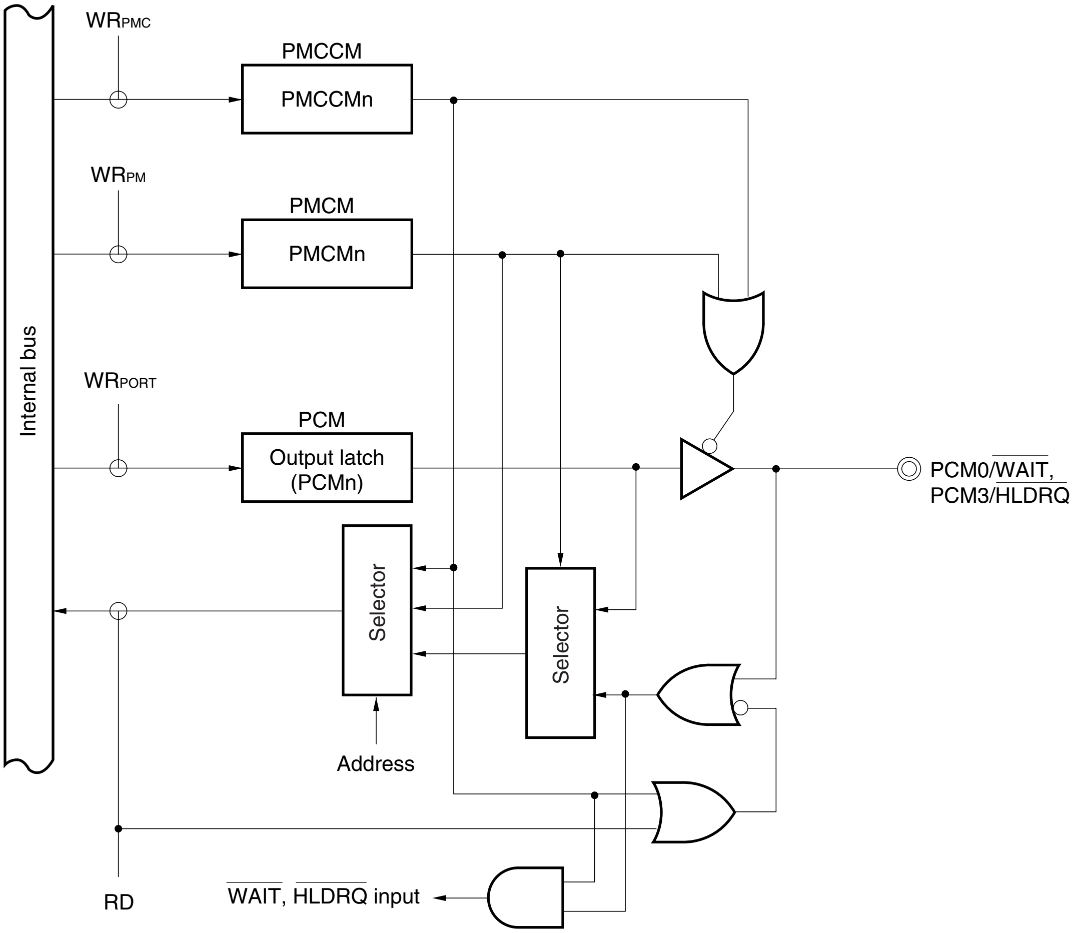
PMCCM2	Specifies operation mode of PCM2 pin
0	I/O port
1	$\overline{\text{HLDAK}}$ output

PMCCM1	Specifies operation mode of PCM1 pin
0	I/O port
1	CLKOUT output

PMCCM0	Specifies operation mode of PCM0 pin
0	I/O port
1	$\overline{\text{WAIT}}$ input

(3) Block diagram

Figure 4-25. Block Diagram of PCM0 and PCM3



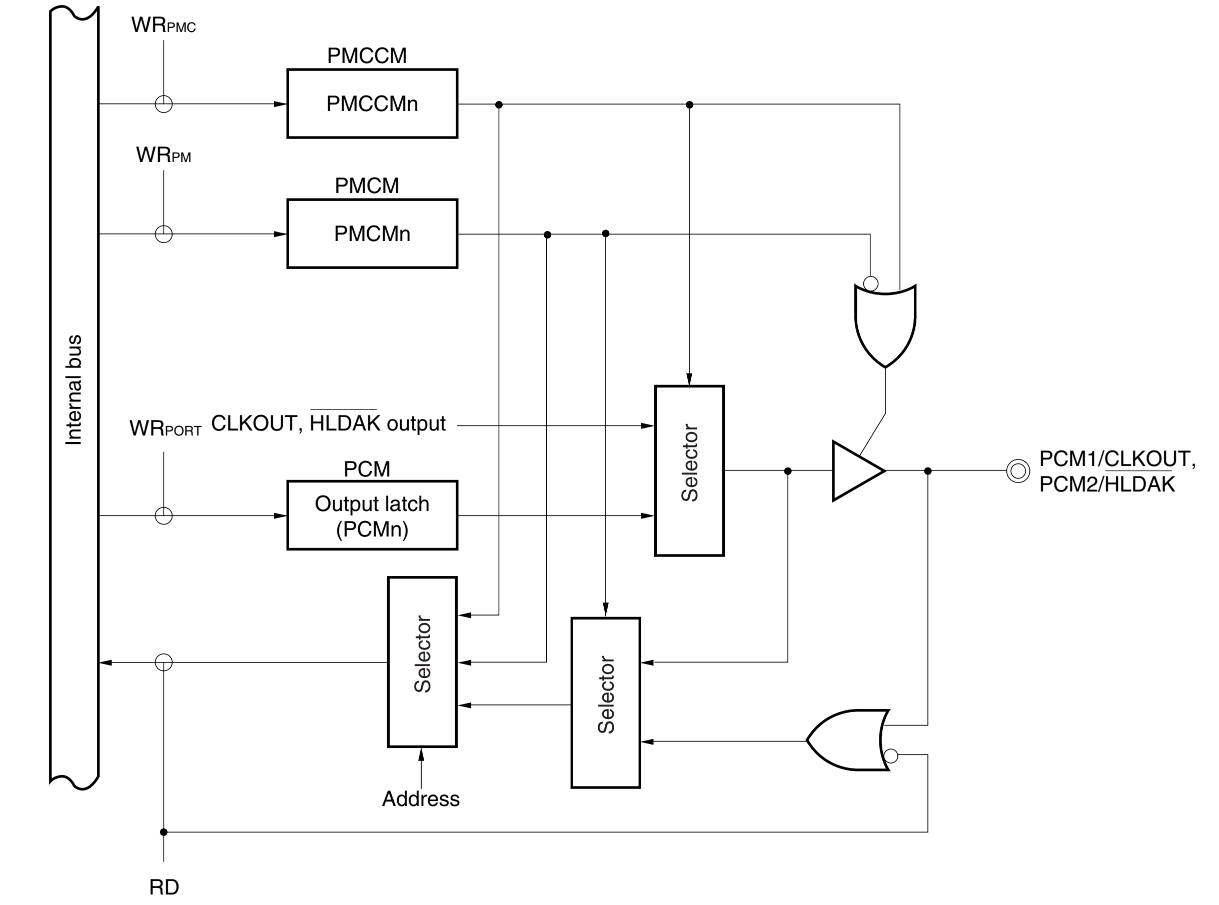
Remarks 1. PCM: Port register CM

PMCM: Port mode register CM

PMCCM: Port mode control register CM

2. n = 0 or 3

Figure 4-26. Block Diagram of PCM1 and PCM2



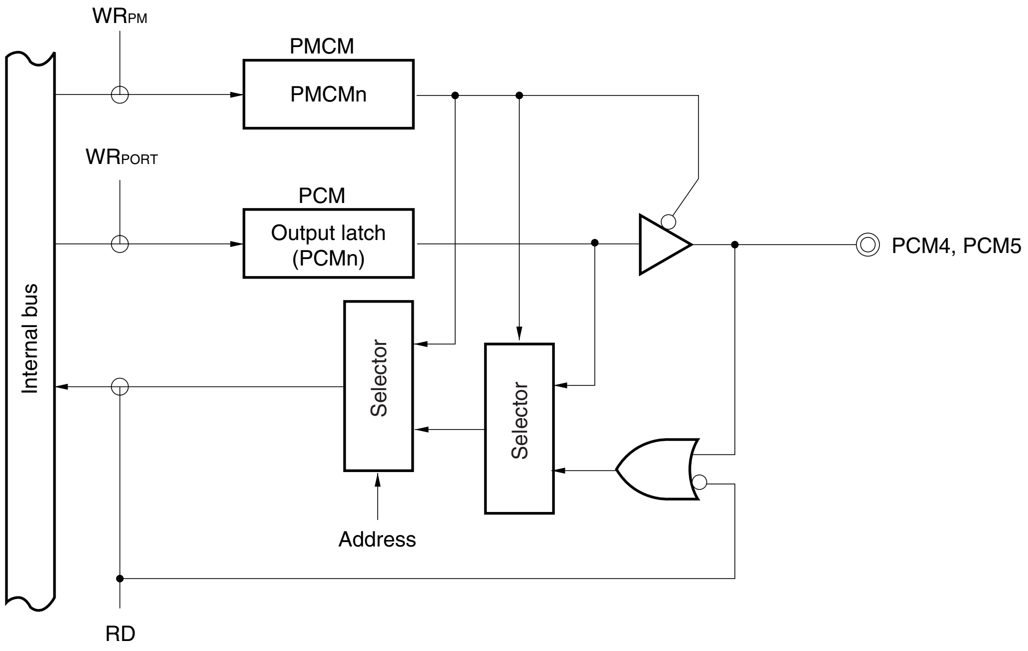
Remarks 1. PCM: Port register CM

PMCM: Port mode register CM

PMCCM: Port mode control register CM

2. n = 1 or 2

Figure 4-27. Block Diagram of PCM4 and PCM5



Remarks 1. PCM: Port register CM

PMCM: Port mode register CM

2. n = 4 or 5

4.3.10 Port CS

Port CS can be set to the input or output mode in 1-bit units.

The number of I/O port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	4-bit I/O port
V850ES/SA3	8-bit I/O port

(1) Functions of port CS

○ Input/output data can be specified in 1-bit units by using port register CS (PCS).

○ Can be set to the input or output mode in 1-bit units by using port mode register CS (PMCS).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register CS (PMCCS).

Port CS has an alternate function as the following pins.

Table 4-14. Alternate-Function Pins of Port CS

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port CS	PCS0	$\overline{CS0}$	I/O	None	–
	PCS1	$\overline{CS1}$			
	PCS2	$\overline{CS2}$			
	PCS3	$\overline{CS3}$			
	PCS4[Note 2]	–			
	PCS5[Note 2]	–			
	PCS6[Note 2]	–			
	PCS7[Note 2]	–			

Notes 1. Software pull-up function

2. V850ES/SA3 only

(2) Registers

(a) Port register CS (PCS)

Port register CS (PCS) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	PCS7[Note]	PCS6[Note]	PCS5[Note]	PCS4[Note]	PCS3	PCS2	PCS1	PCS0

PCSn	Controls output data (in output mode) (V850ES/SA2: n = 0 to 3, V850ES/SA3: n = 0 to 7)
0	Outputs 0.
1	Outputs 1.

Note Bits 7 to 4 are provided in the V850ES/SA3 only. Be sure to clear these bits to 0 in the V850ES/SA2.

Remarks 1. In input mode: When port CS (PCS) is read, the pin level at that time is read. When written, the data written to PCS is written. The input pin is not affected.

In output mode: When port CS (PCS) is read, the value of PCS is read. When a value is written to PCS, it is immediately output.

2. After reset, an undefined value (pin input level) is read from PCS in the input mode. When PCS is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register CS (PMCS)

This is an 8-bit register that specifies the input or output mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	PMCS7[Note]	PMCS6[Note]	PMCS5[Note]	PMCS4[Note]	PMCS3	PMCS2	PMCS1	PMCS0

PMCSn	Controls input/output mode (V850ES/SA2: n = 0 to 3, V850ES/SA3: n = 0 to 7)
0	Output mode
1	Input mode

Note Bits 7 to 4 are provided in the V850ES/SA3 only. Be sure to set these bits to 1 in the V850ES/SA2.

(c) Port mode control register CS (PMCCS)

This is an 8-bit register that specifies the port mode or control mode.

It can be read or written in 8-bit or 1-bit units.

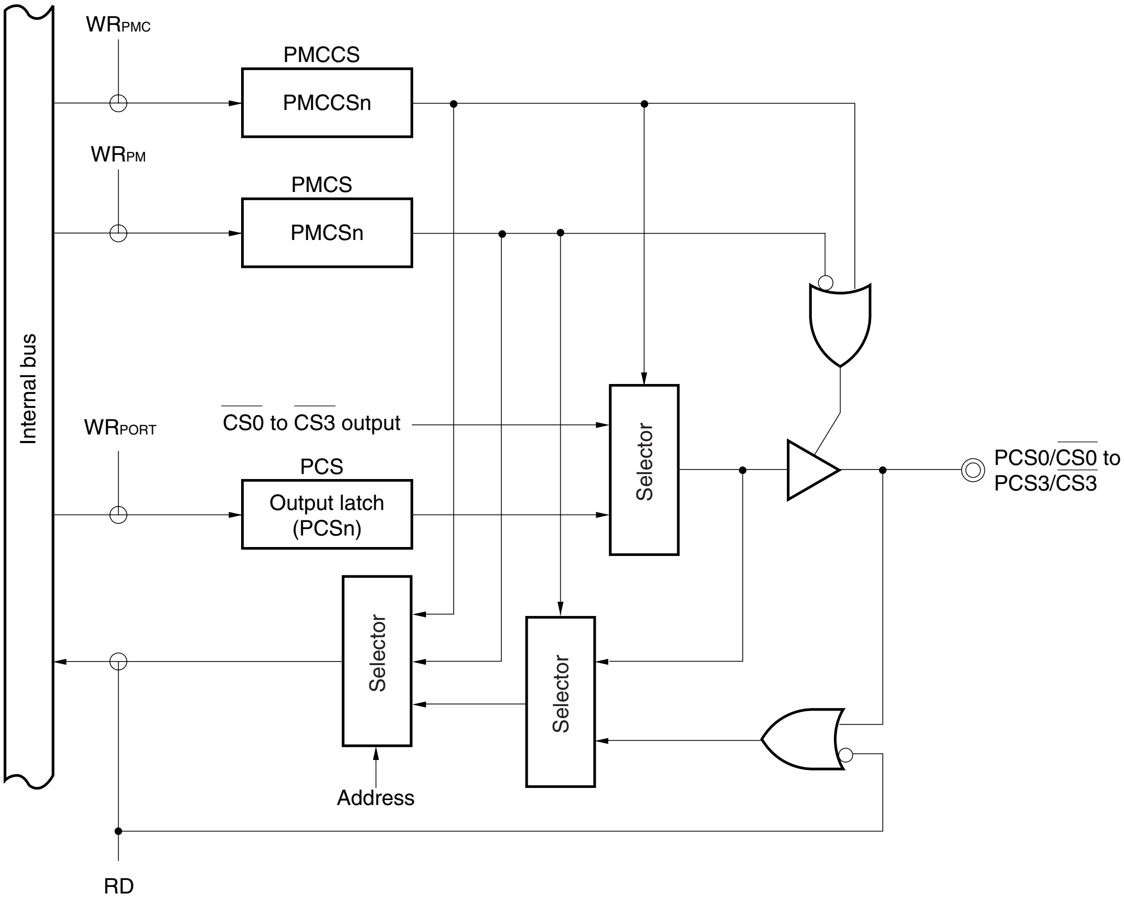
After reset: 00H R/W Address: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	PMCCS3	PMCCS2	PMCCS1	PMCCS0

PMCCSn	Specifies operation mode of PCSn pin (n = 0 to 3)
0	I/O port
1	$\overline{\text{CSn}}$ output

(3) Block diagram

Figure 4-28. Block Diagram of PCS0 to PCS3



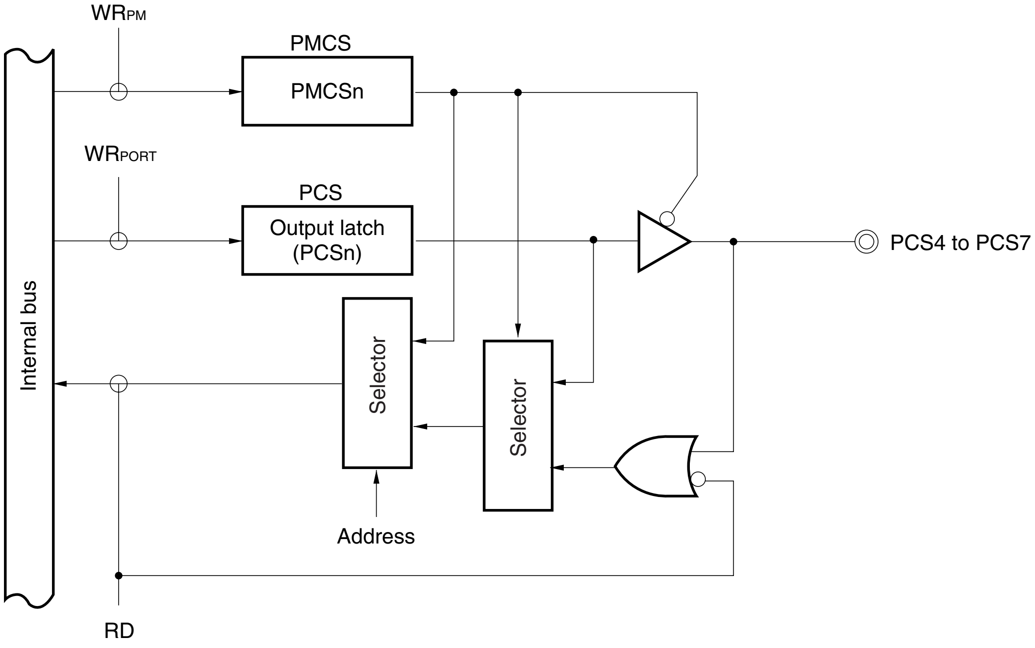
Remarks 1. PCS: Port register CS

PMCS: Port mode register CS

PMCCS: Port mode control register CS

2. n = 0 to 3

Figure 4-29. Block Diagram of PCS4 to PCS7



Remarks 1. PCS: Port register CS
PMCS: Port mode register CS

2. n = 4 to 7

4.3.11 Port CT

Port CT can be set to the input or output mode in 1-bit units.

The number of I/O port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	6-bit I/O port
V850ES/SA3	8-bit I/O port

(1) Functions of port CT

○ Input/output data can be specified in 1-bit units by using port register CT (PCT).

○ Can be set to the input or output mode in 1-bit units by using port mode register CT (PMCT).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register CT (PMCCT).

Table 4-15. Alternate-Function Pins of Port CT

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port CT	PCT0	$\overline{\text{WR0}}$	I/O	None	–
	PCT1	$\overline{\text{WR1}}$			
	PCT2[Note 2]	–			
	PCT3[Note 2]	–			
	PCT4	$\overline{\text{RD}}$			
	PCT5	–			
	PCT6	ASTB			
	PCT7	–			

Notes 1. Software pull-up function

2. V850ES/SA3 only

(2) Registers

(a) Port register CT (PCT)

Port register PCT (PCT) is an 8-bit register that controls reading the pin level and writing the output level.

This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	PCT7	PCT6	PCT5	PCT4	PCT3[Note]	PCT2[Note]	PCT1	PCT0

PCTn	Controls output data (in output mode) (V850ES/SA2: n = 0, 1, 4 to 7, V850ES/SA3: n = 0 to 7)
0	Outputs 0.
1	Outputs 1.

Note Bits 3 and 2 are provided in the V850ES/SA3 only. Be sure to clear these bits to 0 in the V850ES/SA2.

Remarks 1. In input mode: When port CT (PCT) is read, the pin level at that time is read. When written, the data written to PCT is written. The input pin is not affected.

In output mode: When port CT (PCT) is read, the value of PCT is read. When a value is written to PCT, it is immediately output.

2. After reset, an undefined value (pin input level) is read from PCT in the input mode. When PCT is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register CT (PMCT)

This is an 8-bit register that specifies the input or output mode.

This register can be read or written in 8-bit or 1-bit units.

After reset: FFH R/W Address: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	PMCT7	PMCT6	PMCT5	PMCT4	PMCT3[Note]	PMCT2[Note]	PMCT1	PMCT0

PMCTn	Controls input/output mode (V850ES/SA2: n = 0, 1, 4 to 7, V850ES/SA3: n = 0 to 7)
0	Output mode
1	Input mode

Note Bits 3 and 2 are provided in the V850ES/SA3 only. Be sure to set these bits to 1 in the V850ES/SA2.

(c) Port mode control register CT (PMCCT)

This is an 8-bit register that specifies the port mode or control mode.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	PMCCT6	0	PMCCT4	0	0	PMCCT1	PMCCT0

PMCCT6	Specifies operation mode of PCT6 pin
0	I/O port
1	ASTB output

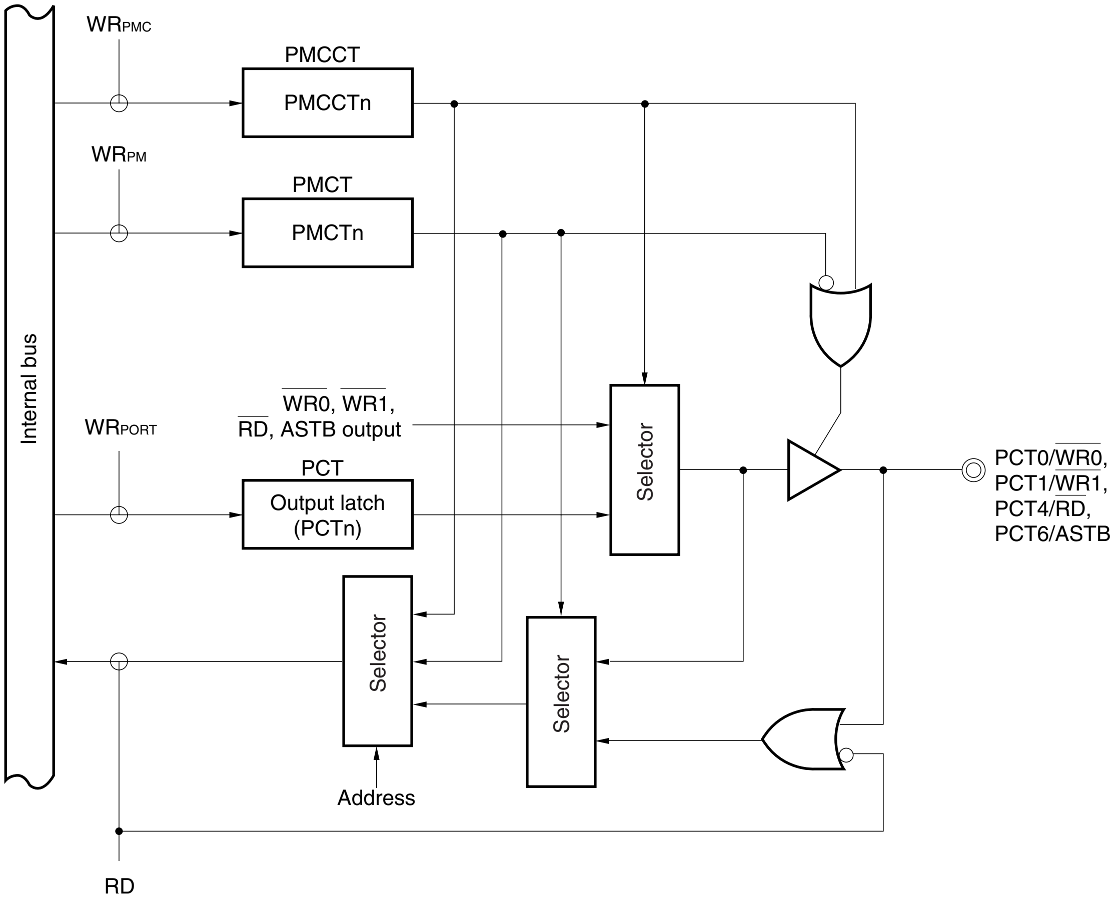
PMCCT4	Specifies operation mode of PCT4 pin
0	I/O port
1	$\overline{\text{RD}}$ output

PMCCT1	Specifies operation mode of PCT1 pin
0	I/O port
1	$\overline{\text{WR1}}$ output

PMCCT0	Specifies operation mode of PCT0 pin
0	I/O port
1	$\overline{\text{WR0}}$ output

(3) Block diagram

Figure 4-30. Block Diagram of PCT0, PCT1, PCT4, and PCT6



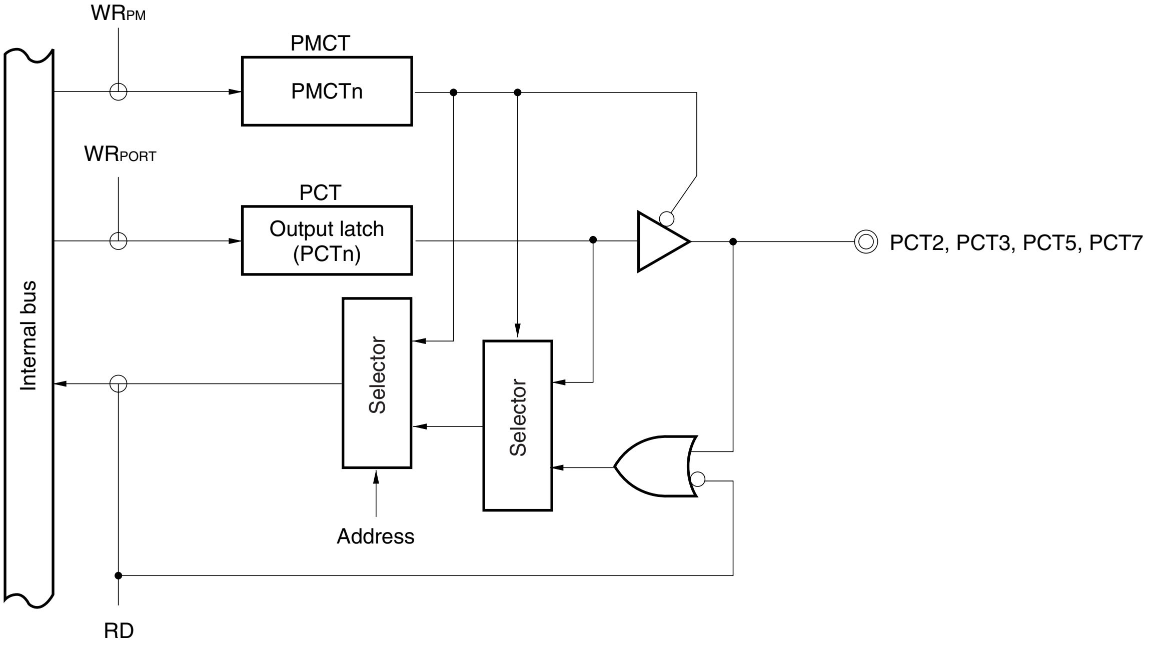
Remarks 1. PCT: Port register CT

PMCT: Port mode register CT

PMCCT: Port mode control register CT

2. n = 0, 1, 4, or 6

Figure 4-31. Block Diagram of PCT2, PCT3, PCT5, and PCT7



Remarks 1. PCT: Port register CT

PMCT: Port mode register CT

2. n = 2, 3, 5, or 7

4.3.12 Port DH

Port DH can be set to the input or output mode in 1-bit units.

The number of I/O port bits differs depending on the product.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	6-bit I/O port
V850ES/SA3	8-bit I/O port

(1) Functions of port DH

○ Input/output data can be specified in 1-bit units by using port register DH (PDH).

○ Can be set to the input or output mode in 1-bit units by using port mode register DH (PMDH).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register DH (PMCDH).

Port DH has an alternate function as the following pins.

Table 4-16. Alternate-Function Pins of Port DH

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port DH	PDH0	A16	I/O	None	–
	PDH1	A17			
	PDH2	A18			
	PDH3	A19			
	PDH4	A20			
	PDH5	A21			
	PDH6[Note 2]	A22[Note 2]			
	PDH7[Note 2]	A23[Note 2]			

Notes 1. Software pull-up function

2. V850ES/SA3 only

(2) Registers

(a) Port register DH (PDH)

Port register PDH (PDH) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

After reset: Undefined   R/W   Address: FFFFF006H

	7	6	5	4	3	2	1	0
PDH	PDH7[Note]	PDH6[Note]	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0

PDHn	Controls output data (in output mode) (V850ES/SA2: n = 0 to 5, V850ES/SA3: n = 0 to 7)
0	Outputs 0.
1	Outputs 1.

Note Bits 7 and 6 are provided in the V850ES/SA3 only. Be sure to clear these bits to 0 in the V850ES/SA2.

Remarks 1. In input mode: When port DH (PDH) is read, the pin level at that time is read. When written, the data written to PDH is written. The input pin is not affected.

In output mode: When port DH (PDH) is read, the value of PDH is read. When a value is written to PDH, it is immediately output.

2. After reset, an undefined value (pin input level) is read from PDH in the input mode. When PDH is read in the output mode, 00H (value of the output latch) is read.

(b) Port mode register DH (PMDH)

This is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.

After reset: FFH   R/W   Address: FFFFF026H

	7	6	5	4	3	2	1	0
PMDH	PMDH7[Note]	PMDH6[Note]	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0

PMDHn	Controls input/output mode (V850ES/SA2: n = 0 to 5, V850ES/SA3: n = 0 to 7)
0	Output mode
1	Input mode

Note Bits 7 and 6 are provided in the V850ES/SA3 only. Be sure to set these bits to 1 in the V850ES/SA2.

(c) Port mode control register DH (PMCDH)

This is an 8-bit register that specifies the port mode or control mode.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF046H

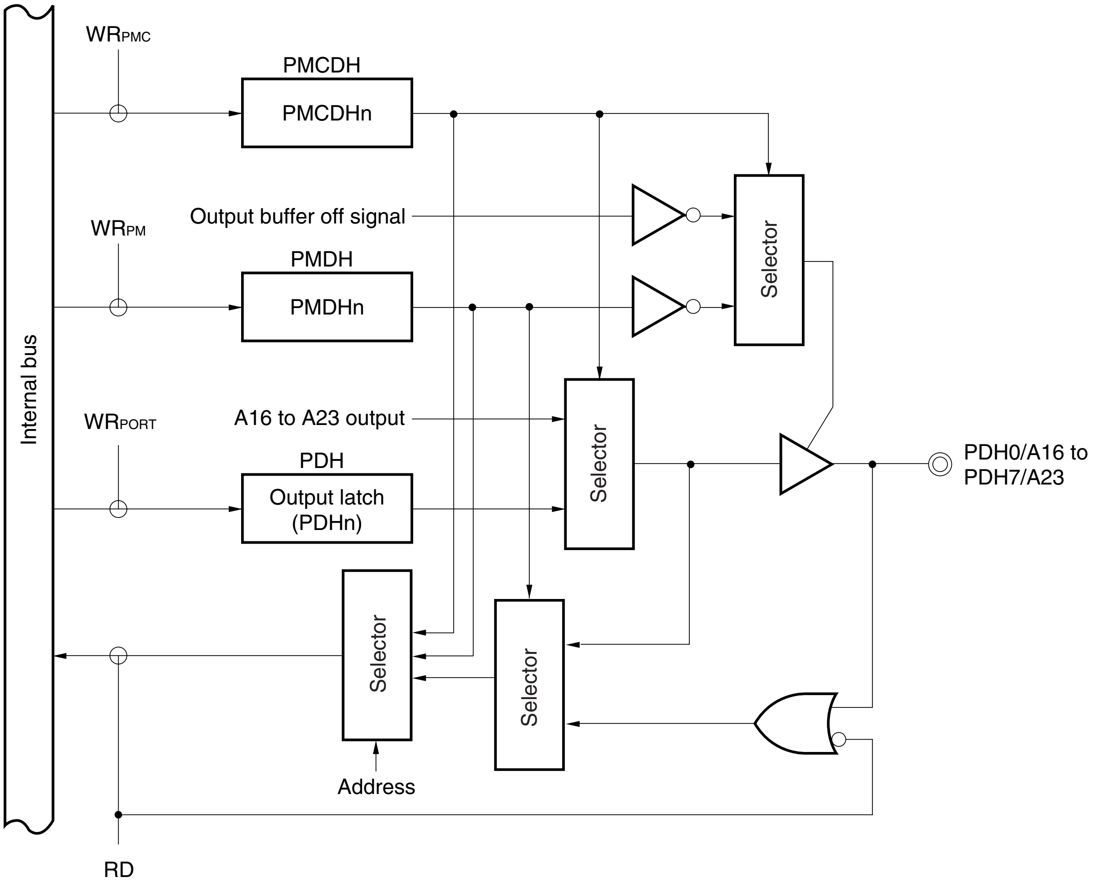
	7	6	5	4	3	2	1	0
PMCDH	PMCDH7[Note]	PMCDH6[Note]	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0

PMCDHn	Specifies operation mode of PDHn pin (V850ES/SA2: n = 0 to 5, V850ES/SA3: n = 0 to 7)
0	I/O port
1	Am output (address bus output) (V850ES/SA2: m = 16 to 21, V850ES/SA3: m = 16 to 23)

Note Bits 7 and 6 are provided in the V850ES/SA3 only. Be sure to clear these bits to 0 in the V850ES/SA2.

(3) Block diagram

★ **Figure 4-32. Block Diagram of PDH0 to PDH7**



Remarks 1. PDH: Port register DH

PMDH: Port mode register DH

PMCDH: Port mode control register DH

PUDH: Pull-up resistor option register DH

Output buffer off signal: Signal that is active in IDLE/STOP mode

2. n = 0 to 7

4.3.13 Port DL

Port DL can be set to the input or output mode in 1-bit units.

The number of I/O port bits of each product is the same.

Commercial Name	Number of I/O Port Bits
V850ES/SA2	16-bit I/O port
V850ES/SA3	16-bit I/O port

(1) Functions of port DL

○ Input/output data can be specified in 1-bit units by using port register DL (PDL).

○ Can be set to the input or output mode in 1-bit units by using port mode register DL (PMDL).

○ Can be set to the port mode or control mode (alternate function) in 1-bit units by using port mode control register DL (PMCDL).

Port DL has an alternate function as the following pins.

Table 4-17. Alternate-Function Pins of Port DL

Pin Name		Alternate-Function Pin	I/O	PULL[Note 1]	Remark
Port DL	PDL0	AD0	I/O	None	–
	PDL1	AD1			
	PDL2	AD2			
	PDL3	AD3			
	PDL4	AD4			
	PDL5	AD5/FLMD1[Note 2]			
	PDL6	AD6			
	PDL7	AD7			
	PDL8	AD8			
	PDLDL	AD9			
	PDL10	AD10			
	PDL11	AD11			
	PDL12	AD12			
	PDL13	AD13			
	PDL14	AD14			
	PDL15	AD15			

Notes 1. Software pull-up function

2. Because these pins are used for setting in the flash programming mode, they do not have to be manipulated by using a port control register. For details, refer to **CHAPTER 21 FLASH MEMORY** (μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y only).

(2) Registers

(a) Port register DL (PDL)

Port register DL (PDL) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

If the higher 8 bits of the PDL register are used as PDLH, and the lower 8 bits as PDLL, however, PDLH and PDLL can be used as an 8-bit I/O port whose input or output can be manipulated in 8-bit or 1-bit units.

After reset: Undefined R/W Address: PDL FFFFF004H,
PDLL FFFFF004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH[Note])	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8

	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Controls output data (in output mode) (n = 0 to 15)
0	Outputs 0.
1	Outputs 1.

Note When reading or writing bits 8 to 15 of the PDL register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PDLH register.

Remarks 1. In input mode: When port DL (PDL) is read, the pin level at that time is read. When written, the data written to PDL is written. The input pin is not affected.

In output mode: When port DL (PDL) is read, the value of PDL is read. When a value is written to PDL, it is immediately output.

2. After reset, an undefined value (pin input level) is read from PDL in the input mode. When PDL is read in the output mode, 0000H (value of the output latch) is read.

(b) Port mode register DL (PMDL)

This is a 16-bit register that specifies the input or output mode.

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

If the higher 8 bits of the PMDL register are used as PMDLH, and the lower 8 bits as PMDLL, however, PMDLH and PMDLL can be read or written in 8-bit or 1-bit units.

After reset: FFFFH R/W Address: PMDL FFFFF024H,
PMDLL FFFFF024H, PMDLH FFFFF025H

	15	14	13	12	11	10	9	8
PMDLH (PMDLH[Note])	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8

	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	Controls input/output mode (n = 0 to 15)
0	Output mode
1	Input mode

Note When reading or writing bits 8 to 15 of the PMDL register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMDLH register.

(c) Port mode control register DL (PMCDL)

This is a 16-bit register that specifies the port mode or control mode.

It can be read or written only in 16-bit units.

If the higher 8 bits of the PMCDL register are used as PMCDLH, and the lower 8 bits as PMCDLL, however, PMCDLH and PMCDLL can be read or written in 8-bit units.

After reset: 0000H R/W Address: PMCDL FFFFF044H,
PMCDLL FFFFF044H, PMCDLH FFFFF045H

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH[Note])	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

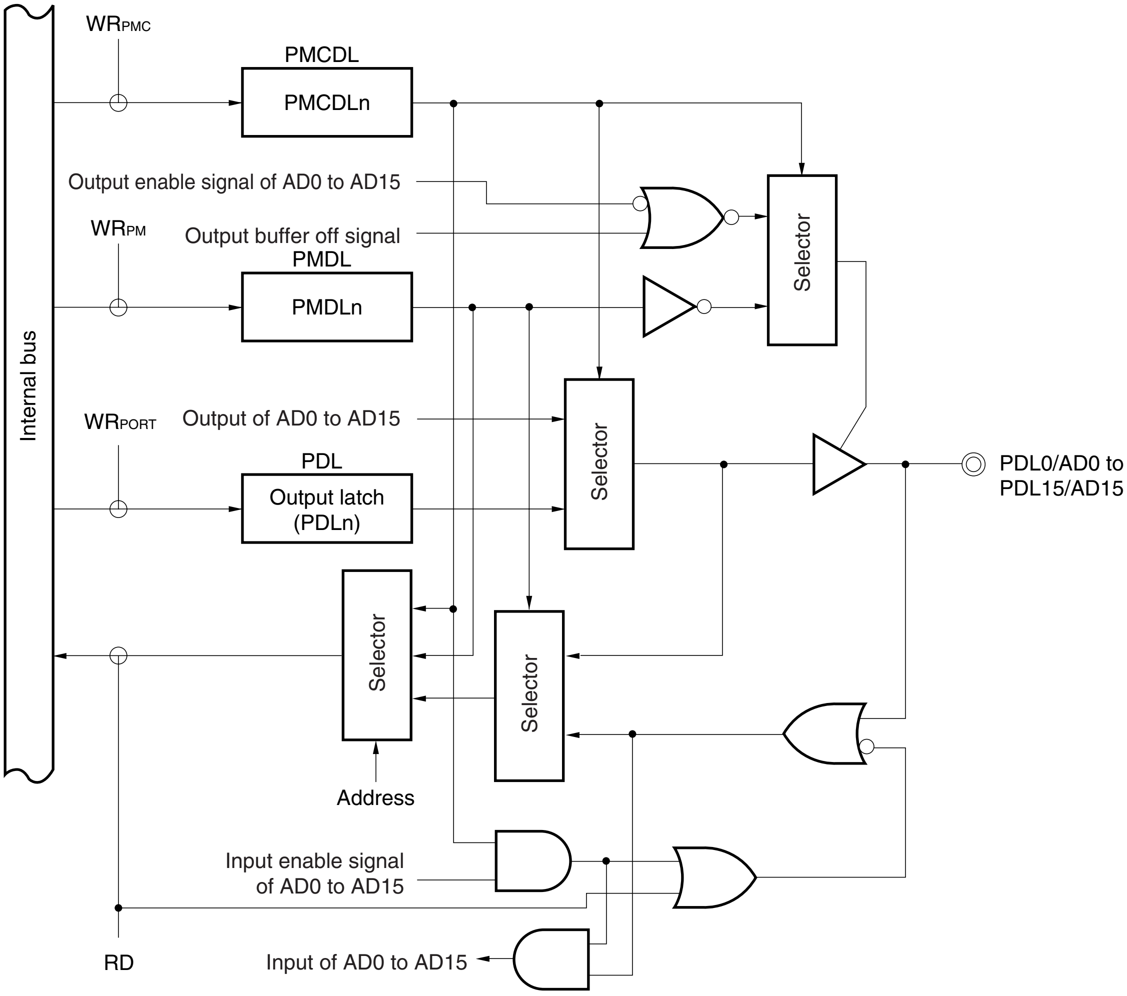
PMCDLn	Specifies operation mode of PDLn pin (n = 0 to 15)
0	I/O port
1	ADn output (address/data bus input/output)

Note When reading or writing bits 8 to 15 of the PMCDL register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMCDLH register.

Caution Do not specify AD8 to AD15 when the SMSEL bit of the EXIMC register = 1 (separate mode) and when the BS30 to BS00 bits of the BSC register = 0 (8-bit bus width).

(3) Block diagram

Figure 4-33. Block Diagram of PDL0 to PDL15



Remarks 1. PDL: Port register DL

PMDL: Port mode register DL

PMCDL: Port mode control register DL

Output buffer off signal: Signal that is active in IDLE/STOP mode

2. n = 0 to 15

Table 4-18. Using Alternate Function of Port Pins (1/6)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Register)
	Name	I/O					
P00	NMI	Input	P00 = Setting not needed	PM00 = Setting not needed	PMC00 = 1	–	INTR00 (INTR0), INTF00 (INTF0)
P01	INTP0	Input	P01 = Setting not needed	PM01 = Setting not needed	PMC01 = 1	–	INTR01 (INTR0), INTF01 (INTF0)
	TI2	Input	P01 = Setting not needed	PM01 = Setting not needed	PMC01 = 1	–	–
P02	INTP1	Input	P02 = Setting not needed	PM02 = Setting not needed	PMC02 = 1	–	INTR02 (INTR0), INTF02 (INTF0)
	TI3	Input	P02 = Setting not needed	PM02 = Setting not needed	PMC02 = 1	–	–
P03	INTP2	Input	P03 = Setting not needed	PM03 = Setting not needed	PMC03 = 1	–	INTR03 (INTR0), INTF00 (INTF0)
	TI4	Input	P03 = Setting not needed	PM03 = Setting not needed	PMC03 = 1	–	–
P04	INTP3	Input	P04 = Setting not needed	PM04 = Setting not needed	PMC04 = 1	–	INTR04 (INTR0), INTF04 (INTF0)
	TI5	Input	P04 = Setting not needed	PM04 = Setting not needed	PMC04 = 1	–	–
P05	INTP4	Input	P05 = Setting not needed	PM05 = Setting not needed	PMC05 = 1	–	INTR05 (INTR0), INTF05 (INTF0)
P20	SI4	Input	P20 = Setting not needed	PM20 = Setting not needed	PMC20 = 1	–	–
P21	SO4	Output	P21 = Setting not needed	PM21 = Setting not needed	PMC21 = 1	–	PF21 = don’t care (PF2)
P22	$\overline{\text{SCK4}}$	I/O	P22 = Setting not needed	PM22 = Setting not needed	PMC22 = 1	–	PF22 = don’t care (PF2)
P30	SI1	Input	P30 = Setting not needed	PM30 = Setting not needed	PMC30 = 1	PFC30 = 0	–
	RXD0	Input	P30 = Setting not needed	PM30 = Setting not needed	PMC30 = 1	PFC30 = 1	–
P31	SO1	Output	P31 = Setting not needed	PM31 = Setting not needed	PMC31 = 1	PFC31 = 0	PF31 = 0 (PF3)
	TXD0	Output	P31 = Setting not needed	PM31 = Setting not needed	PMC31 = 1	PFC31 = 1	PF31 = don’t care (PF3)
P32	$\overline{\text{SCK1}}$	I/O	P32 = Setting not needed	PM32 = Setting not needed	PMC32 = 1	–	PF32 = don’t care (PF3)
P40	SI0	Input	P40 = Setting not needed	PM40 = Setting not needed	PMC40 = 1	–	–
P41	SO0	Output	P41 = Setting not needed	PM41 = Setting not needed	PMC41 = 1	PFC41 = 0	PF41 = don't care (PF4)
	SDA[Note]	I/O	P41 = Setting not needed	PM41 = Setting not needed	PMC41 = 1	PFC41 = 1	PF41 = 1 (PF4)
P42	$\overline{\text{SCK0}}$	I/O	P42 = Setting not needed	PM42 = Setting not needed	PMC42 = 1	PFC42 = 0	PF42= don’t care (PF4)
	SCL[Note]	I/O	P42 = Setting not needed	PM42 = Setting not needed	PMC42 = 1	PFC42 = 1	PF42 = 1 (PF4)

Note μPD703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Table 4-18. Using Alternate Function of Port Pins (2/6)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Register)
	Name	I/O					
P43	INTP00	Input	P43 = Setting not needed	PM43 = Setting not needed	PMC43 = 1	–	ETI0 = 0 (TMC01), **Note**
	TI0	Input	P43 = Setting not needed	PM43 = Setting not needed	PMC43 = 1	–	ETI0 = 1 (TMC01), **Note**
	TCLR0	Input	P43 = Setting not needed	PM43 = Setting not needed	PMC43 = 1	–	ECLR0 = 1 (TMC01), **Note**
P44	INTP01	Input	P44 = Setting not needed	PM44 = Setting not needed	PMC44 = 1	PFC44 = 0	–
	TO0	Output	P44 = Setting not needed	PM44 = Setting not needed	PMC44 = 1	PFC44 = 1	–
P45	INTP10	Input	P45 = Setting not needed	PM45 = Setting not needed	PMC45 = 1	–	ETI1 = 0 (TMC11), **Note**
	TI1	Input	P45 = Setting not needed	PM45 = Setting not needed	PMC45 = 1	–	ETI1 = 1 (TMC11), **Note**
	TCLR1	Input	P45 = Setting not needed	PM45 = Setting not needed	PMC45 = 1	–	ECLR1 = 1 (TMC11), **Note**
P46	INTP11	Input	P46 = Setting not needed	PM46 = Setting not needed	PMC46 = 1	PFC46 = 0	–
	TO1	Output	P46 = Setting not needed	PM46 = Setting not needed	PMC46 = 1	PFC46 = 1	–
P70	ANI0	Input	P70 = Setting impossible	–	–	–	–
P71	ANI1	Input	P71 = Setting impossible	–	–	–	–
P72	ANI2	Input	P72 = Setting impossible	–	–	–	–
P73	ANI3	Input	P73 = Setting impossible	–	–	–	–
P74	ANI4	Input	P74 = Setting impossible	–	–	–	–
P75	ANI5	Input	P75 = Setting impossible	–	–	–	–
P76	ANI6	Input	P76 = Setting impossible	–	–	–	–
P77	ANI7	Input	P77 = Setting impossible	–	–	–	–
P78	ANI8	Input	P78 = Setting impossible	–	–	–	–
P79	ANI9	Input	P79 = Setting impossible	–	–	–	–
P710	ANI10	Input	P710 = Setting impossible	–	–	–	–

Note Set the valid edge by using the timer n valid edge selection register (SESn).

Table 4-18. Using Alternate Function of Port Pins (3/6)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Register)
	Name	I/O					
P711	ANI11	Input	P711 = Setting impossible	–	–	–	–
P712	ANI12	Input	P712 = Setting impossible	–	–	–	–
P713	ANI13	Input	P713 = Setting impossible	–	–	–	–
P714	ANI14	Input	P714 = Setting impossible	–	–	–	–
P715	ANI15	Input	P715 = Setting impossible	–	–	–	–
P80	ANO0	Output	P80 = Setting impossible	–	–	–	–
P81	ANO1	Output	P780 = Setting impossible	–	–	–	–
P90	A0	Output	P90 = Setting not needed	PM90 = Setting not needed	PMC90 = 1	PFC90 = 0	**Note**
P91	A1	Output	P91 = Setting not needed	PM91 = Setting not needed	PMC91 = 1	PFC91 = 0	–
P92	A2	Output	P92 = Setting not needed	PM92 = Setting not needed	PMC92 = 1	PFC92 = 0	**Note**
	INTP5	Input	P92 = Setting not needed	PM92 = Setting not needed	PMC92 = 1	PFC92 = 1	INTR92 (INTR9), INTF92 (INTF9)
P93	A3	Output	P93 = Setting not needed	PM93 = Setting not needed	PMC93 = 1	PFC93 = 0	**Note**
	INTP6	Input	P93 = Setting not needed	PM93 = Setting not needed	PMC93 = 1	PFC93 = 1	INTR93 (INTR9), INTF93 (INTF9)
P94	A4	Output	P94 = Setting not needed	PM94 = Setting not needed	PMC94 = 1	PFC94 = 0	**Note**
	TO2	Output	P94 = Setting not needed	PM94 = Setting not needed	PMC94 = 1	PFC94 = 1	–
P95	A5	Output	P95 = Setting not needed	PM95 = Setting not needed	PMC95 = 1	PFC95 = 0	**Note**
	TO3	Output	P95 = Setting not needed	PM95 = Setting not needed	PMC95 = 1	PFC95 = 1	–
P96	A6	Output	P96 = Setting not needed	PM96 = Setting not needed	PMC96 = 1	PFC96 = 0	**Note**
	TO4	Output	P96 = Setting not needed	PM96 = Setting not needed	PMC96 = 1	PFC96 = 1	–
P97	A7	Output	P97 = Setting not needed	PM97 = Setting not needed	PMC97 = 1	PFC97 = 0	**Note**
	TO5	Output	P97 = Setting not needed	PM97 = Setting not needed	PMC97 = 1	PFC97 = 1	–
P98	A8	Output	P98 = Setting not needed	PM98 = Setting not needed	PMC98 = 1	PFC98 = 0	**Note**
	RXD1	Input	P98 = Setting not needed	PM98 = Setting not needed	PMC98 = 1	PFC98 = 1	–

Note To set the A0 to A15 pins, clear the PFC9 register to 0000H and set the PMC9 register to FFFFH in 16-bit units.

Table 4-18. Using Alternate Function of Port Pins (4/6)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Register)
	Name	I/O					
P99	A9	Output	P99 = Setting not needed	PM99 = Setting not needed	PMC99 = 1	PFC99 = 0	**Note**
	TXD1	Output	P99 = Setting not needed	PM99 = Setting not needed	PMC99 = 1	PFC99 = 1	–
P910	A10	Output	P910 = Setting not needed	PM910 = Setting not needed	PMC910 = 1	PFC910 = 0	**Note**
	SI2	Input	P910 = Setting not needed	PM910 = Setting not needed	PMC910 = 1	PFC910 = 1	–
P911	A11	Output	P911 = Setting not needed	PM911 = Setting not needed	PMC911 = 1	PFC911 = 0	**Note**
	SO2	Output	P911 = Setting not needed	PM911 = Setting not needed	PMC911 = 1	PFC911 = 1	PF911 = don't care (PF9)
P912	A12	Output	P912 = Setting not needed	PM912 = Setting not needed	PMC912 = 1	PFC912 = 0	**Note**
	$\overline{\text{SCK2}}$	Output	P912 = Setting not needed	PM912 = Setting not needed	PMC912 = 1	PFC912 = 1	PF912 = don't care (PF9)
P913	A13	Output	P913 = Setting not needed	PM913 = Setting not needed	PMC913 = 1	PFC913 = 0	**Note**
	SI3	Input	P913 = Setting not needed	PM913 = Setting not needed	PMC913 = 1	PFC913 = 1	–
P914	A14	Output	P914 = Setting not needed	PM914 = Setting not needed	PMC914 = 1	PFC914 = 0	**Note**
	SO3	Output	P914 = Setting not needed	PM914 = Setting not needed	PMC914 = 1	PFC914 = 1	PF914 = don't care (PF9)
P915	A15	Output	P915 = Setting not needed	PM915 = Setting not needed	PMC915 = 1	PFC915 = 0	**Note**
	$\overline{\text{SCK3}}$	Output	P915 = Setting not needed	PM915 = Setting not needed	PMC915 = 1	PFC915 = 1	PF915 = don't care (PF9)

Note To set the A0 to A15 pins, clear the PFC9 register to 0000H and set the PMC9 register to FFFFH in 16-bit units.

Table 4-18. Using Alternate Function of Port Pins (5/6)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Register)
	Name	I/O					
PCM0	$\overline{\text{WAIT}}$	Input	PCM0 = Setting not needed	PMCM0 = Setting not needed	PMCCM0 = 1	–	–
PCM1	CLKOUT	Output	PCM1 = Setting not needed	PMCM1 = Setting not needed	PMCCM1 = 1	–	–
PCM2	$\overline{\text{HLDAK}}$	Output	PCM2 = Setting not needed	PMCM2 = Setting not needed	PMCCM2 = 1	–	–
PCM3	$\overline{\text{HLDQR}}$	Input	PCM3 = Setting not needed	PMCM3 = Setting not needed	PMCCM3 = 1	–	–
PCS0	$\overline{\text{CS0}}$	Output	PCS0 = Setting not needed	PMCS0 = Setting not needed	PMCCS0 = 1	–	–
PCS1	$\overline{\text{CS1}}$	Output	PCS1 = Setting not needed	PMCS1 = Setting not needed	PMCCS1 = 1	–	–
PCS2	$\overline{\text{CS2}}$	Output	PCS2 = Setting not needed	PMCS2 = Setting not needed	PMCCS2 = 1	–	–
PCS3	$\overline{\text{CS3}}$	Output	PCS3 = Setting not needed	PMCS3 = Setting not needed	PMCCS3 = 1	–	–
PCT0	$\overline{\text{WR0}}$	Output	PCT0 = Setting not needed	PMCT0 = Setting not needed	PMCCT0 = 1	–	–
PCT1	$\overline{\text{WR1}}$	Output	PCT1 = Setting not needed	PMCT1 = Setting not needed	PMCCT1 = 1	–	–
PCT4	$\overline{\text{RD}}$	Output	PCT4 = Setting not needed	PMCT4 = Setting not needed	PMCCT4 = 1	–	–
PCT6	ASTB	Output	PCT6 = Setting not needed	PMCT6 = Setting not needed	PMCCT6 = 1	–	–

Table 4-18. Using Alternate Function of Port Pins (6/6)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCnx Bit of PFCn Register	Other Bits (Register)
	Name	I/O					
PDH0	A16	Output	PDH0 = Setting not needed	PMDH0 = Setting not needed	PMCDH0 = 1	–	–
PDH1	A17	Output	PDH1 = Setting not needed	PMDH1 = Setting not needed	PMCDH1 = 1	–	–
PDH2	A18	Output	PDH2 = Setting not needed	PMDH2 = Setting not needed	PMCDH2 = 1	–	–
PDH3	A19	Output	PDH3 = Setting not needed	PMDH3 = Setting not needed	PMCDH3 = 1	–	–
PDH4	A20	Output	PDH4 = Setting not needed	PMDH4 = Setting not needed	PMCDH4 = 1	–	–
PDH5	A21	Output	PDH5 = Setting not needed	PMDH5 = Setting not needed	PMCDH5 = 1	–	–
PDH6	A22	Output	PDH6 = Setting not needed	PMDH6 = Setting not needed	PMCDH6 = 1	–	–
PDH7	A23	Output	PDH7 = Setting not needed	PMDH7 = Setting not needed	PMCDH7 = 1	–	–
PDL0	AD0	I/O	PDL0 = Setting not needed	PMDL0 = Setting not needed	PMCDL0 = 1	–	–
PDL1	AD1	I/O	PDL1 = Setting not needed	PMDL1 = Setting not needed	PMCDL1 = 1	–	–
PDL2	AD2	I/O	PDL2 = Setting not needed	PMDL2 = Setting not needed	PMCDL2 = 1	–	–
PDL3	AD3	I/O	PDL3 = Setting not needed	PMDL3 = Setting not needed	PMCDL3 = 1	–	–
PDL4	AD4	I/O	PDL4 = Setting not needed	PMDL4 = Setting not needed	PMCDL4 = 1	–	–
PDL5	AD5	I/O	PDL5 = Setting not needed	PMDL5 = Setting not needed	PMCDL5 = 1	–	–
PDL6	AD6	I/O	PDL6 = Setting not needed	PMDL6 = Setting not needed	PMCDL6 = 1	–	–
PDL7	AD7	I/O	PDL7 = Setting not needed	PMDL7 = Setting not needed	PMCDL7 = 1	–	–
PDL8	AD8	I/O	PDL8 = Setting not needed	PMDL8 = Setting not needed	PMCDL8 = 1	–	–
PDL9	AD9	I/O	PDL9 = Setting not needed	PMDL9 = Setting not needed	PMCDL9 = 1	–	–
PDL10	AD10	I/O	PDL10 = Setting not needed	PMDL10 = Setting not needed	PMCDL10 = 1	–	–
PDL11	AD11	I/O	PDL11 = Setting not needed	PMDL11 = Setting not needed	PMCDL11 = 1	–	–
PDL12	AD12	I/O	PDL12 = Setting not needed	PMDL12 = Setting not needed	PMCDL12 = 1	–	–
PDL13	AD13	I/O	PDL13 = Setting not needed	PMDL13 = Setting not needed	PMCDL13 = 1	–	–
PDL14	AD14	I/O	PDL14 = Setting not needed	PMDL14 = Setting not needed	PMCDL14 = 1	–	–
PDL15	AD15	I/O	PDL15 = Setting not needed	PMDL15 = Setting not needed	PMCDL15 = 1	–	–

4.4 Cautions

The operation of a port differs depending on whether the port is in the input or output mode, as described below.

4.4.1 Writing data to I/O port

(1) In output mode

A value can be written to the output latch by using a transfer instruction. The contents of the output latch are output from the pin. Once data has been written to the output latch, it is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. Because the output buffer is off, however, the status of the pin does not change.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

Caution A 1-bit memory manipulation instruction manipulates 1 bit but accesses a port in 8-bit units. If this instruction is executed to manipulate a port with a mixture of input and output bits, the contents of the output latch of a pin set in the input mode, in addition to the bit to be manipulated, become undefined.

4.4.2 Reading data from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch do not change.

(2) In input mode

The status of the pin can be read by using a transfer instruction. The contents of the output latch do not change.

CHAPTER 5 BUS CONTROL FUNCTION

The V850ES/SA2 and V850ES/SA3 are provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

5.1 Features

○ Output is selectable from a multiplexed bus with a minimum of 3 bus cycles and a separate bus with a minimum of 2 bus cycles.

○ Four-space chip select signal output function

○ 8-bit/16-bit data bus selectable (for each area selected by chip select function)

○ Wait function

- Programmable wait function of up to 7 states (selectable for each area selected by chip select function)
- External wait function using $\overline{\text{WAIT}}$ pin

○ Idle state function

○ Bus hold function

5.2 Bus Control Pins

The pins used to connect an external device are listed in the table below.

Table 5-1. Bus Control Pins (Multiplexed Bus)

Bus Control Pin	Alternate-Function Pin	I/O	Function
AD0 to AD15	PDL0 to PDL15	I/O	Address/data bus
A16 to A23[Note]	PDH0 to PDH7	Output	Address bus
$\overline{\text{WAIT}}$	PCM0	Input	External wait control
CLKOUT	PCM1	Output	Internal system clock
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	PCS0 to PCS3	Output	Chip select signal
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	PCT0, PCT1	Output	Write strobe signal
$\overline{\text{RD}}$	PCT4	Output	Read strobe signal
ASTB	PCT6	Output	Address strobe signal
$\overline{\text{HLDRQ}}$	PCM3	Input	Bus hold control
$\overline{\text{HLDAK}}$	PCM2	Output	

Note A16 to A21 in the V850ES/SA2

Table 5-2. External Control Pins (Separate Bus)

Bus Control Pin	Alternate-Function Pin	I/O	Function
AD0 to AD15	PDL0 to PDL15	I/O	Data bus
A0 to A15	P90 to P915	Output	Address bus
A16 to A23[Note]	PDH0 to PDH7	Output	Address bus
$\overline{\text{WAIT}}$	PCM0	Input	External wait control
CLKOUT	PCM1	Output	Internal system clock
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	PCS0 to PCS3	Output	Chip select
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	PCT0, PCT1	Output	Write strobe signal
$\overline{\text{RD}}$	PCT4	Output	Read strobe signal
$\overline{\text{HLDRQ}}$	PCM3	Input	Bus hold control
$\overline{\text{HLDAK}}$	PCM2	Output	

Note A16 to A21 in the V850ES/SA2

★ 5.2.1 Pin status when internal ROM, internal RAM, or internal peripheral I/O is accessed

When the internal ROM, internal RAM, or internal peripheral I/O is accessed, the status of each pin is as follows.

Table 5-3. Pin Status When Internal ROM, Internal RAM, or Internal Peripheral I/O Is Accessed

Separate Mode		Multiplexed Bus Mode	
Address bus (A23 to A0)	Undefined	Address bus (A23 to A16)	Undefined
Data bus (AD15 to AD0)	Hi-Z	Data bus (AD15 to AD0)	Undefined
Control signal	Inactive	Control signal	Inactive

Caution When the internal ROM area is write-accessed, the addresses, data, and control signals are activated in the same way as access to the external memory area.

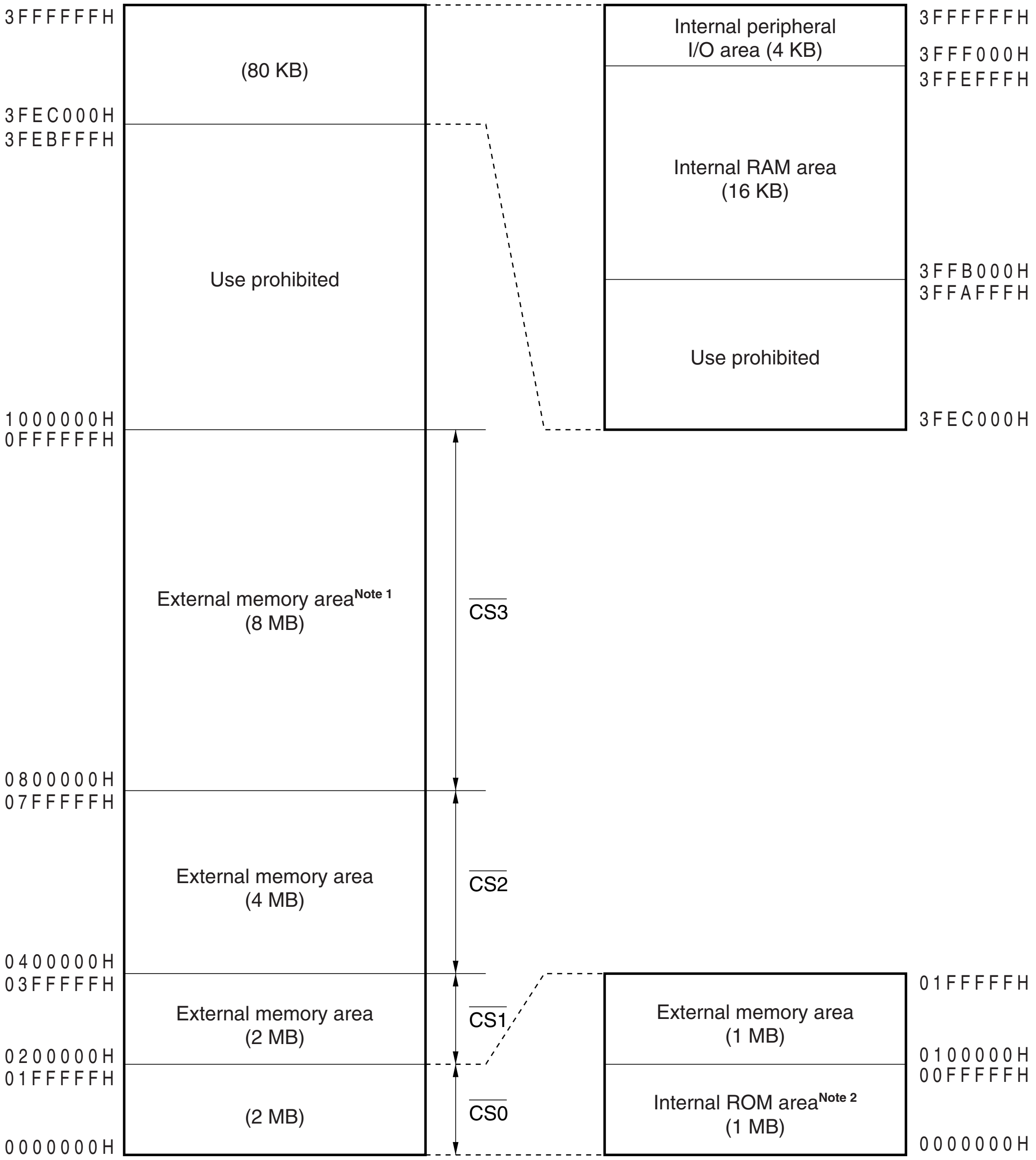
5.2.2 Pin status in each operation mode

For the pin status of the V850ES/SA2 and V850ES/SA3 in each operation mode, refer to **2.2 Pin Status**.

5.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of (lower) 2 MB, 2MB, 4MB, and 8MB. The programmable wait function and bus cycle operation mode for each of these blocks can be independently controlled in one-block units.

Figure 5-1. Data Memory Map



Notes 1. This area is the 4 MB space of 0800000H to 0BFFFFFH in the V850ES/SA2 (0C00000H to 0FFFFFFH are the image of 0800000H to 0BFFFFFH).

2. This area is an external memory area in the case of a data write access.

5.3.1 Chip select control function

Of the 64 MB (linear) address space, the lower 16 MB (0000000H to 0FFFFFFH) include four chip select functions, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$. The areas that can be selected by $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ are fixed.

By using these chip select functions, the memory block can be divided to enable effective use of the memory space. The allocation of the memory blocks is shown in the table below.

	V850ES/SA2	V850ES/SA3
$\overline{\text{CS0}}$	0000000H to 01FFFFFH (2 MB)	0000000H to 01FFFFFH (2 MB)
$\overline{\text{CS1}}$	0200000H to 03FFFFFH (2 MB)	0200000H to 03FFFFFH (2 MB)
$\overline{\text{CS2}}$	0400000H to 07FFFFFH (4 MB)	0400000H to 07FFFFFH (4 MB)
$\overline{\text{CS3}}$	0800000H to 0BFFFFFH (4 MB)	0800000H to 0FFFFFFH (8 MB)

5.4 External Bus Interface Mode Control Function

The V850ES/SA2 and V850ES/SA3 include the following two external bus interface modes.

- Multiplexed bus mode
- Separate bus mode

These two modes can be selected by using the external bus interface mode control register (EXIMC).

(1) External bus interface mode control register (EXIMC)

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFFFBEH

	7	6	5	4	3	2	1	0
EXIMC	0	0	0	0	0	0	0	SMSEL

SMSEL	Mode selection
0	Multiplexed bus mode
1	Separate bus mode

5.5 Bus Access

5.5.1 Number of clocks for access

The following table shows the number of basic clocks required for accessing each resource.

Area (Bus Width) Bus Cycle Type	Internal ROM (32 Bits)	Internal RAM (32 Bits)	External Memory (16 Bits)
Instruction fetch (normal access)	1	1 or 2	3 + n[Note]
Instruction fetch (branch)	2	1 or 2	3 + n[Note]
Operand data access	3	1	3 + n[Note]

Note 2 + n clocks (n: Number of wait states) when the separate bus mode is selected.

Remark Unit: Clocks/access

5.5.2 Bus size setting function

The bus size of each external memory area selected by $\overline{\text{CSn}}$ can be set (to 8 bits or 16 bits) by using the BSC register.

The external memory area of the V850ES/SA2 (0100000H to 0BFFFFFH) is selected by $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$.

The external memory area of the V850ES/SA3 (0100000H to 0FFFFFFH) is selected by $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$.

(1) Bus size configuration register (BSC)

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

Caution Write to the BSC register after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the BSC register are complete. However, external memory areas whose initial settings are complete may be accessed.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1

	7	6	5	4	3	2	1	0
	0	BS30	0	BS20	0	BS10	0	BS00
$\overline{\text{CSn}}$ signal		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$

BSn0	Data bus width of CSn space (n = 0 to 3)
0	8 bits
1	16 bits

Caution Be sure to set bits 14, 12, 10, and 8 to 1, and clear bits 15, 13, 11, 9, 7, 5, 3, and 1 to 0.

5.5.3 Access by bus size

The V850ES/SA2 and V850ES/SA3 access the internal peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The bus size is as follows.

- The bus size of the internal peripheral I/O is fixed to 16 bits.
- The bus size of the external memory is selectable from 8 bits or 16 bits (by using the BSC register).

The operation when each of the above is accessed is described below. All data is accessed starting from the lower side.

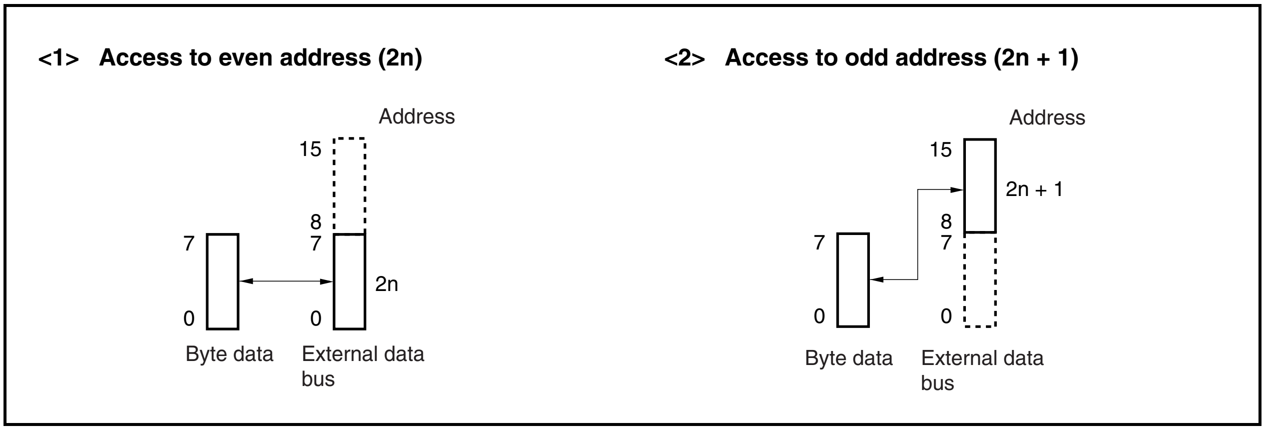
The V850ES/SA2 and V850ES/SA3 support only the little endian format.

Figure 5-2. Little Endian Address in Word

31 24	23 16	15 8	7 0
000BH	000AH	0009H	0008H
0007H	0006H	0005H	0004H
0003H	0002H	0001H	0000H

(1) Byte access (8 bits)

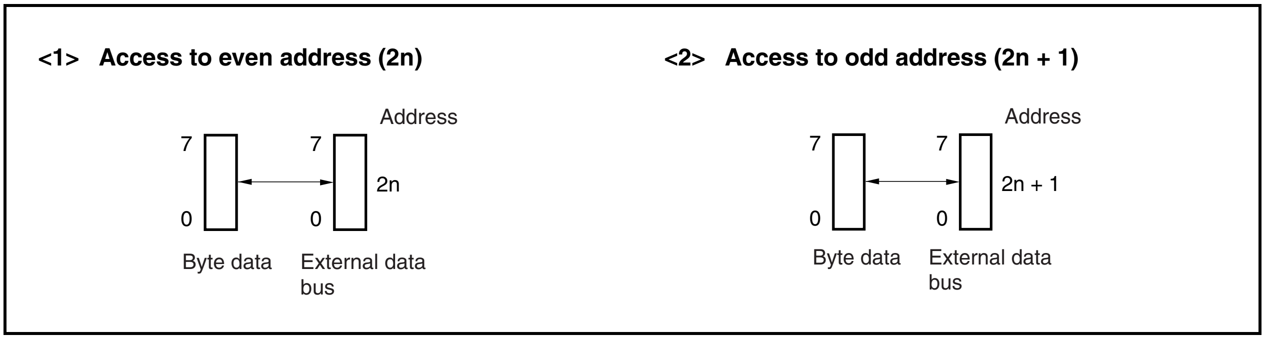
(a) 16-bit data bus width



<1> Access to even address (2n)

<2> Access to odd address (2n + 1)

(b) 8-bit data bus width

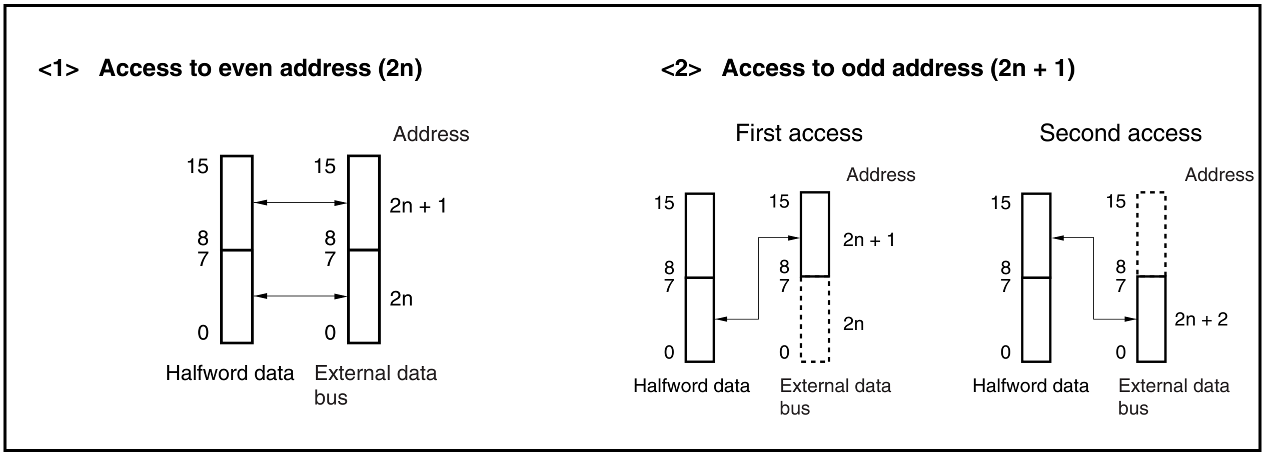


<1> Access to even address (2n)

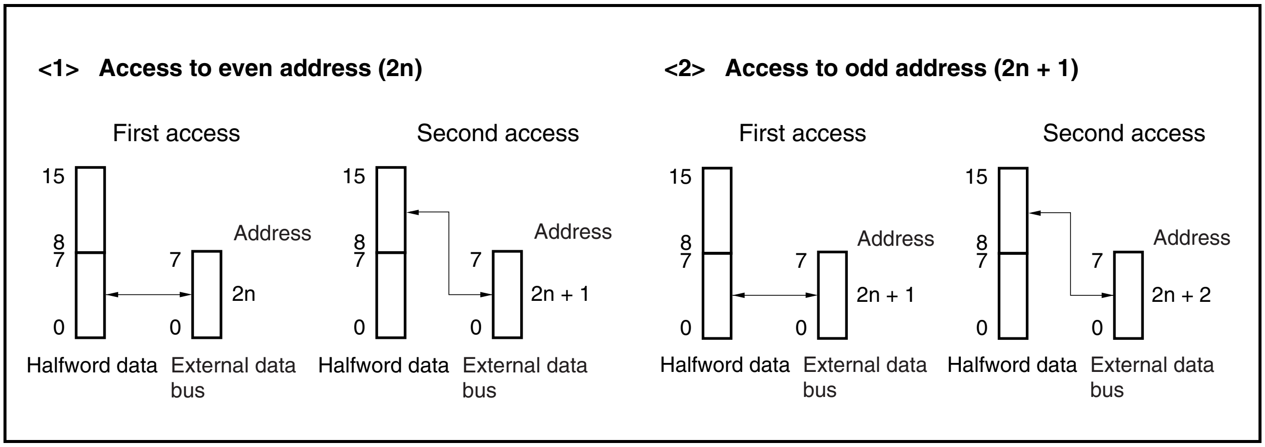
<2> Access to odd address (2n + 1)

(2) Halfword access (16 bits)

(a) With 16-bit data bus width

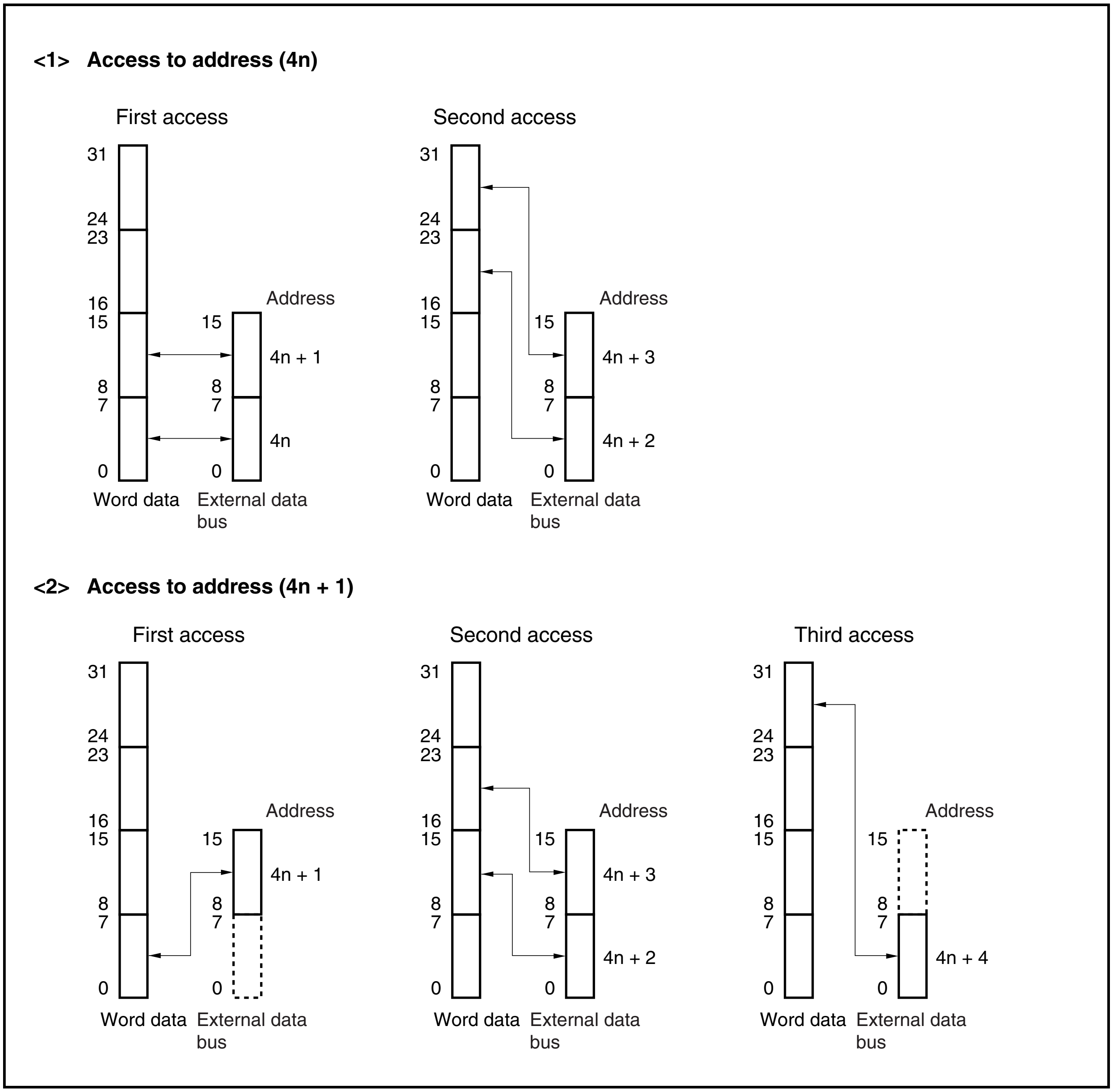


(b) 8-bit data bus width

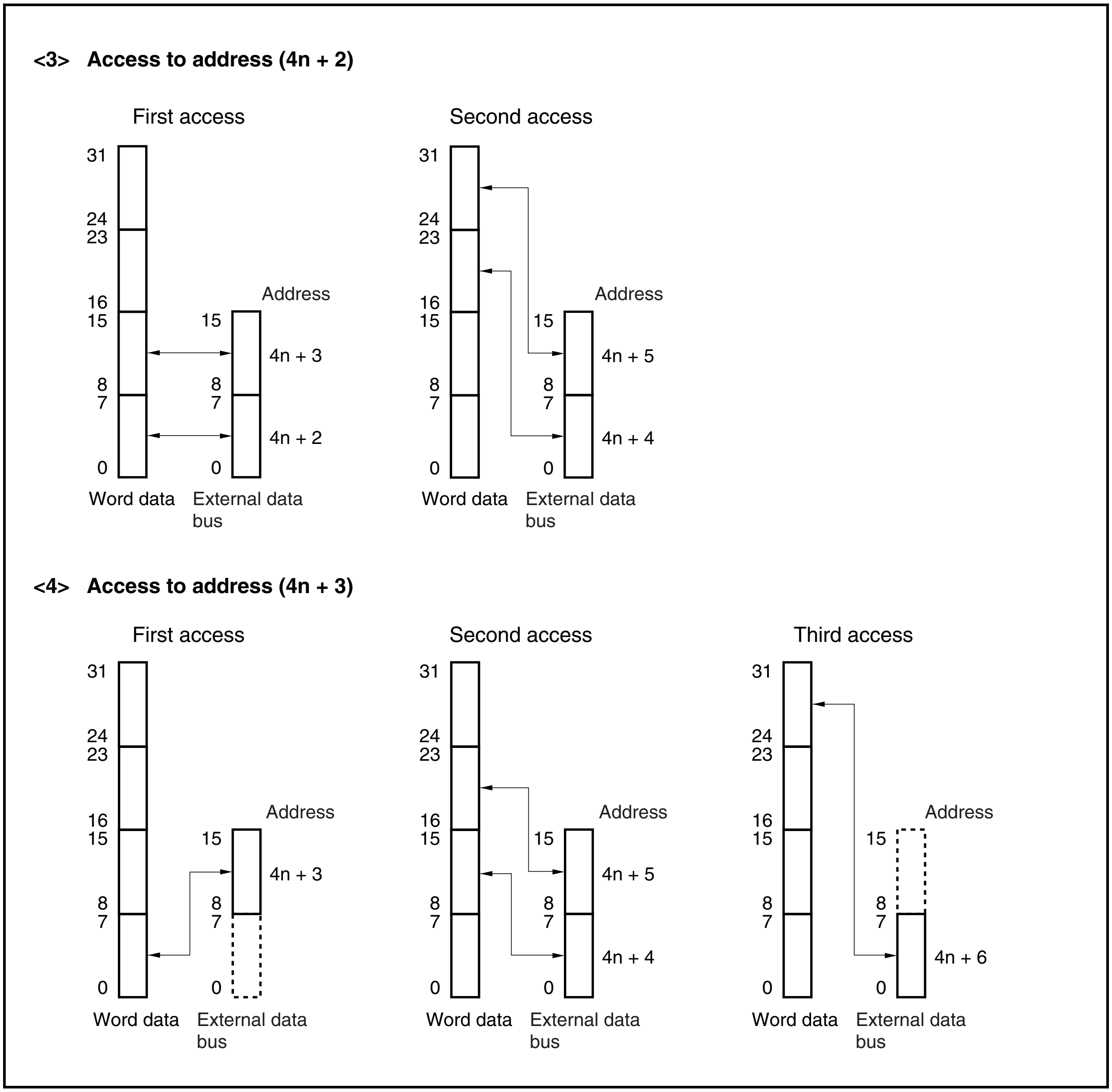


(3) Word access (32 bits)

(a) 16-bit data bus width (1/2)



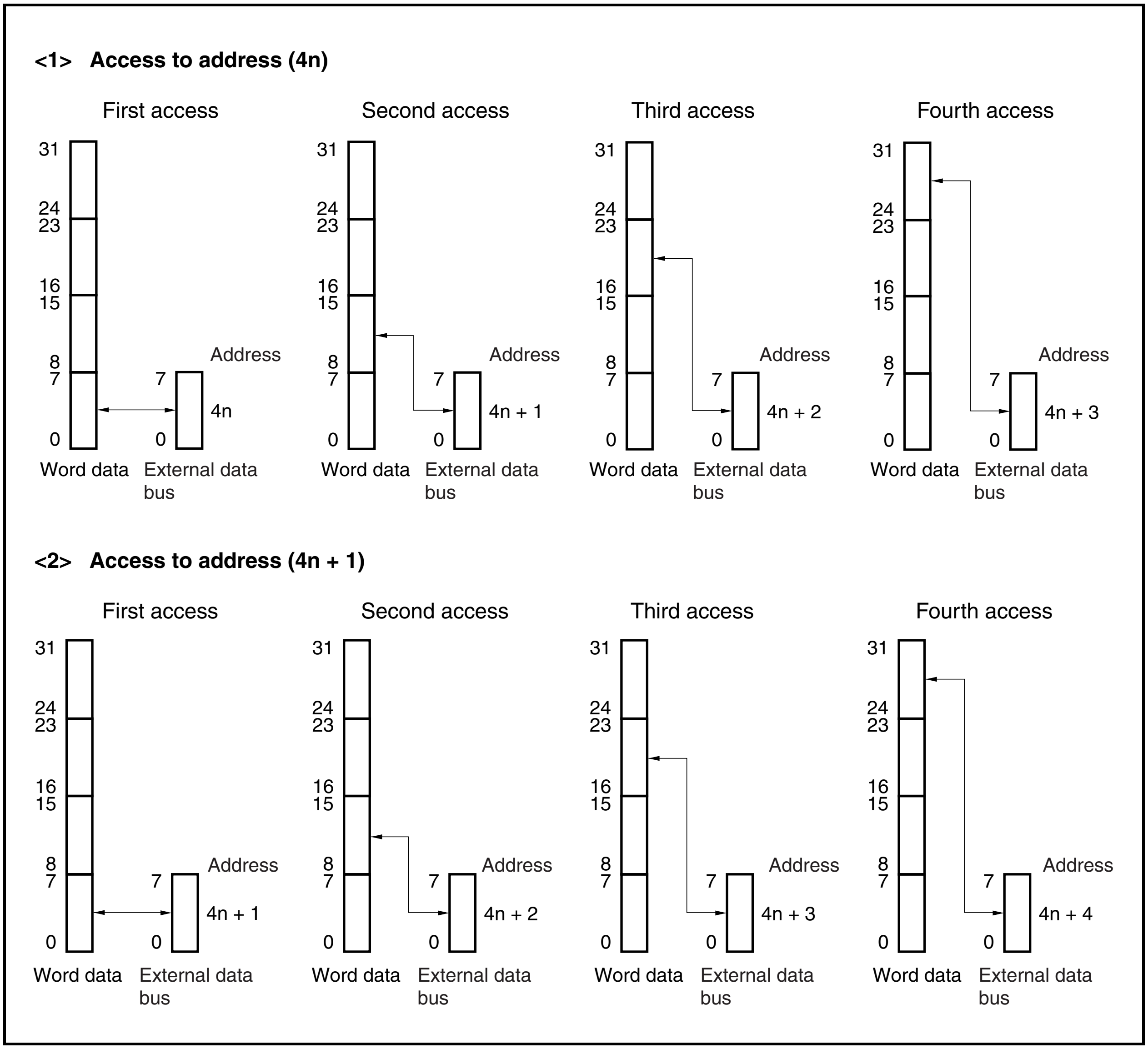
(a) 16-bit data bus width (2/2)



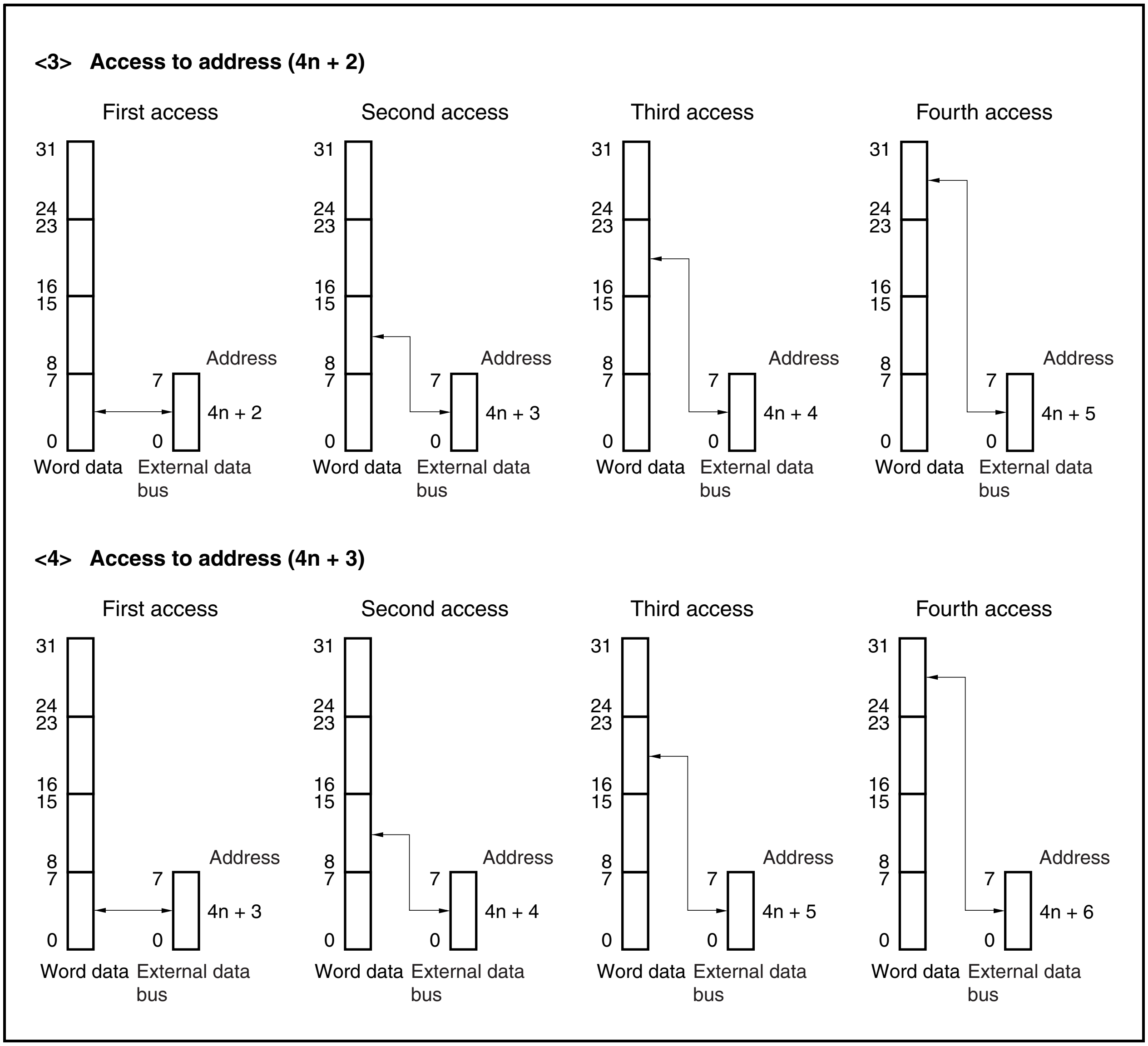
<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

(b) 8-bit data bus width (1/2)



(b) 8-bit data bus width (2/2)



<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

5.6 Wait Function

5.6.1 Programmable wait function

(1) Data wait control register 0 (DWC0)

To realize interfacing with a low-speed memory or I/O, up to seven data wait states can be inserted in the bus cycle that is executed for each CS space.

The number of wait states can be programmed by using data wait control register 0 (DWC0). Immediately after system reset, 7 data wait states are inserted for all the blocks.

The DWC0 register can be read or written in 16-bit units.

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable wait, and are always accessed without a wait state. The internal peripheral I/O area is also not subject to programmable wait, and only wait control from each peripheral function is performed.

2. Write to the DWC0 register after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the DWC0 register are complete. However, external memory areas whose initial settings are complete may be accessed.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20
$\overline{\text{CS}}$n signal		$\overline{\text{CS3}}$				$\overline{\text{CS2}}$		

	7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00
$\overline{\text{CS}}$n signal		$\overline{\text{CS1}}$				$\overline{\text{CS0}}$		

DWn2	DWn1	DWn0	Number of wait states inserted in $\overline{\text{CS}}$n space (n = 0 to 3)
0	0	0	None
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Caution Be sure to clear bits 15, 11, 7, and 3 to 0.

5.6.2 External wait function

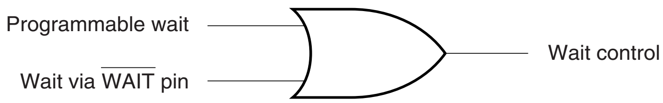
To synchronize an extremely slow external device, I/O, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).

Access to each area of the internal ROM, internal RAM, and internal peripheral I/O is not subject to control by the external wait function, in the same manner as the programmable wait function.

The $\overline{\text{WAIT}}$ signal can be input asynchronously to CLKOUT, and is sampled at the falling edge of the clock in the T2 and TW states of the bus cycle in the multiplexed bus mode. In the separate bus mode, it is sampled at the rising edge of the clock immediately after the T1 and TW states of the bus cycle. If the setup/hold time of the sampling timing is not satisfied, a wait state is inserted in the next state, or not inserted at all.

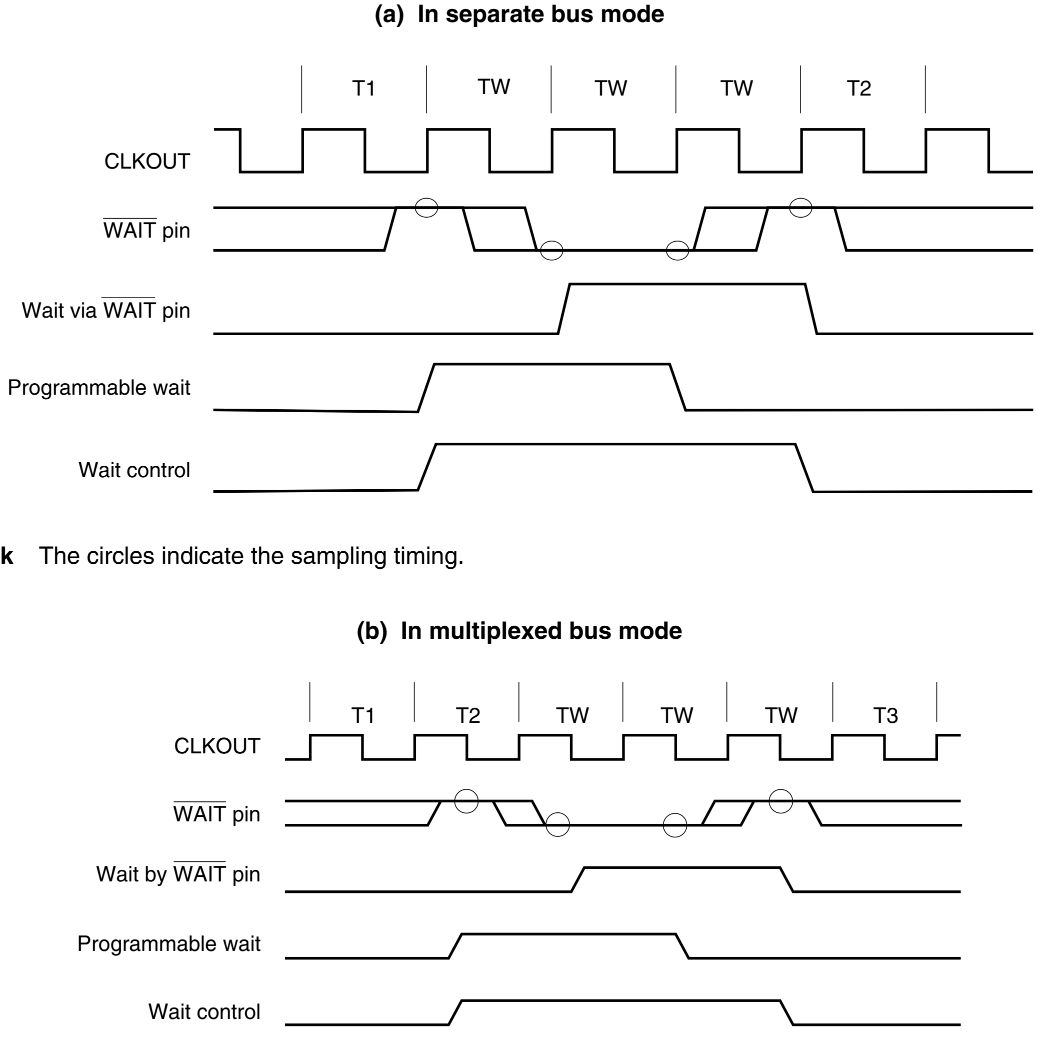
5.6.3 Relationship between programmable wait and external wait

Wait cycles are inserted as the result of an OR operation between the wait cycles specified by the set value of the programmable wait and the wait cycles controlled by the $\overline{\text{WAIT}}$ pin.



For example, if the timing of the programmable wait and the $\overline{\text{WAIT}}$ pin signal is as illustrated below, three wait states will be inserted in the bus cycle.

Figure 5-3. Example of Wait Insertion


T1
TW
TW
TW
T2
CLKOUT
$\overline{\text{WAIT}}$ pin
Wait via $\overline{\text{WAIT}}$ pin
Programmable wait
Wait control
T1
T2
TW
TW
TW
T3
CLKOUT
$\overline{\text{WAIT}}$ pin
Wait by $\overline{\text{WAIT}}$ pin
Programmable wait
Wait control

(a) In separate bus mode

Remark The circles indicate the sampling timing.

(b) In multiplexed bus mode

Remark The circles indicate the sampling timing.

5.6.4 Programmable address wait function

Address-setup or address-hold waits to be inserted in each bus cycle can be set by using the address wait control register (AWC). Address wait insertion is set for each chip select area ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$).

If an address setup wait is inserted, it seems that the high-clock period of T1 state is extended by 1 clock. If an address hold wait is inserted, it seems that the low-clock period of T1 state is extended by 1 clock.

(1) Address wait control register (AWC)

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

After reset: FFFFH   R/W   Address: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
	AHW3	ASW3	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
$\overline{\text{CSn}}$ signal	$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

AHWn	Specifies insertion of address hold wait (n = 0 to 3)
0	Not inserted
1	Inserted

ASWn	Specifies insertion of address setup wait (n = 0 to 3)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15 to 8 to 1.

5.7 Idle State Insertion Function

To facilitate interfacing with low-speed memories, one idle state (TI) can be inserted after the T3 state in the bus cycle that is executed for each space selected by the chip select function in the multiplexed address/data bus mode. In the separate bus mode, one idle state (TI) can be inserted after the T2 state. By inserting an idle state, the data output float delay time of the memory can be secured during read access (an idle state cannot be inserted during write access).

Whether the idle state is to be inserted can be programmed by using the bus cycle control register (BCC).

An idle state is inserted for all the areas immediately after system reset.

(1) Bus cycle control register (BCC)

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

Cautions 1. The internal ROM, internal RAM, and internal peripheral I/O areas are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the BCC register are complete. However, external memory areas whose initial settings are complete may be accessed.

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0

	7	6	5	4	3	2	1	0
	BC31	0	BC21	0	BC11	0	BC01	0
$\overline{CSn}$ signal	$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

BCn1	Specifies insertion of idle state (n = 0 to 3)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15, 13, 11, and 9 to 1, and clear bits 14, 12, 10, 8, 6, 4, 2, and 0 to 0.

5.8 Bus Hold Function

5.8.1 Functional outline

The $\overline{\text{HLDAK}}$ and $\overline{\text{HLDRQ}}$ functions are valid if the PCM2 and PCM3 pins are set in the control mode.

When the $\overline{\text{HLDRQ}}$ pin is asserted (low level), indicating that another bus master has requested bus mastership, the external address/data bus goes into a high-impedance state and is released (bus hold status). If the request for the bus mastership is cleared and the $\overline{\text{HLDRQ}}$ pin is deasserted (high level), driving these pins is started again.

During the bus hold period, execution of the program in the internal ROM and internal RAM is continued until a peripheral I/O register or the external memory is accessed.

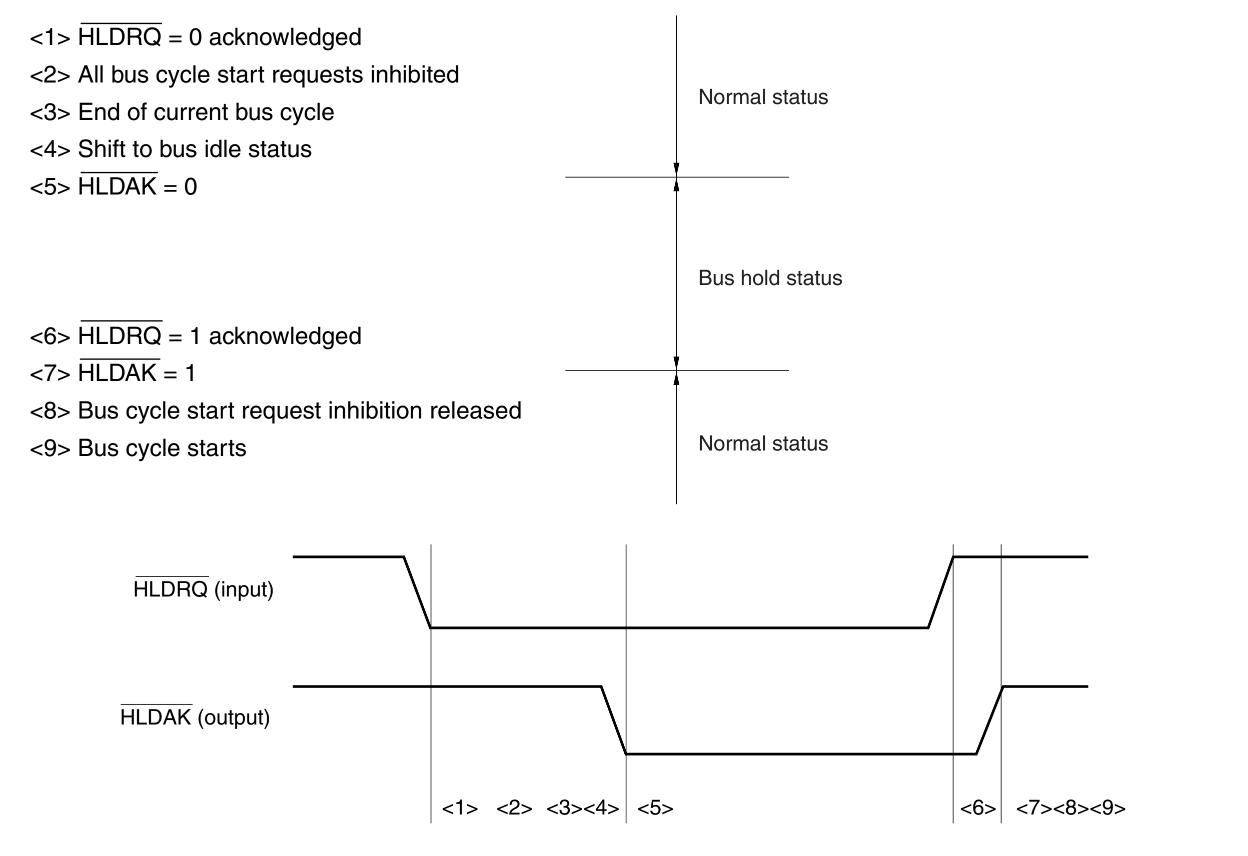
The bus hold status is indicated by assertion of the $\overline{\text{HLDAK}}$ pin (low level). The bus hold function enables the configuration multi-processor type systems in which two or more bus masters exist.

Note that the bus hold request is not acknowledged during a multiple-access cycle initiated by the bus sizing function or a bit manipulation instruction.

Status	Data Bus Width	Access Type	Timing in Which Bus Hold Request Not Acknowledged
CPU bus lock	16 bits	Word access to even address	Between first and second access
		Word access to odd address	Between first and second access
			Between second and third access
		Halfword access to odd address	Between first and second access
	8 bits	Word access	Between first and second access
			Between second and third access
			Between third and fourth access
		Halfword access	Between first and second access
Read-modify-write access of bit manipulation instruction	–	–	Between read access and write access

5.8.2 Bus hold procedure

The bus hold status transition procedure is shown below.



<1> $\overline{\text{HLDRQ}}$ = 0 acknowledged
<2> All bus cycle start requests inhibited
<3> End of current bus cycle
<4> Shift to bus idle status
<5> $\overline{\text{HLDAK}}$ = 0

<6> $\overline{\text{HLDRQ}}$ = 1 acknowledged
<7> $\overline{\text{HLDAK}}$ = 1
<8> Bus cycle start request inhibition released
<9> Bus cycle starts

5.8.3 Operation in power save mode

Because the internal system clock is stopped in the software STOP and IDLE modes, the bus hold status is not entered even if the $\overline{\text{HLDRQ}}$ pin is asserted.

In the HALT mode, the $\overline{\text{HLDAK}}$ pin is asserted as soon as the $\overline{\text{HLDRQ}}$ pin has been asserted, and the bus hold status is entered. When the $\overline{\text{HLDRQ}}$ pin is later deasserted, the $\overline{\text{HLDAK}}$ pin is also deasserted, and the bus hold status is cleared.

5.9 Bus Priority

Bus hold, instruction fetch (branch), instruction fetch (successive), and operand data accesses are executed in the external bus cycle.

Bus hold has the highest priority, followed by operand data access, instruction fetch (branch), and instruction fetch (successive).

An instruction fetch may be inserted between the read access and write access in a read-modify-write access.

If an instruction is executed for two or more accesses, an instruction fetch and bus hold are not inserted between accesses due to bus size limitations.

Table 5-4. Bus Priority

Priority	External Bus Cycle	Bus Master
High	Bus hold	External device
	DMA transfer	DMAC
↕	Operand data access	CPU
	Instruction fetch (branch)	CPU
Low	Instruction fetch (successive)	CPU

5.10 Boundary Operation Conditions

5.10.1 Program space

(1) If a branch instruction exists at the upper limit of the internal RAM area, a prefetch operation straddling over the internal peripheral I/O area (invalid fetch) does not occur.

(2) Instruction execution to the external memory area cannot be continued without a branch from the internal ROM area to the external memory area.

5.10.2 Data space

The V850ES/SA2 and V850ES/SA3 have an address misalign function.

With this function, data can be placed at all addresses, regardless of the format of the data (word data or halfword data). However, if the word data or halfword data is not aligned at the boundary, a bus cycle is generated at least twice, causing the bus efficiency to drop.

(1) Halfword-length data access

A byte-length bus cycle is generated twice if the least significant bit of the address is 1.

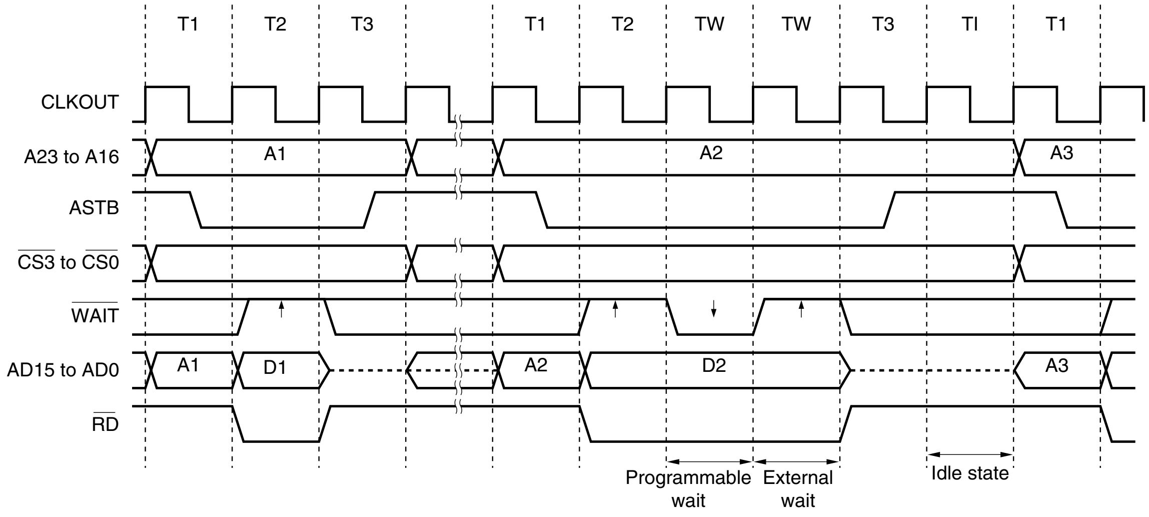
(2) Word-length data access

(a) A byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle are generated in that order if the least significant bit of the address is 1.

(b) A halfword-length bus cycle is generated twice if the lower 2 bits of the address are 10.

★ 5.11 Bus Timing

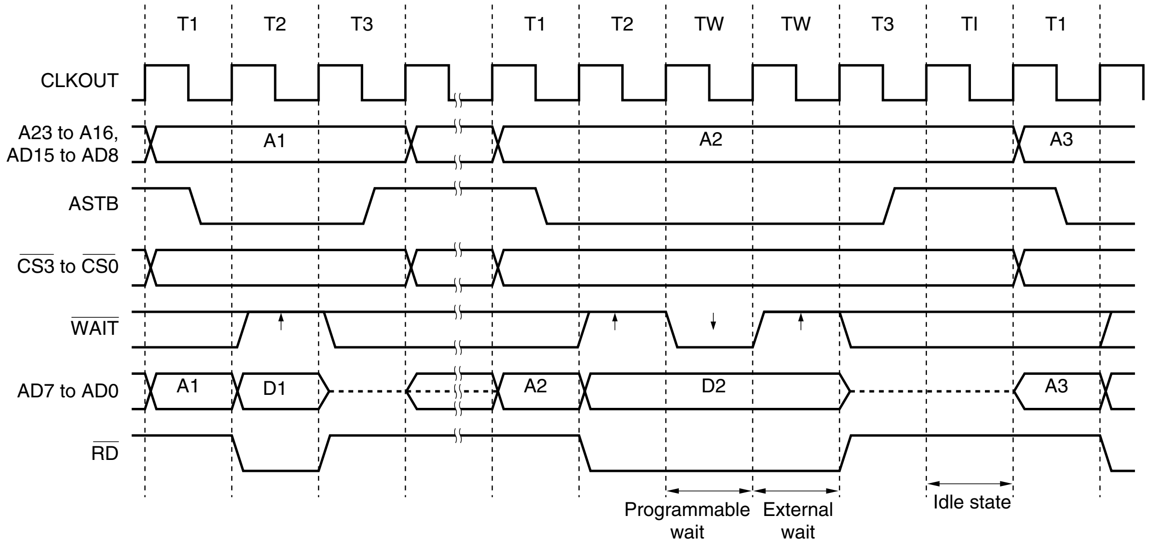
Figure 5-4. Multiplexed Bus Read Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
AD15 to AD8	Active	Hi-Z
AD7 to AD0	Hi-Z	Active

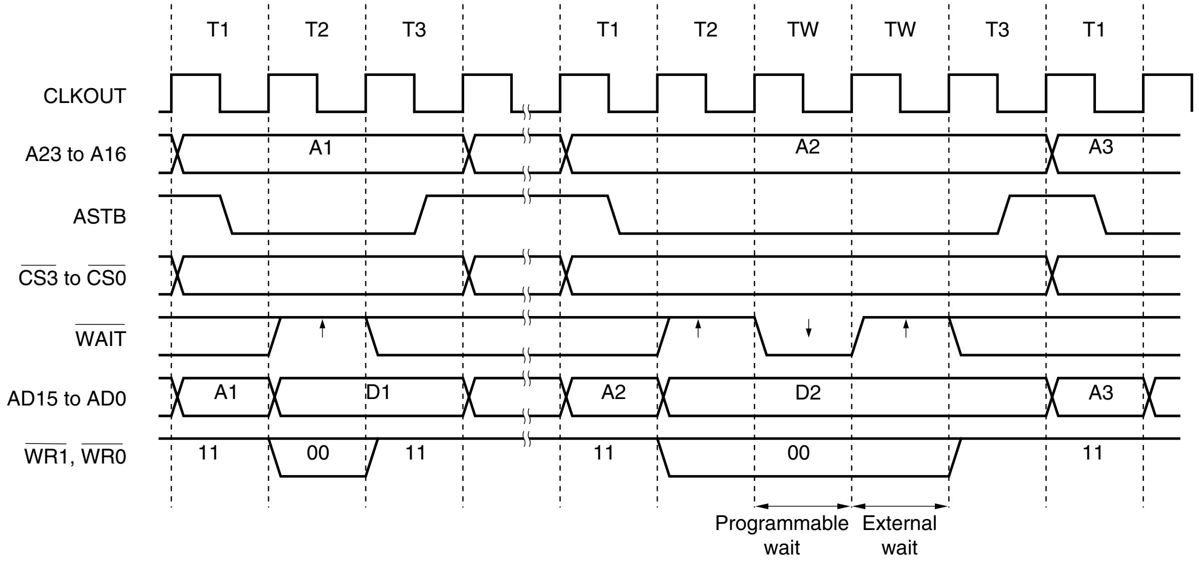
Remark The broken lines indicate high impedance.

Figure 5-5. Multiplexed Bus Read Timing (Bus Size: 8 Bits)



Remark The broken lines indicate high impedance.

Figure 5-6. Multiplexed Bus Write Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
AD15 to AD8	Active	Undefined
AD7 to AD0	Undefined	Active
WR1, WR0	01	10

Figure 5-7. Multiplexed Bus Write Timing (Bus Size: 8 Bits)

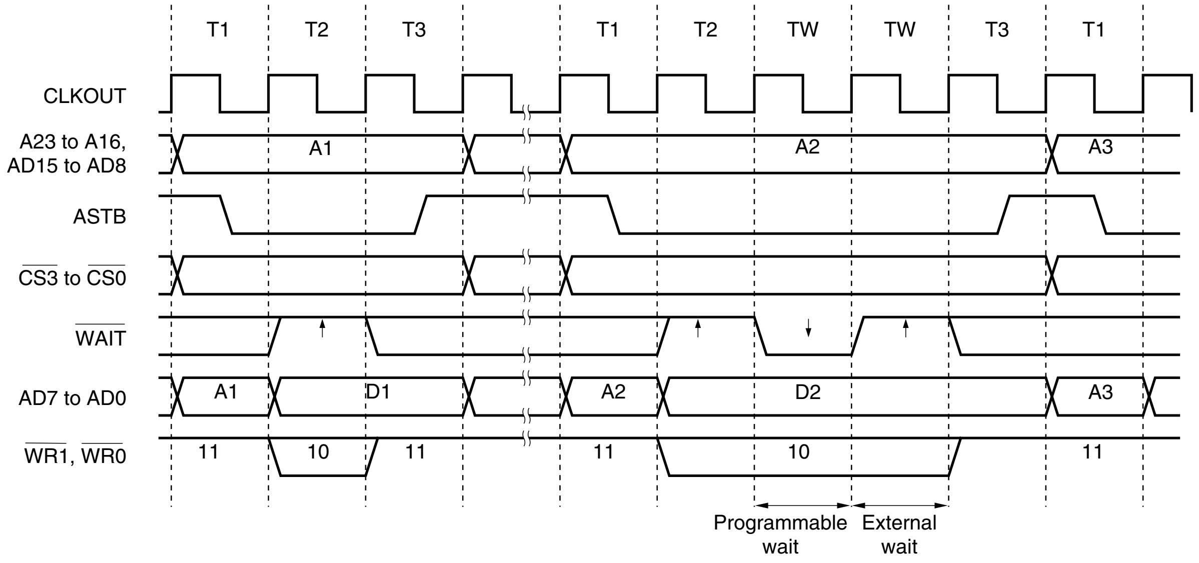
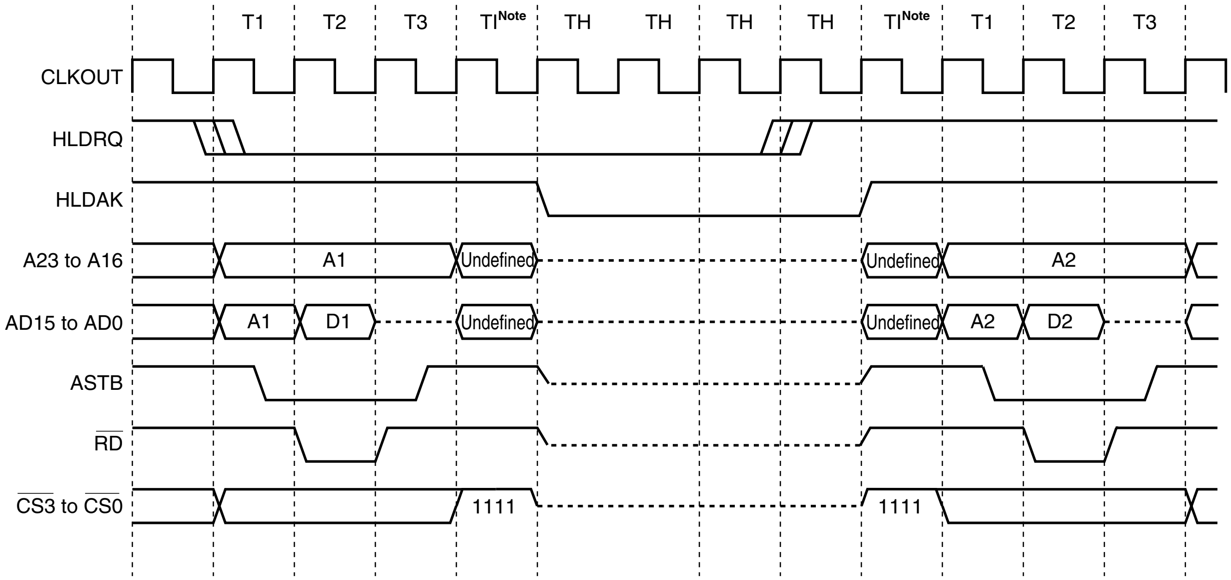


Figure 5-8. Multiplexed Bus Hold Timing (Bus Size: 16 Bits, 16-Bit Access)

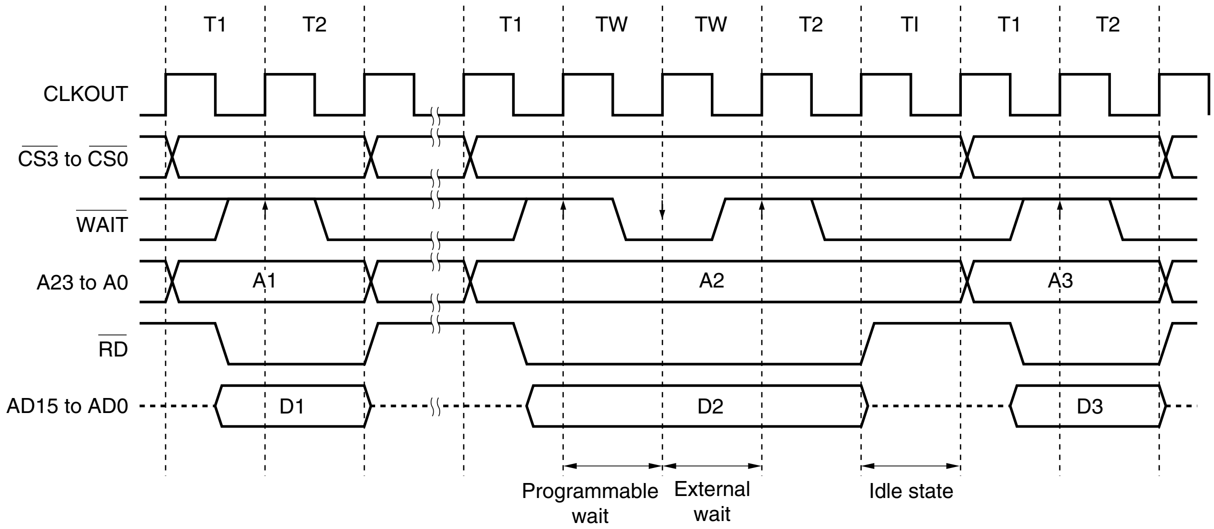


Note This idle state (TI) does not depend on the BCC register settings.

Remarks 1. Refer to **Table 2-3** for the pin statuses in the bus hold mode.

2. The broken lines indicate high impedance.

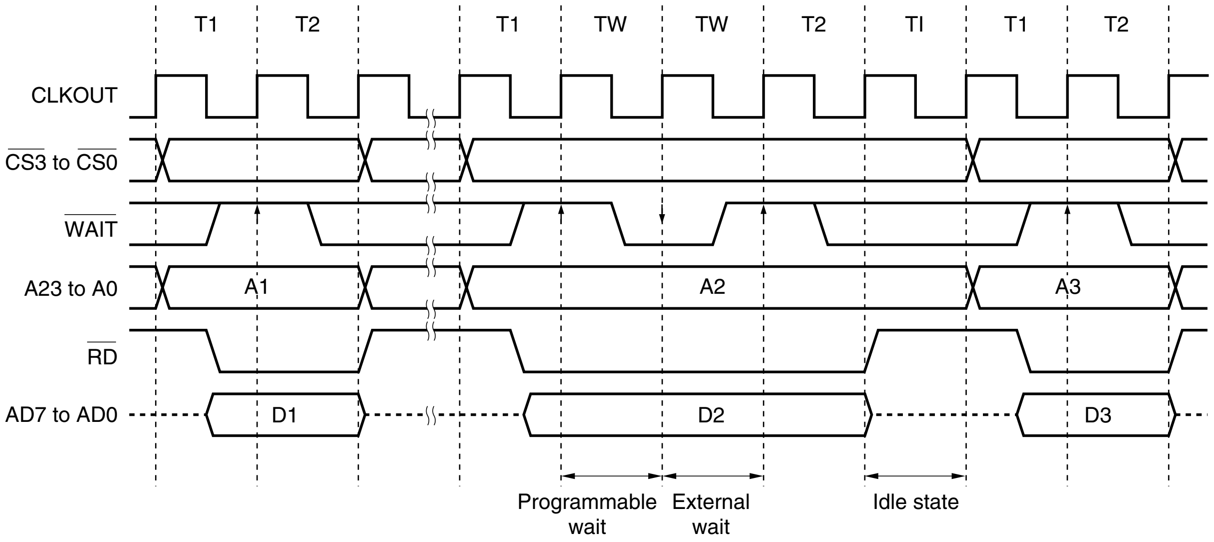
Figure 5-9. Separate Bus Read Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
AD15 to AD8	Active	Hi-Z
AD7 to AD0	Hi-Z	Active

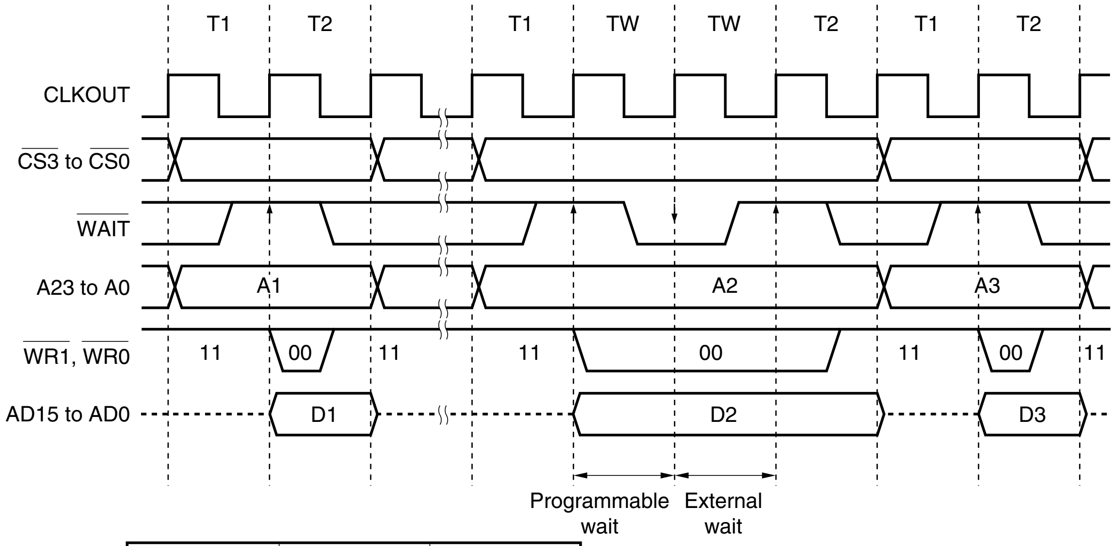
Remark The broken lines indicate high impedance.

Figure 5-10. Separate Bus Read Timing (Bus Size: 8 Bits)



Remark The broken lines indicate high impedance.

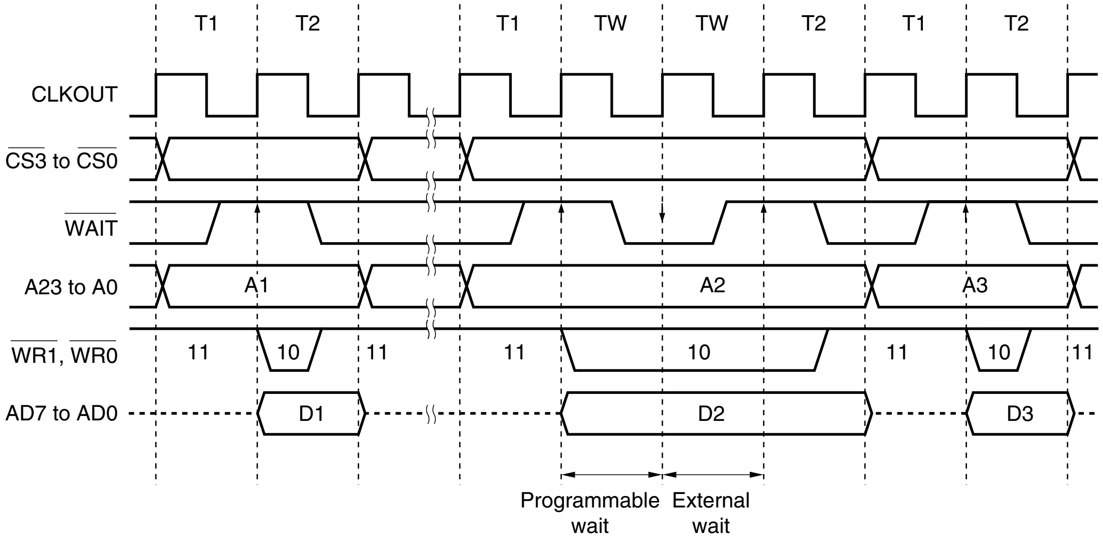
Figure 5-11. Separate Bus Write Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
AD15 to AD8	Active	Undefined
AD7 to AD0	Undefined	Active
WR1, WR0	01	10

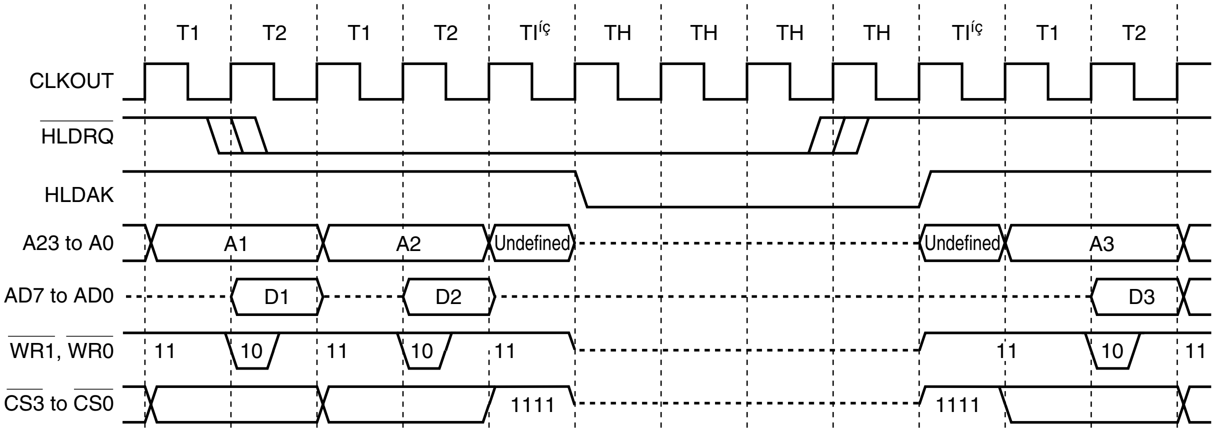
Remark The broken lines indicate high impedance.

Figure 5-12. Separate Bus Write Timing (Bus Size: 8 Bits)



Remark The broken lines indicate high impedance.

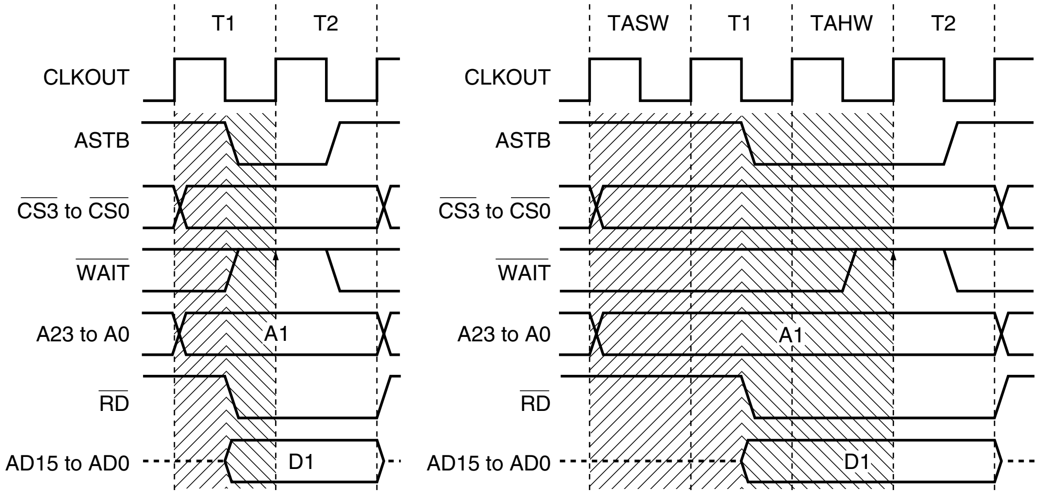
Figure 5-13. Separate Bus Hold Timing (Bus Size: 8 Bits, Write)



Note This idle state (TI) does not depend on the BCC register settings.

Remark The broken lines indicate high impedance.

Figure 5-14. Address Wait Timing (Separate Bus Read, Bus Size: 16 Bits, 16-Bit Access)



Remarks 1. TASW (address setup wait): Image of high-level width of T1 state expanded.

2. TAHW (address hold wait): Image of low-level width of T1 state expanded.

3. The broken lines indicate high impedance.

CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Overview

The features of the clock generation function are as follows.

○ Main clock oscillator (f_X)

- 2 to 20 MHz (at 2.2 to 2.7 V operation)

○ Subclock oscillator (f_{XT})

- 32.768 kHz

○ Generation of internal system clock (f_{CLK})

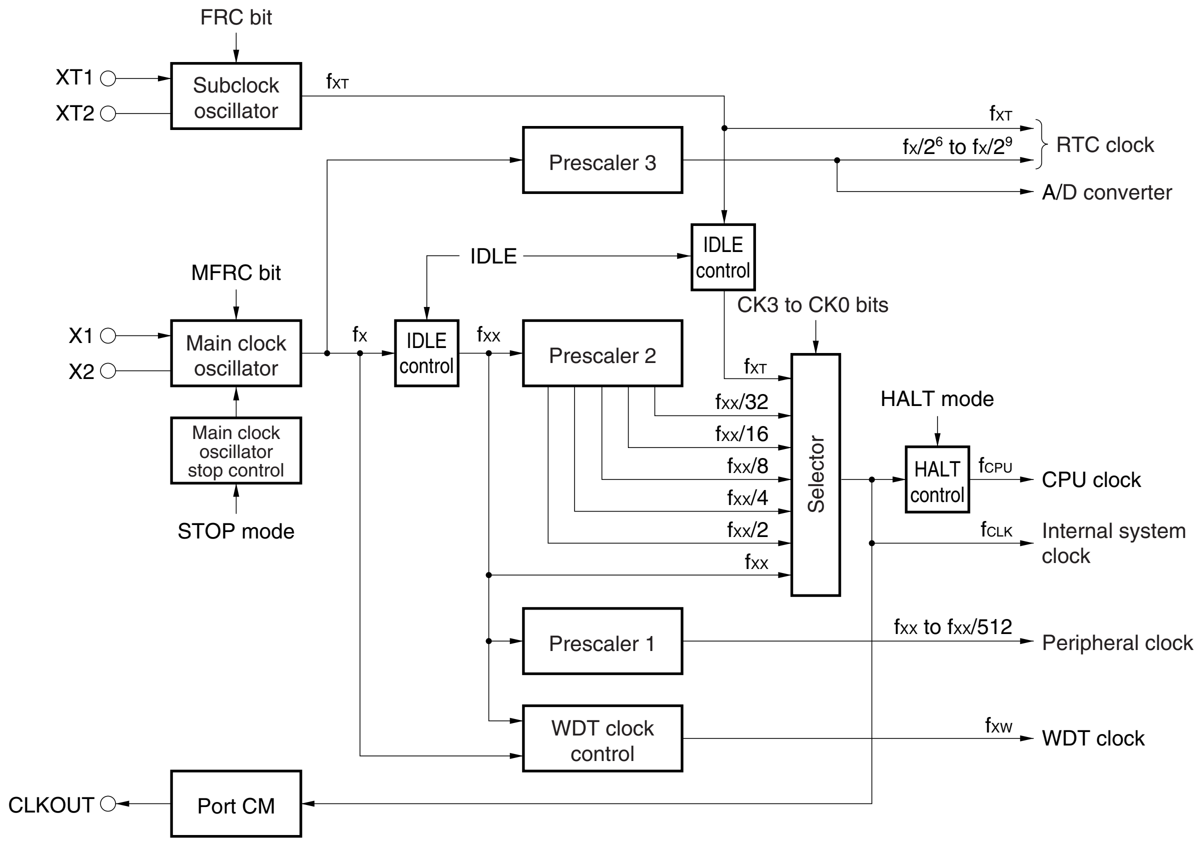
- Seven steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{XT})

○ Generation of peripheral clock

○ Clock output function (CLKOUT)

6.2 Configuration

Figure 6-1. Clock Generator



Remark f_X: Main clock oscillation frequency

f_{XX}: Main clock frequency

f_{XT}: Subclock frequency

f_{CPU}: CPU clock frequency

f_{CLK}: Internal system clock frequency

f_{XW}: Watchdog timer clock frequency

(1) Main clock oscillator

This circuit oscillates the following frequency (f_X):

- 2 to 20 MHz (at 2.2 to 2.7 V operation)

(2) Subclock oscillator

This circuit oscillates a frequency of 32.768 kHz (f_{XT}).

(3) Main clock resonator stop control

This circuit generates a control signal that stops oscillation of the main clock resonator.

It stops the oscillation of the main clock resonator in the software STOP mode or when the MCK bit = 1 (valid only when the CLS bit = 1).

(4) Prescaler 1

This circuit generates the clock (f_{XX} to $f_{XX}/512$) to be supplied to the internal peripheral functions.

The clock is supplied to the following blocks:

TM0 to TM5, CSI0 to CSI4, UART0, UART1, I^2C, ADC, DAC

(5) Prescaler 2

This circuit divides the main clock (f_{XX}).

The clock generated by prescaler 2 (f_{XX} to $f_{XX}/32$) is supplied to the selector that generates the internal system clock (f_{CLK}).

f_{CLK} is the clock that is supplied to the CPU, INTC, DMAC, and ROMC blocks, and can be output from the CLKOUT pin.

(6) Prescaler 3

This circuit divides the clock (f_X) generated by the main resonator to a specific frequency (32.768 kHz) and supplies it to the RTC and ADC.

For details, refer to **6.5 Prescaler 3**.

(7) Watchdog timer clock control

This circuit generates the clock (f_{XW}) to be supplied to the watchdog timer.

The watchdog timer is used alternately as the oscillation stabilization timer, so the source clock is automatically switched according to the operation status shown below.

- From software STOP mode or $\overline{\text{RESET}}$ pin input to when oscillation stabilization time has been counted: f_X
- Other than above: f_{XX}

6.3 Control Registers

(1) Processor clock control register (PCC)

The processor clock control register (PCC) is a special register. Data can be written to it only in combination of specific sequences (refer to **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units. The CLS bit is a read-only bit.

After reset: 03H R/W Address: FFFFF828H

	7	<6>	5	<4>	<3>	2	1	0
PCC	FRC	MCK	MFRC	CLS[Note]	CK3	CK2	CK1	CK0

FRC	Selects internal feedback resistor of subclock
0	Used
1	Not used

MCK	Operation of main clock
0	Operating
1	Stopped
• Even if the MCK bit is set to 1 while the system is operating with the main clock as the CPU clock, the operation of the main system clock does not stop. It stops after the CPU clock has been changed to the subclock. • When the main clock is stopped and the device is operating on the subclock, clear the MCK bit to 0 and wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.	

MFRC	Selects internal feedback resistor of main clock
0	Used
1	Not used

CLS	Status of CPU clock (f_{CPU})
0	Main clock operation
1	Subclock operation

CK3	CK2	CK1	CK0	Selects clock (f_{CLK}/f_{CPU})
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	X	Setting prohibited
1	X	X	X	f_{XT} (subclock: 32.768 kHz)

Note The CLS bit is a read-only bit.

Caution Do not change the CPU clock (by using the CK2 to CK0 bits of the PCC register) while CLKOUT is being output.

Remark X: Don’t care.

Examples of settings to change between the main clock and subclock are shown below

(a) Example of setting when changing from main clock to subclock

<1> Checking internal system clock: Check if the following condition is satisfied.
- Internal system clock (f_{CLK}) > Subclock (32.768 kHz) × 4

If this condition is not satisfied, change the setting of the CK2 to CK0 bits so that the condition is satisfied. At this time, do not change the setting of the CK3 bit.

<2> CK3 bit ← “1”: Use of a bit manipulation instruction is recommended. Do not change the setting of the CK2 to CK0 bits.

<3> Subclock operation: The following time is required between when the CK3 bit is set and when the subclock operation is started:
- Maximum: (1/Subclock frequency)

Therefore, read the CLS bit and confirm that the subclock operation has started.

<4> MCK ← “1”: Set MCK to 1 only when stopping the main clock.

(b) Example of setting when changing subclock to main clock

<1> MCK ← “0”: Oscillation of the main clock is started.

<2> Software wait: Insert wait states by program and wait until the oscillation stabilization time of the main clock elapses.

<3> CK3 ← “0”: Use of a bit manipulation instruction is recommended. Do not change the setting of the CK2 to CK0 bits.

<3> Main clock operation: The following time is required between when the CK3 bit is set and when the main clock specified by the CK2 to CK0 bits is selected.
- Maximum: (1/Subclock frequency)

Therefore, read the CLS bit and confirm that the main clock operation has started.

(2) Power save control register (PSC)

The power save control register (PSC) is a special register. Data can be written to this register only in combination of specific sequences (refer to **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	4	3	2	<1>	0
PSC	0	0	0	0	0	0	STP	0

STP	Setting of IDLE/software STOP mode
0	Normal mode
1	IDLE/software STOP mode

(3) Power save mode register (PSMR)

This is an 8-bit register that controls the operation status and clock operation in the power save mode.

It can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM

PSM	Specifies operation in software standby mode (valid when bit 1 (STP) of the PSC register is set to 1)
0	IDLE mode
1	Software STOP mode

Cautions 1. Be sure to clear bits 1 to 7 of the PSMR register to 0.

2. The PSM bit is valid only when the STP bit of the PSC register is set to 1.

(4) Oscillation stabilization time selection register (OSTS)

This is an 8-bit register that controls the operation status and clock in the power save mode.

Refer to **10.3 (1) Oscillation stabilization time selection register (OSTS)**.

6.4 Operation

6.4.1 Operation of each clock

The following table shows the operation status of each clock.

★ **Table 6-1. Operation Status of Each Clock**

	CLS Bit = 0 MCK Bit = 0					CLS Bit = 1 MCK Bit = 0		CLS Bit = 1 MCK Bit = 1	
	<1>	<2>	<3>	<4>	<5>	<6>	<7>	<6>	<7>
Main resonator (f_X)	×	√	√	√	×	√	√	×	×
Sub-resonator (f_{XT})	√	√	√	√	√	√	√	√	√
CPU clock (f_{CPU})	×	×	×	×	×	√	×	√	×
Internal system clock (f_{CLK})	×	×	√	×	×	√	×	√	×
Peripheral clock (f_{XX} to $f_{XX}/512$)	×	×	√	×	×	√	×	×	×
WDT clock (f_{XW})	×	√	√	×	×	**Note**	×	×	×
RTC clock (sub)	√	√	√	√	√	√	√	√	√
RTC clock (main)	×	√	√	√	×	√	√	×	×

Note The watchdog timer clock (f_{XW}) is operable but it stops operating in the watchdog timer if the CLS bit of the processor clock control register (PCC) is set to 1.

Remarks CLS bit: Bit 6 of PCC register
MCK bit: Bit 4 of PCC register
√: Operable
×: Stops
<1>: $\overline{\text{RESET}}$ pin input
<2>: During oscillation stabilization time count
<3>: HALT mode
<4>: IDLE mode
<5>: Software STOP mode
<6>: Subclock operation mode
<7>: Sub-IDLE mode

6.4.2 Clock output function

The clock output function allows the CLKOUT pin to output the internal system clock (f_{CLK}).

The internal system clock (f_{CLK}) is selected by using the CK3 to CK0 bits of the processor clock control register (PCC).

The CLKOUT pin functions alternately as the PCM1 pin and operates as a clock output pin when the control register of port CM is manipulated (refer to **4.3.9 Port CM**).

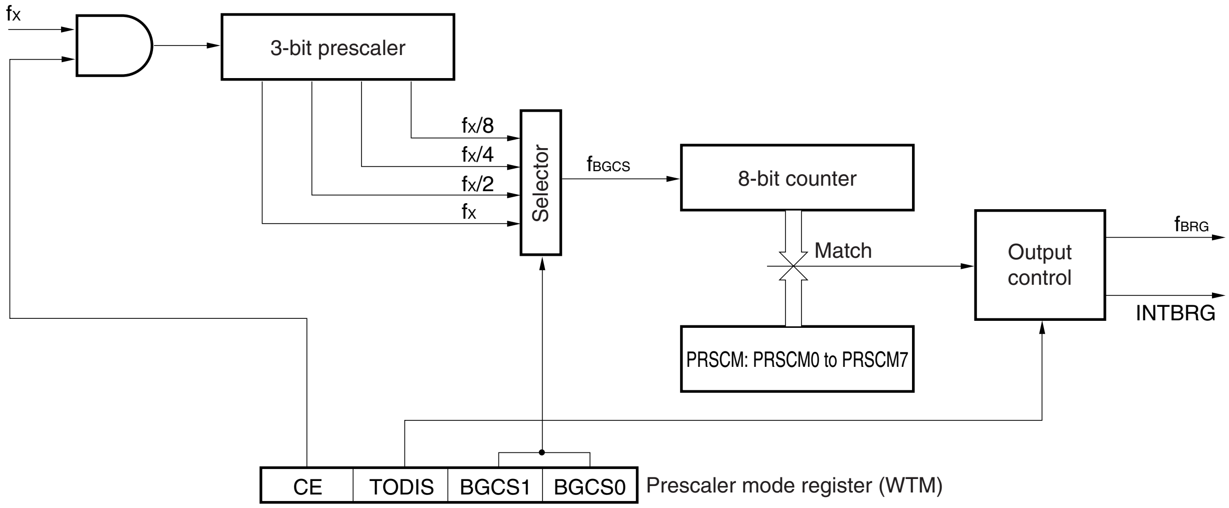
The status of the CLKOUT pin is the same as the internal system clock in Table 6-1, and can output the clock when it is √ (operable). When it is × (stops), it outputs a low level. Immediately after reset <1> and in the operation status of <2>, the alternate function of the CLKOUT pin is used (PCM1: input mode), and therefore the pin goes into a high-impedance state.

6.5 Prescaler 3

Prescaler 3 has the following functions.

- Generation of clock for count clock of watch timer (source clock: main clock oscillation)
- Generation of clock for count clock of A/D converter (source clock: main clock oscillator)
- Interval timer (INTBRG)

Figure 6-2. Block Diagram of Prescaler 3



Remark f_{BRG}: Prescaler 3 clock frequency
f_X: Main clock oscillation frequency

6.5.1 Control register

(1) Prescaler mode register (PRSM)

The PRSM register controls generation of the clock of the watch timer and A/D converter.

This register can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF8B0H

	7	6	5	4	3	2	1	0
PRSM	0	0	0	CE	0	TODIS	BGCS1	BGCS0

CE	TODIS	Baud rate output	Baud rate interrupt signal (INTBRG)
0	X	Fixed to 0	Fixed to 0
1	0	Operates	Operates
1	1	Fixed to 0	Operates

BGCS1	BGCS0	Selects input clock (f_{BRGS})		
			10 MHz	4 MHz
0	0	f_X	100 ns	250 ns
0	1	$f_X/2$	200 ns	500 ns
1	0	$f_X/4$	400 ns	1 μs
1	1	$f_X/8$	800 ns	2 μs

Cautions 1. Do not change the values of BGCS1 and BGCS0 during transmission/reception.

2. Set the PRSM register before setting the CE bit to 1.

3. Set the PRSM and PRSCM registers in accordance with the main clock frequency to be used, so that the frequency of f_{BRG} is 32.768 kHz.

(2) Prescaler compare register (PRSCM)

This is an 8-bit compare register.

It can be read or written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM	PRSCM7	PRSCM6	PRSCM5	PRSCM4	PRSCM3	PRSCM2	PRSCM1	PRSCM0

Cautions 1. Do not change the value of the PRSCM register during transmission/reception.

2. Set the PRSCM register before setting the CE bit of the PRSM register to 1.

3. Set the PRSM and PRSCM registers in accordance with the main clock frequency to be used, so that the frequency of f_{BRG} is 32.768 kHz.

6.5.2 Generation of clock

(1) Count clock of watch timer

The clock (f_{BRG}) input to the watch timer can be corrected to 32.768 kHz or equivalent frequency.

The relationship between the main clock oscillation frequency (f_X), the set value of input clock selection bit BGCSn (m), the set value of the PRCSM register (N), and the output clock (f_{BRG}) is as follows:

Example: Where f_X = 4.00 MHz, m = 0 (BGCS1 = BGCS = 0), N = 3DH, f_{BRG} = 32.768 kHz

$$f_{BRG} = f_X/(2^m \times N \times 2)$$

Remark f_{BRG}: Count clock

N: Value of compare register in prescaler 3 (1 to FFH)

N = 256 if the value of the compare register is “00H”.

(2) Interval timer

This timer generates a baud rate interrupt request (INTBRG) at preset time intervals.

The interval time can be set by using the BGCS1 and BGCS0 bits of the prescaler mode register (PRSM) and the prescaler compare register (PRSCM).

The interval time can be calculated by the following expression.

$$\text{Interval time} = f_X/(2^m \times N)$$

CHAPTER 7 16-BIT TIMER/EVENT COUNTERS 0 AND 1

7.1 Features

16-bit timer/event counters 0 and 1 can perform the following operations.

- Interval timer function
- PWM output
- External signal cycle measurement

7.2 Functional Overview

- 16-bit timer/counter
- Capture/compare common registers: 2 × 2 channels
- Interrupt request sources
  - Capture/match interrupt requests: 2 × 2 channels
  - Overflow interrupt requests: 1 × 2 channels
- Timer/counter count clock sources: 2
  (Selection of external pulse input or internal system clock division)
- Either free-running mode or overflow stop mode can be selected as the operation mode when the timer/counter overflows
- Timer/counter can be cleared by a match of the timer/counter and a compare register
- External pulse outputs: 1 × 2 channels

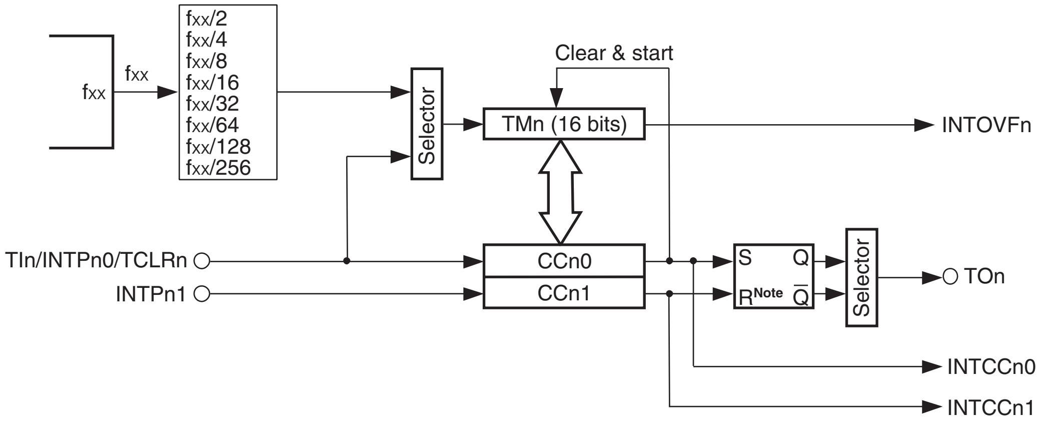
7.3 Configuration

Table 7-1. 16-Bit Timer/Event Counter Configuration

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R
TM0, TM1	fxx/2, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256	TM0	Read	INTOVF0	–	–
		CC00	Read/write	INTCC00	INTP00	TO0 (S)
		CC01	Read/write	INTCC01	INTP01	TO0 (R)
		TM1	Read	INTOVF1	–	–
		CC10	Read/write	INTCC10	INTP10	TO1 (S)
		CC11	Read/write	INTCC11	INTP11	TO1 (R)

Remark fxx: Main clock
S/R: Set/reset

Figure 7-1. Block Diagram of 16-Bit Timer/Event Counter



Note Reset priority

Remarks 1. n = 0, 1

2. fxx: Main clock

(1) Timers 0 and 1 (TM0 and TM1)

TMn functions as a 16-bit free-running timer or as an event counter for an external signal. Besides being used for cycle measurement, TMn can be used for pulse output (n = 0, 1).

TMn is read-only, in 16-bit units.

Cautions 1. The TMn register can only be read. If the TMn register is written, the subsequent operation is undefined.

2. If the TMCAEn bit of the TMCn0 register is cleared (0), a reset is performed asynchronously.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TM0																	FFFFF600H	0000H
TM1																	FFFFF610H	0000H

TMn performs the count-up operations of an internal count clock or external count clock. Timer start and stop are controlled by the TMCEn bit of timer mode control register n0 (TMCn0) (n = 0, 1).

The internal or external count clock is selected by the ETIn bit of timer mode control register n1 (TMCn1) (n = 0, 1).

(a) Selection of the external count clock

TMn operates as an event counter.

When the ETIn bit of timer mode control register n1 (TMCn1) is set (1), TMn counts the valid edges of the external clock input (TIn), synchronized with the internal count clock. The valid edge is specified by valid edge select register n (SESn) (n = 0, 1).

Caution When the INTPn0/TIn/TCLRn pin is used as TIn (external clock input pin), disable the INTPn0 interrupt and set CCn0 to compare mode (n = 0, 1).

(b) Selection of the internal count clock

TMn operates as a free-running timer.

When the internal clock is specified as the count clock by timer mode control register n1 (TMCn1), TMn is counted up for each input clock cycle specified by the CSn0 to CSn2 bits of the TMCn0 register (n = 0, 1).

Division by the prescaler can be selected for the count clock from among $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, and $f_{XX}/256$ by the TMCn0 register (f_{XX}: Internal system clock).

An overflow interrupt can be generated if the timer overflows. Also, the timer can be stopped following an overflow by setting the OSTn bit of the TMCn1 register to 1.

Caution The count clock cannot be changed while the timer is operating.

The conditions when the TMn register becomes 0000H are shown below.

(a) Asynchronous reset

- TMCAEn bit of TMCn0 register = 0
- After reset

(b) Synchronous reset

- TMCEn bit of TMCn0 register = 0
- The CCn0 register is used as a compare register, and the TMn and CCn0 registers match when clearing the TMn register is enabled (CCLRn bit of the TMCn1 register = 1)

(2) Capture/compare registers n0 and n1 (CCn0 and CCn1) (n = 0, 1)

These capture/compare registers (n0 and n1) are 16-bit registers.

They can be used as capture registers or compare registers according to the CMSn0 and CMSn1 bit specifications of timer mode control register n1 (TMCn1) (n = 0, 1).

These registers can be read or written in 16-bit units. (However, write operations can only be performed in compare mode.)

These registers are cleared to 0000H after reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CC0n																	FFFFF602H, FFFFF604H	0000H
CC1n																	FFFFF612H, FFFFF614H	0000H

Remark n = 0, 1

(a) Setting these registers as capture registers (CMSn0 and CMSn1 of TMCn1 = 0)

When these registers are set as capture registers, the valid edges of the corresponding external interrupt signals INTPn0 and INTPn1 are detected as capture triggers. The timer TMn is synchronized with the capture trigger, and the value of TMn is latched in the CCn0 and CCn1 registers (capture operation).

The valid edge of the INTPn0 pin is specified (rising, falling, or both rising and falling edges) according to the IESn01 and IESn00 bits of the SESn register, and the valid edge of the INTPn1 pin is specified according to the IESn11 and IESn10 bits of the SESn register (n = 0, 1).

The capture operation is performed asynchronously to the count clock. The latched value is held in the capture register until another capture operation is performed (n = 0, 1).

When the TMCAEn bit of timer mode control register n0 (TMCn0) is 0, 0000H is read (n = 0, 1).

If these registers are specified as capture registers, an interrupt is generated by detecting the valid edge of signals INTPn0 and INTPn1 (n = 0, 1).

Caution If the capture operation conflicts with the timing of disabling the TMn register from counting (when the TMCEn bit of the TMCn0 register = 0), the captured data becomes undefined. In addition, the INTCCn0 and INTCCn1 interrupts do not occur (n = 0, 1).

(b) Setting these registers as compare registers (CMSn0 and CMSn1 of TMCn1 = 1)

When these registers are set as compare registers, the TMn and register values are compared for each count clock, and an interrupt is generated by a match. If the CCLRn bit of timer mode control register n1 (TMCn1) is set (1), the TMn value is cleared (0) at the same time as a match with the CCn0 register (it is not cleared (0) by a match with the CCn1 register) (n = 0, 1).

Compare registers are equipped with a set/reset function. The corresponding timer output (TOn) is set or reset, in synchronization with the generation of a match signal (n = 0, 1).

The interrupt selection source differs according to the function of the selected register.

Cautions 1. When writing to capture/compare registers n0 and n1, always set the TMCAEn bit to 1 first. If the TMCAEn bit is 0, the data that is written will be invalid.

2. Write to capture/compare registers n0 and n1 after setting them as compare registers via TMCn0 and TMCn1 register settings. If they are set as capture registers (CMSn0 and CMSn1 bits of TMCn1 register = 0), no data is written even if a write operation is performed to CCn0 and CCn1.

3. When these registers are set as compare registers, INTPn0 and INTPn1 cannot be used (n = 0, 1).

7.4 Control Registers

(1) Timer mode control registers 00 and 10 (TMC00 and TMC10)

The TMCn0 registers control the operation of TMn (n = 0, 1).

These registers can be read or written in 8-bit or 1-bit units.

Be sure to set bits 3 and 2 to 0. If they are set to 1, the operation is not guaranteed.

These registers are cleared to 00H after reset.

Cautions 1. The TMCAEn bit cannot be set at the same time as the other bits. The other bits and the registers of the other TMn units should always be set after the TMCAEn bit has been set. Also, to use external pins related to the timer function when the 16-bit timer/event counter is used, be sure to set (1) the TMCAEn bit after setting the external pins to control mode.

2. When conflict occurs between an overflow and a TMCn0 register write, the OVFn bit value becomes the value written during the TMCn0 register write (n = 0, 1).

(1/2)

After reset: 00H R/W Address: TMC00 FFFFF606H TMC10 FFFFF616H

	<7>	6	5	4	3	2	<1>	<0>
TMCn0 (n = 0, 1)	OVFn	CSn2	CSn1	CSn0	0	0	TMCEn	TMCAEn

OVFn	TMn register overflow detection
0	No overflow occurs
1	Overflow occurs
When TMn has counted up from FFFFH to 0000H, the OVFn bit becomes 1 and an overflow interrupt request (INTOVFn) is generated at the same time. However, if TMn is cleared to 0000H after a match at FFFFH when the CCn0 register is set to compare mode (CMSn0 bit of TMCn1 register = 1) and clearing is enabled for a match when TMn and CCn0 are compared (CCLRn bit of TMCn1 register = 1), then TMn is considered to be cleared and the OVFn bit does not become 1. Also, no INTOVFn interrupt is generated. The OVFn bit retains the value 1 until 0 is written directly or until an asynchronous reset is performed because the TMCAEn bit is 0. An interrupt operation due to an overflow is independent of the OVFn bit, and the interrupt request flag (OVFIFn) for INTOVFn is not affected even if the OVFn bit is manipulated. If an overflow occurs while the OVFn bit is being read, the flag value changes, and the change is reflected when the next read operation occurs.	

(2/2)

CSn2	CSn1	CSn0	Internal count clock selection
0	0	0	$f_{xx}/2$
0	0	1	$f_{xx}/4$
0	1	0	$f_{xx}/8$
0	1	1	$f_{xx}/16$
1	0	0	$f_{xx}/32$
1	0	1	$f_{xx}/64$
1	1	0	$f_{xx}/128$
1	1	1	$f_{xx}/256$

<table>
<tr><th>TMCEn</th><th>TMn register operation control</th></tr>
<tr><td>0</td><td>Count disabled (stops at 0000H and does not operate).</td></tr>
<tr><td>1</td><td>Counting operation is performed.</td></tr>
<tr><td colspan="2">When TMCEn = 0, the external pulse output (TOn) becomes inactive (the active level of TOn output is set by the ALVn bit of the TMCn1 register).</td></tr>
</table>

<table>
<tr><th>TMCAEn</th><th>Internal count clock control</th></tr>
<tr><td>0</td><td>The entire TMn unit is asynchronously reset. The supply of clocks to the TMn unit stops.</td></tr>
<tr><td>1</td><td>Clocks are supplied to the TMn unit.</td></tr>
<tr><td colspan="2">• When the TMCAEn bit is set to 0, the TMn unit can be asynchronously reset.
• When TMCAEn = 0, the TMn unit is in a reset state. Therefore, to operate TMn, the TMCAEn bit must be set to 1.
• When the TMCAEn bit is changed from 1 to 0, all registers of the TMn unit are initialized. When TMCAEn is set to 1 again, the TMn unit registers must be set again.</td></tr>
</table>

(2) Timer mode control registers 01 and 11 (TMC01 and TMC11)

The TMCn1 registers control the operation of TMn (n = 0, 1).

These registers can be read or written in 8-bit units.

These registers are set to 20H after reset.

Cautions 1. The various bits of the TMCn1 register must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMCEn bit of the TMCn0 register to 0. If these bits are overwritten during timer operation, operation cannot be guaranteed (n = 0, 1).

2. If the ENTOn and ALVn bits are changed at the same time, a glitch (spike shaped noise) may be generated in the TOn pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTOn and ALVn bits do not change at the same time (n = 0, 1).

3. TOn output is not changed by an external interrupt signal (INTPn0 or INTPn1). To use the TOn signal, specify that the capture/compare registers are compare registers (CMSn0 and CMSn1 bits of TMCn1 register = 1) (n = 0, 1).

(1/2)

After reset: 20H R/W Address: TMC01 FFFFF608H TMC11 FFFFF618H

	7	6	5	4	3	2	1	0
TMCn1 (n = 0, 1)	OSTn	ENTOn	ALVn	ETIn	CCLRn	ECLRn	CMSn1	CMSn0

OSTn	Setting of operation when TMn register overflowed
0	After the overflow, counting continues (free-running mode).
1	After the overflow, the timer maintains the value 0000H, and counting stops (overflow stop mode).
When OSTn bit = 1, the TMCEn bit of TMCn0 remains at 1. Counting is restarted by writing 1 to the TMCEn bit.	

ENTOn	External pulse output (TOn) enable/disable
0	External pulse output is disabled.
1	External pulse output is enabled.
• When OSTn bit = 0, output of the ALVn bit inactive level to the TOn pin is fixed. The TOn pin level is not changed even if a match signal from the corresponding compare register is generated. • When OSTn bit = 1, a compare register match causes TOn output to change. However, if capture mode is set, TOn output does not change. The ALVn bit inactive level is output from the time when timer output is enabled until a match signal is first generated. • If either CCn0 or CCn1 is specified as a capture register, the ENTOn bit must be set to 0.	

(2/2)

ALVn	External pulse output (TOn) active level specification
0	Low level
1	High level
The initial value of the ALVn bit is 1.	

ETIn	Count clock external/internal switch specification
0	Specifies the input clock (internal).
1	Specifies the external clock (TIn0).
• When ETIn bit = 0, the clock can be selected according to the CSn2 to CSn0 bits of TMCn0. • When ETIn bit = 1, the valid edge can be selected according to the TESn1 and TESn0 bit specifications of SESn.	

CCLRn	TMn register clear enable/disable specification during compare operation
0	Clearing is disabled
1	Clearing is enabled (if CCn0 and TMn match during a compare operation, TMn is cleared)

ECLRn	TMn register clear enable/disable specification by external clear input (TCLRn)
0	Clearing is disabled
1	Clearing is enabled (after the clearing, restarts counting)

CMSn1	Capture/compare register (CCn1) operation mode selection
0	The register operates as a capture register.
1	The register operates as a compare register.

CMSn0	Capture/compare register (CCn0) operation mode selection
0	The register operates as a capture register.
1	The register operates as a compare register.

Remark A reset takes precedence for the flip-flop of the TOn output (n = 0, 1).

(3) Valid edge select registers 0 and 1 (SES0 and SES1)

These registers specify the valid edge of an external interrupt request (INTP00, INTP01, INTP10, INTP11, TI0, TI1, TCLR10, and TCLR11) from an external pin.

The rising edge, the falling edge, or both rising and falling edges can be specified as the valid edge independently for each pin.

Each of these registers can be read or written in 8-bit units.

These registers are cleared to 00H after reset.

Caution The various bits of the SESn register must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMCEn bit of the TMCn0 register to 0. If the SESn register is overwritten during timer operation, operation cannot be guaranteed.

After reset: 00H R/W Address: SES0 FFFFF609H SES1 FFFFF619H

	7	6	5	4	3	2	1	0
SESn (n = 0, 1)	TESn1	TESn0	CESn1	CESn0	IESn11	IESn10	IESn01	IESn00

TESn1	TESn0	Valid edge of TIn pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

CESn1	CESn0	Valid edge of TCLRn pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

IESn11	IESn10	Valid edge of INTPn1 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

IESn01	IESn00	Valid edge of INTPn0 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

7.5 Operation

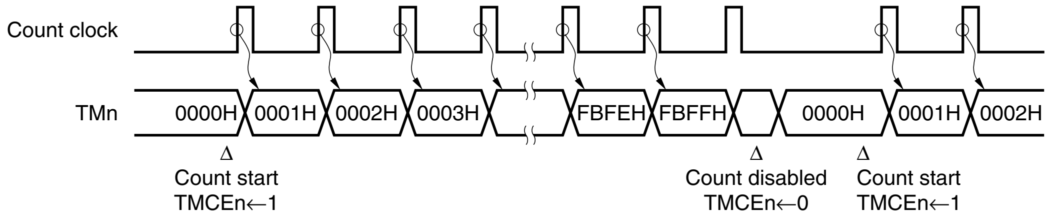
(1) Count operation

The 16-bit timer/event counter can function as a 16-bit free-running timer or as an external signal event counter. The setting for the type of operation is specified by timer mode control registers n0 and n1 (TMCn0 and TMCn1) (n = 0, 1).

When it operates as a free-running timer, if the CCn0 or CCn1 register and the TMn register count value match, an interrupt signal is generated and the timer output signal (TOn) can be set or reset. Also, a capture operation that holds the TMn register count value in the CCn0 or CCn1 register is performed, in synchronization with the valid edge that was detected from the external interrupt request input pin as an external trigger. The capture value is held until the next capture trigger is generated.

Caution When using the INTPn0/TIn0 pin as an external clock input pin (TIn0), be sure to disable the INTPn0 interrupt and set CCn0 to compare mode (n = 0, 1).

Figure 7-2. Basic Operation of 16-Bit Timer/Event Counter



Remark n = 0, 1

(2) Overflow

When the TMn register has counted the count clock from FFFFH to 0000H, the OVFn bit of the TMCn0 register is set (1), and an overflow interrupt (INTOVFn) is generated at the same time (n = 0, 1). However, if the CCn0 register is set to compare mode (CMSn0 bit = 1) and to the value FFFFH when match clearing is enabled (CCLRn bit = 1), then the TMn register is considered to be cleared and the OVFn bit is not set (1) when the TMn register changes from FFFFH to 0000H. Also, the overflow interrupt (INTOVFn) is not generated .

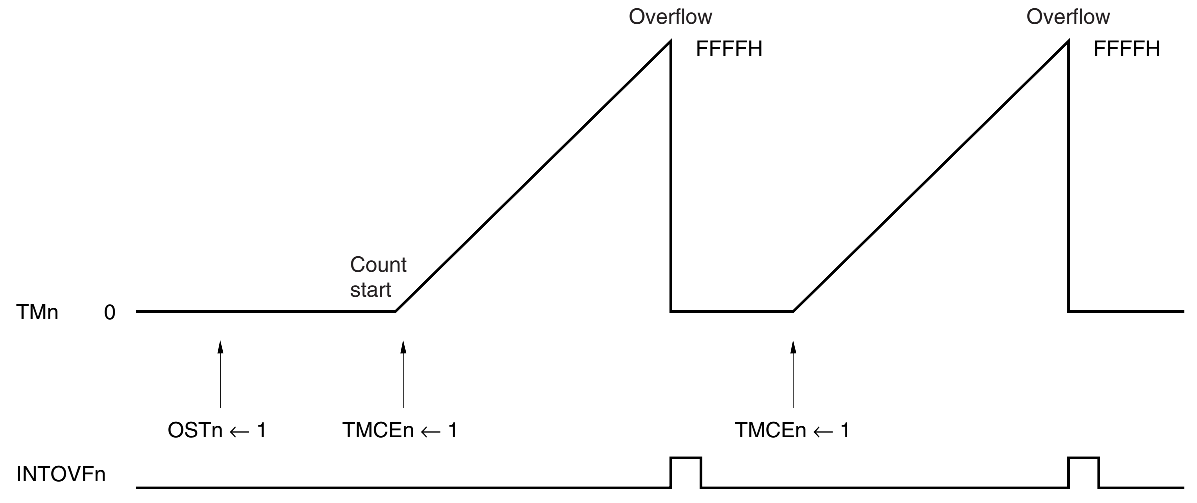
When the TMn register is changed from FFFFH to 0000H because the TMCEn bit changes from 1 to 0, the TMn register is considered to be cleared, but the OVFn bit is not set (1) and no INTOVFn interrupt is generated.

Also, timer operation can be stopped after an overflow by setting the OSTn bit of the TMCn1 register to 1. When the timer is stopped due to an overflow, the count operation is not restarted until the TMCEn bit of the TMCn0 register is set (1).

Operation is not affected even if the TMCEn bit is set (1) during a count operation.

Remark n = 0, 1

Figure 7-3. Operation After Overflow (When OSTn = 1)



Remark n = 0, 1

(3) Capture operation

The TMn register has two capture/compare registers. These are the CCn0 register and the CCn1 register. A capture operation or a compare operation is performed according to the settings of both the CMSn1 and CMSn0 bits of the TMCn1 register. If the CMSn1 and CMSn0 bits of the TMCn1 register are set to 0, the register operates as a capture register.

A capture operation that captures and holds the TMn count value asynchronously to the count clock is performed in synchronization with an external trigger. The valid edge that is detected from an external interrupt request input pin (INTPn0 or INTPn1) is used as an external trigger (capture trigger). The TMn count value during counting is captured and held in the capture register, in synchronization with that capture trigger signal. The capture register value is held until the next capture trigger is generated.

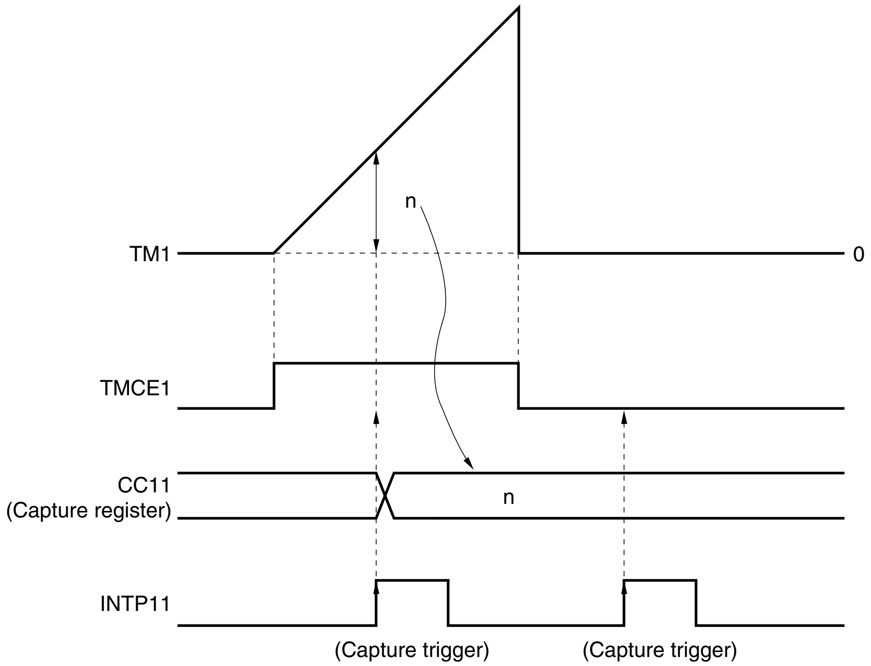
Also, an interrupt request (INTCCn0 or INTCCn1) is generated by INTPn0 or INTPn1 signal input.

The valid edge of the capture trigger is set by valid edge select register n (SESn).

If both the rising and falling edges are set as capture triggers, the input pulse width from an external source can be measured. Also, if only one of the edges is set as the capture trigger, the input pulse cycle can be measured.

Remark n = 0, 1

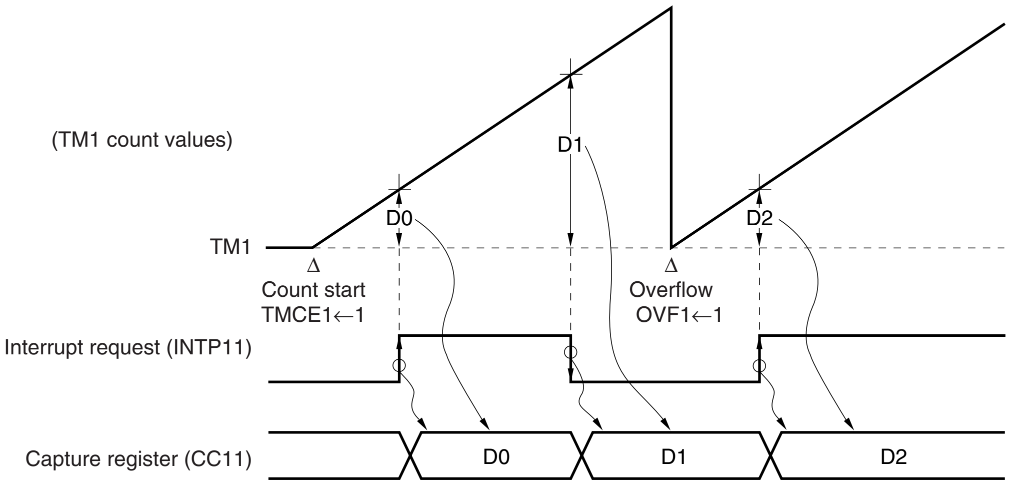
Figure 7-4. Capture Operation Example (TM1)



Remarks 1. When the TMCE1 bit is 0, no capture operation is performed even if INTP11 is input.

2. Valid edge of INTP11: Rising edge

Figure 7-5. TM1 Capture Operation Example (When Both Edges Are Specified)



Remark D0 to D2: TM1 register count values

(4) Compare operation

The TMn register has two capture/compare registers. These are the CCn0 register and the CCn1 register. A capture operation or a compare operation is performed according to the settings of both the CMSn1 and CMSn0 bits of the TMCn1 register. If the CMSn1 and CMSn0 bits of the TMCn1 register are set to 1, the register operates as a compare register.

A compare operation that compares the value that was set in the compare register and the TMn register count value is performed.

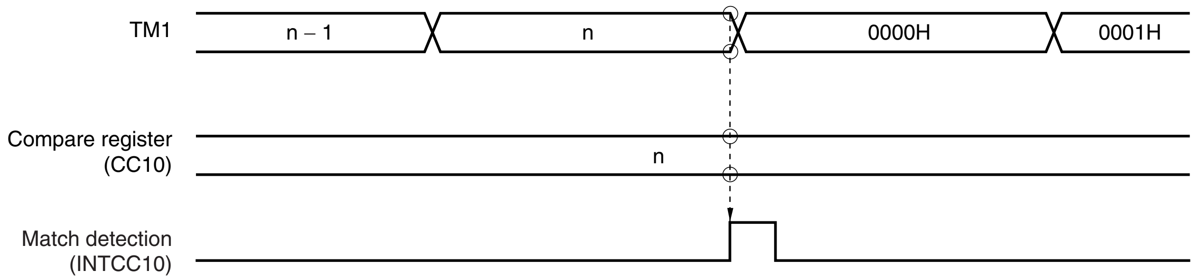
If the TMn register count value matches the value of the compare register, which had been set in advance, a match signal is sent to the output controller. The match signal causes the timer output pin (TOn) to change and an interrupt request signal (INTCCnn) to be generated at the same time.

If the CCn0 or CCn1 registers are set to 0000H, the 0000H after the TMn register counts up from FFFFH to 0000H is judged as a match. In this case, the TMn register value is cleared (0) at the next count timing, however, this 0000H is not judged as a match. Also, the 0000H when the TMn register begins counting is not judged as a match.

If match clearing is enabled (CCLRn bit = 1) for the CCn0 register, the TMn register is cleared when a match with the TMn register occurs during a compare operation.

Remark n = 0, 1

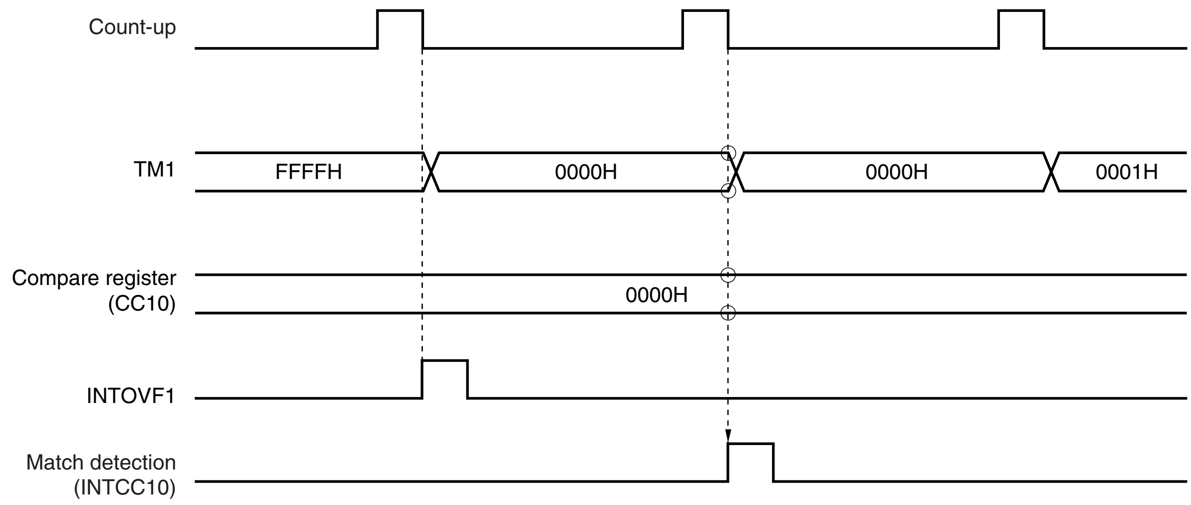
Figure 7-6. Compare Operation Example (When CCLR1 = 1 and CC10 Is Other Than 0000H)



Remarks 1. A match is detected immediately after the count-up, and the match detection signal is generated.

2. n ≠ 0000H

Figure 7-7. Compare Operation Example (When CCLR1 = 1 and CC10 Is 0000H)



Remark A match is detected immediately after the count-up, and the match detection signal is generated.

(5) External pulse output

The 16-bit timer/event counter has two timer output pins (TOn).

An external pulse output (TOn) is generated when a match of the two compare registers (CCn0 and CCn1) and the TMn register is detected.

If a match is detected when the TMn register count value and the CCn0 value are compared, the output level of the TOn pin is set. Also, if a match is detected when the TMn register count value and the CCn1 value are compared, the output level of the TOn pin is reset.

The output level of the TOn pin can be specified by the TMCn1 register.

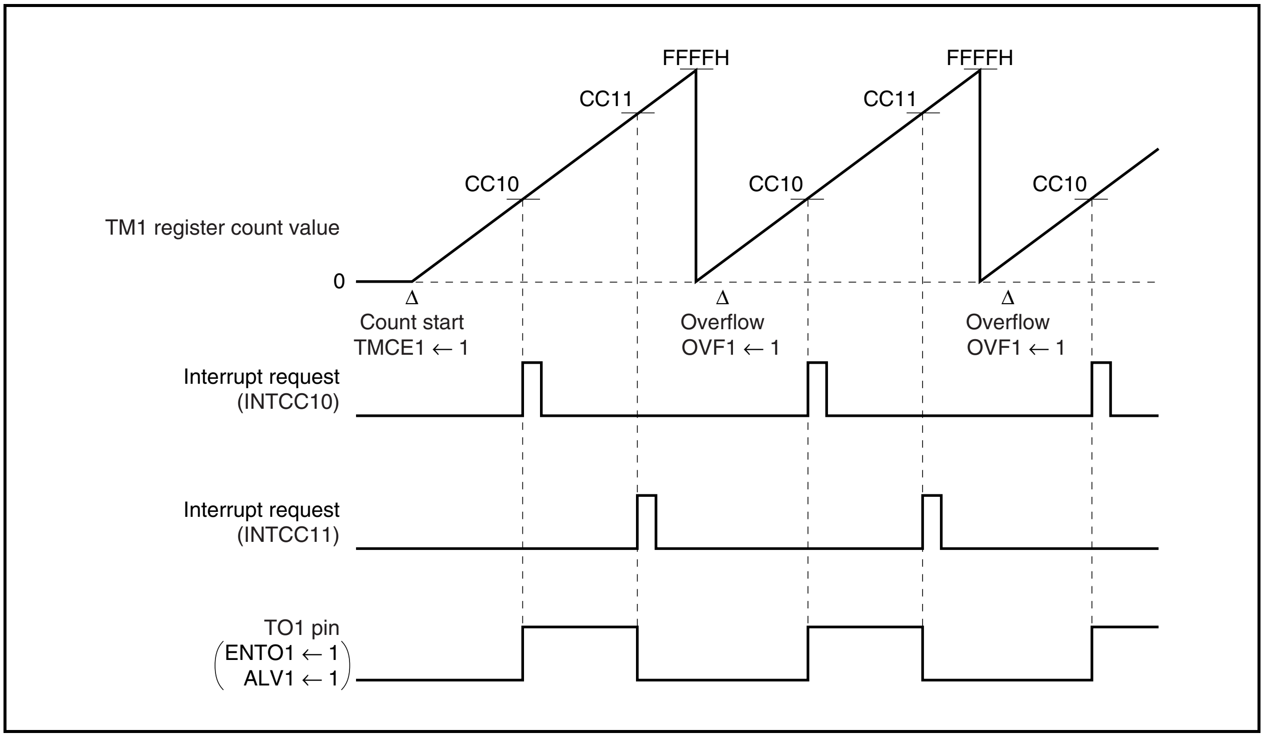
Remark n = 0, 1

Table 7-2. TOn Output Control

ETIn	ALVn	TOn Output	
		External Pulse Output	Output Level
0	0	Disable	High level
0	1	Disable	Low level
1	0	Enable	When the CCn0 register is matched: low level When the CCn1 register is matched: high level
1	1	Enable	When the CCn0 register is matched: high level When the CCn1 register is matched: low level

Remark n = 0, 1

Figure 7-8. TM1 Compare Operation Example (Set/Reset Output Mode)



7.6 Application Examples

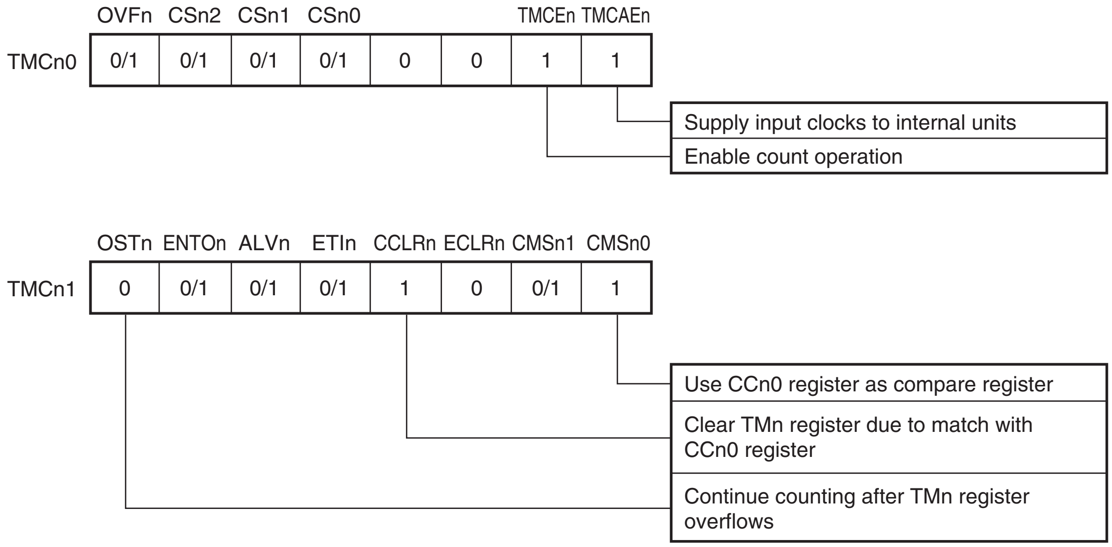
(1) Interval timer

By setting the TMCn0 and TMCn1 registers as shown in Figure 7-9, the 16-bit timer/event counter operates as an interval timer that repeatedly generates interrupt requests with the value that was preset in the CCn0 register as the interval.

When the counter value of the TMn register matches the setting value of the CCn0 register, the TMn register is cleared (0000H) and an interrupt request signal (INTCCn0) is generated at the same time that the count operation resumes.

Remark n = 0, 1

Figure 7-9. Contents of Register Settings When 16-Bit Timer/Event Counter Is Used as Interval Timer



	OVFn	CSn2	CSn1	CSn0			TMCEn	TMCAEn
TMCn0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units
Enable count operation

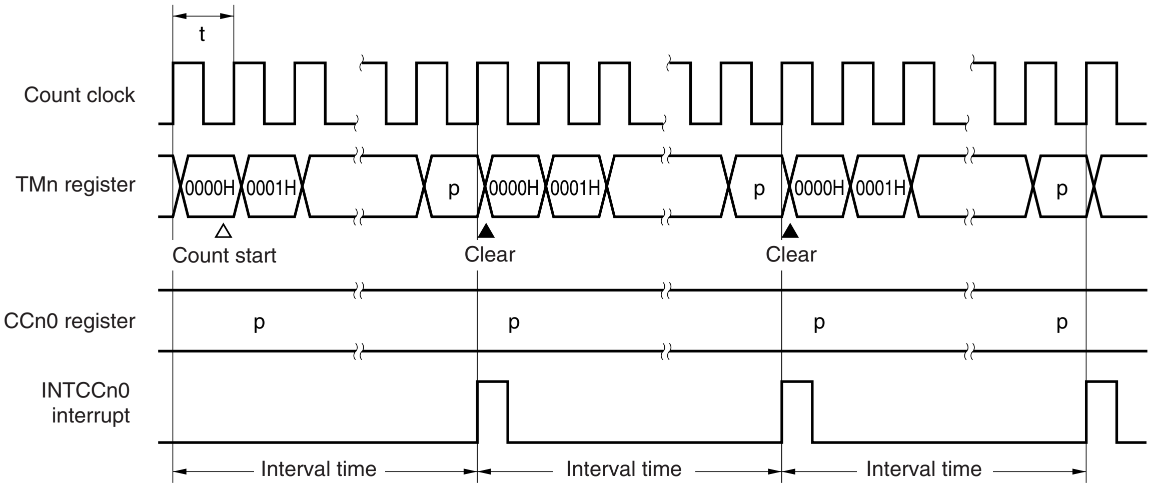
	OSTn	ENTOn	ALVn	ETIn	CCLRn	ECLRn	CMSn1	CMSn0
TMCn1	0	0/1	0/1	0/1	1	0	0/1	1

Use CCn0 register as compare register
Clear TMn register due to match with CCn0 register
Continue counting after TMn register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 7-10. Interval Timer Operation Timing Example



Remarks 1. p: Setting value of CCn0 register (0000H to FFFFH)

t: Count clock cycle

Interval time = (p + 1) × t

2. n = 0, 1

(2) PWM output

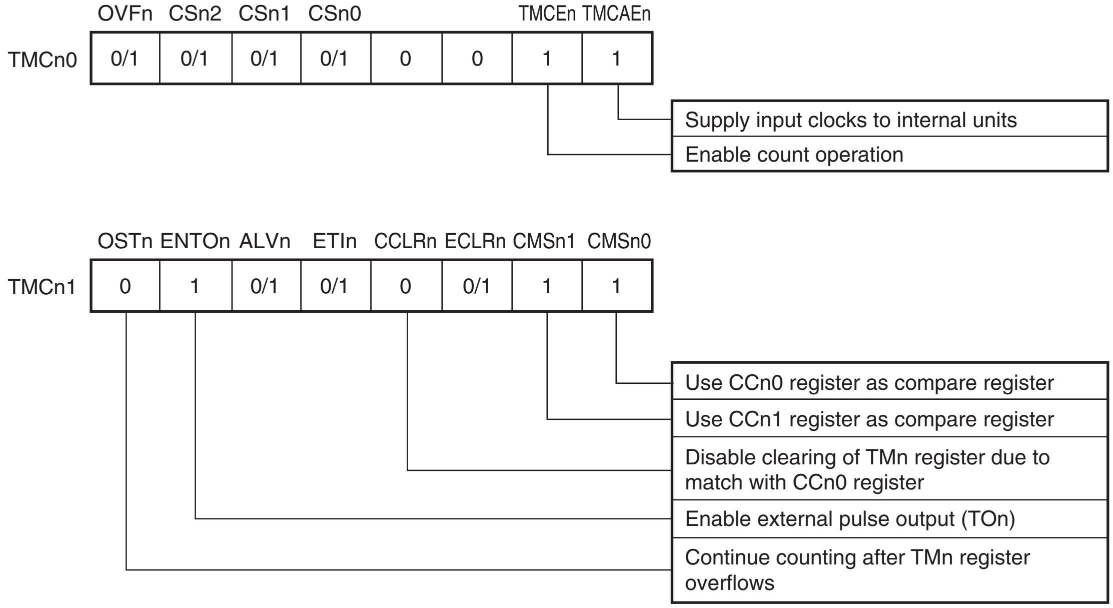
By setting the TMCn0 and TMCn1 registers as shown in Figure 7-11, the 16-bit timer/event counter can output a PWM signal, whose frequency is determined according to the setting of the CSn2 to CSn0 bits of the TMCn0 register, with the values that were preset in the CCn0 and CCn1 registers determining the intervals.

When the counter value of the TMn register matches the setting value of the CCn0 register, the TOn output becomes active. Then, when the counter value of the TMn register matches the setting value of the CCn1 register, the TOn output becomes inactive. The TMn register continues counting. When it overflows, its count value is cleared to 0000H, and the register continues counting. In this way, a PWM signal whose frequency is determined according to the setting of the CSn2 to CSn0 bits of the TMCn0 register can be output. When the setting value of the CCn0 register and the setting value of the CCn1 register are the same, the TOn output remains inactive and does not change.

The active level of the TOn output can be set by the ALVn bit of the TMCn1 register.

Remark n = 0, 1

Figure 7-11. Contents of Register Settings When 16-Bit Timer/Event Counter Is Used for PWM Output



	OVFn	CSn2	CSn1	CSn0			TMCEn	TMCAEn
TMCn0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units
Enable count operation

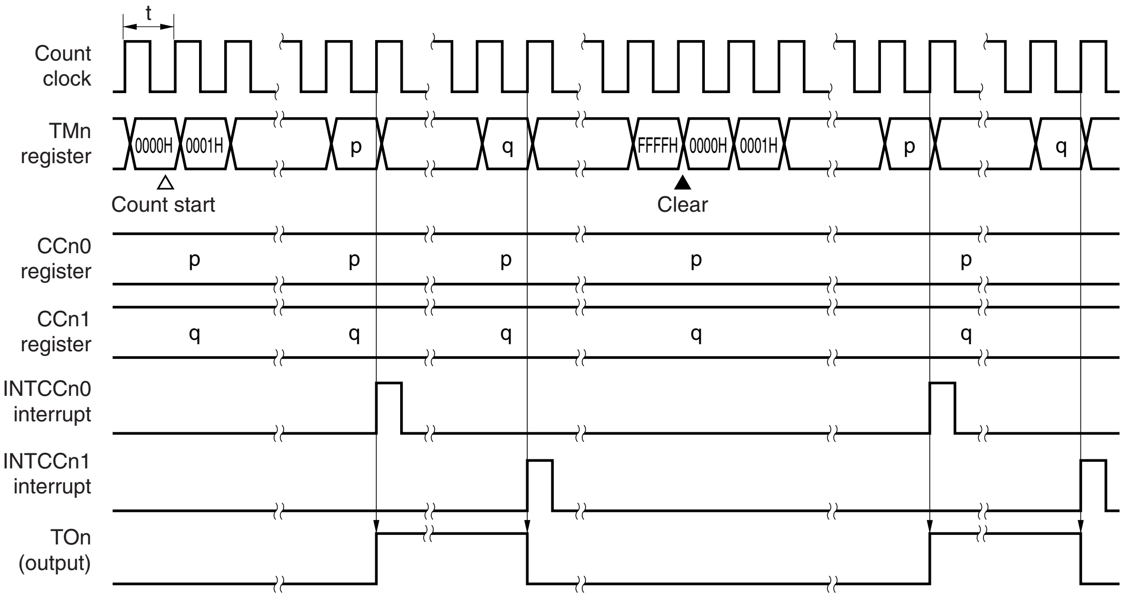
	OSTn	ENTOn	ALVn	ETIn	CCLRn	ECLRn	CMSn1	CMSn0
TMCn1	0	1	0/1	0/1	0	0/1	1	1

Use CCn0 register as compare register
Use CCn1 register as compare register
Disable clearing of TMn register due to match with CCn0 register
Enable external pulse output (TOn)
Continue counting after TMn register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 7-12. PWM Output Timing Example



Remarks 1. p: Setting value of CCn0 register (0000H to FFFFH)

q: Setting value of CCn1 register (0000H to FFFFH)

$p \neq q$

t: Count clock cycle

PWM cycle = 65,536 × t

$$\text{Duty} = \frac{q - p}{65{,}536}$$

2. In this example, the active level of the TOn output is set to the high level.

3. n = 0, 1

(3) One-shot pulse output

By setting the TMCn0 and TMCn1 registers as shown in Figure 7-13, the 16-bit timer/event counter can output a one-shot pulse from the TOn pin by using the valid edge of the TCLRn pin as an external trigger.

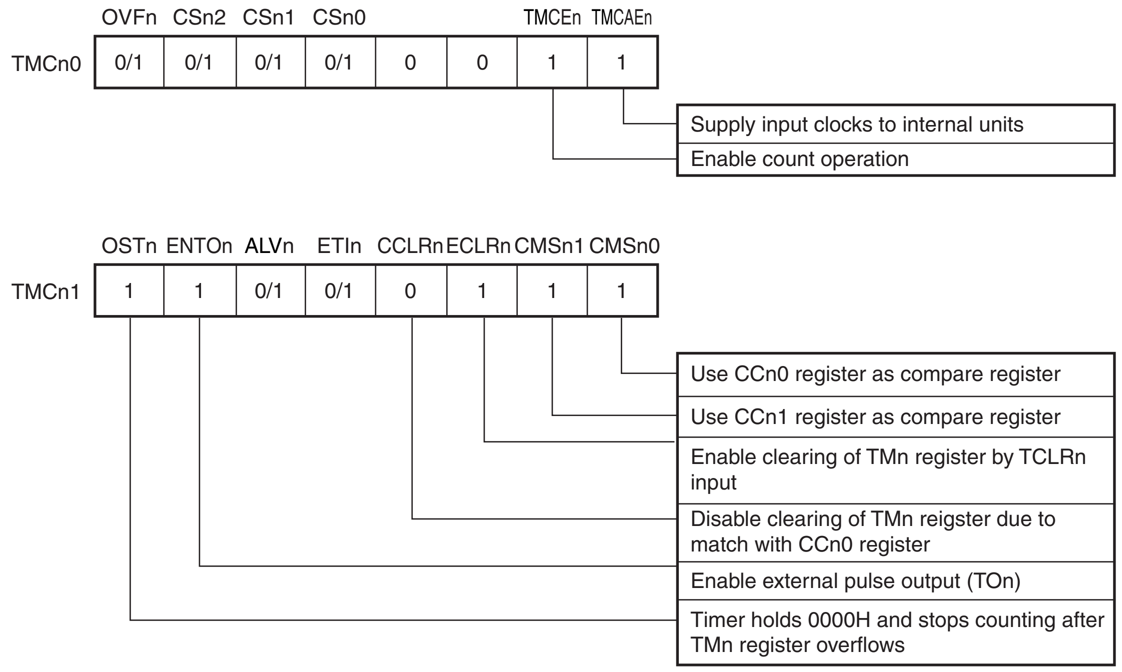
The valid edge of the TCLRn pin is selected according to the CESn0 and CESn1 bits of the SESn register. The rising edge, falling edge, or both rising and falling edges can be selected as the valid edge of both pins.

The TMn register is cleared and started by setting a valid edge to the TCLRn pin. TOn output becomes active at the count value set in advance to the CCn0 register. After that, the TOn output becomes inactive at the count value set in advance to CCn1 register. The active level of the TOn output can be set by the ALVn bit of the TMCn1 register. When the setting value of the CCn0 register and the setting value of the CCn1 register are the same, the TOn output remains inactive and does not change.

The active level of the TOn output can be set by the ALVn bit of the TMCn1 register.

Remark n = 0, 1

Figure 7-13. Contents of Register Settings
When 16-Bit Timer/Event Counter Is Used for One-Shot Pulse Output



	OVFn	CSn2	CSn1	CSn0			TMCEn	TMCAEn
TMCn0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units

Enable count operation

	OSTn	ENTOn	ALVn	ETIn	CCLRn	ECLRn	CMSn1	CMSn0
TMCn1	1	1	0/1	0/1	0	1	1	1

Use CCn0 register as compare register

Use CCn1 register as compare register

Enable clearing of TMn register by TCLRn input

Disable clearing of TMn reigster due to match with CCn0 register

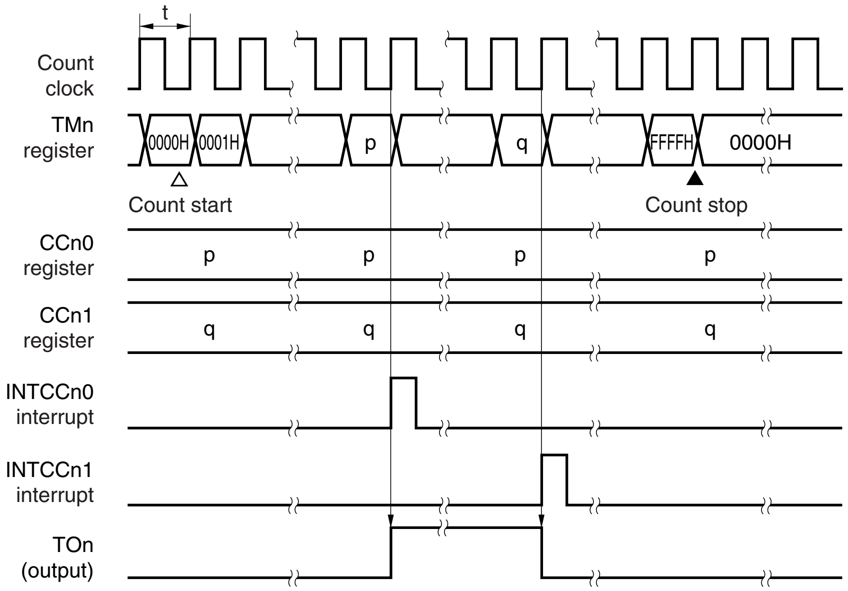
Enable external pulse output (TOn)

Timer holds 0000H and stops counting after TMn register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 7-14. One-Shot Pulse Output Operation Timing Example



Remarks 1. p: Setting value of CCn0 register (0000H to FFFFH)

q: Setting value of CCn1 register (0000H to FFFFH)

p ≠ q

t: Count clock cycle

2. In this example, the valid edge of the TCLRn input is set to the rising edge and the active level of the TOn output is set to the high level.

3. n = 0, 1

(4) Cycle measurement

By setting the TMCn0 and TMCn1 registers as shown in Figure 7-15, the 16-bit timer/event counter can measure the cycle of signals input to the INTPn0 or INTPn1 pin.

The valid edge of the INTPn0 pin is selected according to the IESn01 and IESn00 bits of the SESn register, and the valid edge of the INTPn1 pin is selected according to the IESn11 and IESn10 bits of the SESn register. Either the rising edge, the falling edge, or both edges can be selected as the valid edges of both pins.

If the CCn0 register is set as a capture register, the valid edge input of the INTPn0 pin is set as the trigger for capturing the TMn register value in the CCn0 register. When this value is captured, an INTCCn0 interrupt is generated.

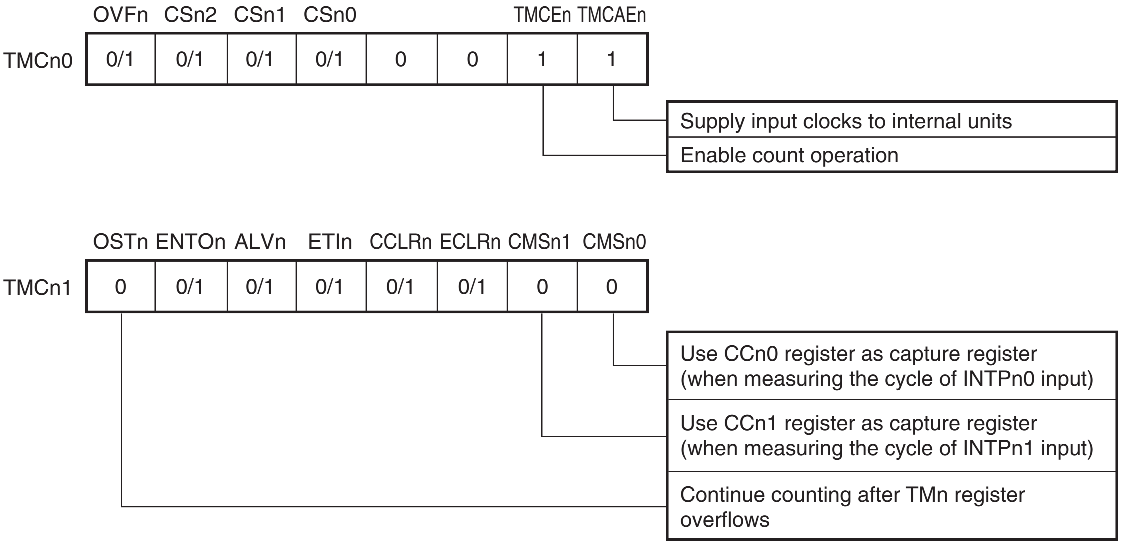
Similarly, if the CCn1 register is set as a capture register, the valid edge input of the INTPn1 pin is set as the trigger for capturing the TMn register value in the CCn1 register. When this value is captured, an INTCCn1 interrupt is generated.

The cycle of signals input to the INTPn0 pin is calculated by obtaining the difference between the TMn register's count value (Dx) that was captured in the CCn0 register according to the x-th valid edge input of the INTPn0 pin and the TMn register's count value (D(x+1)) that was captured in the CCn0 register according to the (x+1)-th valid edge input of the INTPn0 pin and multiplying the value of this difference by the cycle of the internal count clock.

The cycle of signals input to the INTPn1 pin is calculated by obtaining the difference between the TMn register's count value (Dx) that was captured in the CCn1 register according to the x-th valid edge input of the INTPn1 pin and the TMn register's count value (D(x+1)) that was captured in the CCn1 register according to the (x+1)-th valid edge input of the INTPn1 pin and multiplying the value of this difference by the cycle of the internal count clock.

Remark n = 0, 1

Figure 7-15. Contents of Register Settings When 16-Bit Timer/Event Counter Is Used for Cycle Measurement



	OVFn	CSn2	CSn1	CSn0			TMCEn	TMCAEn
TMCn0	0/1	0/1	0/1	0/1	0	0	1	1

Supply input clocks to internal units
Enable count operation

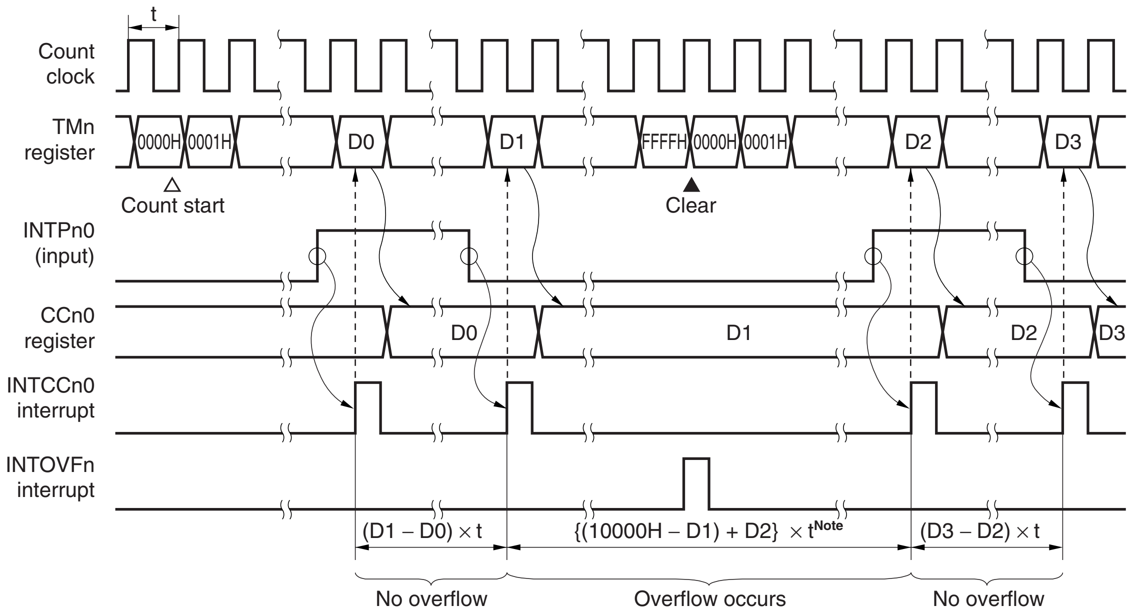
	OSTn	ENTOn	ALVn	ETIn	CCLRn	ECLRn	CMSn1	CMSn0
TMCn1	0	0/1	0/1	0/1	0/1	0/1	0	0

Use CCn0 register as capture register (when measuring the cycle of INTPn0 input)
Use CCn1 register as capture register (when measuring the cycle of INTPn1 input)
Continue counting after TMn register overflows

Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0, 1

Figure 7-16. Cycle Measurement Operation Timing Example



Note When an overflow occurs once.

Remarks 1. D0 to D3: TMn register count values

t: Count clock cycle

2. In this example, the valid edge of the INTPn0 input has been set to both edges (rising and falling).

3. n = 0, 1

7.7 Cautions

Various cautions concerning the 16-bit timer/event counter are shown below.

(1) If a conflict occurs between the reading of the CCn0 register and a capture operation when the CCn0 register is used in capture mode, an external trigger (INTPn0) valid edge is detected and an external interrupt request signal (INTCCn0) is generated, however, the timer value is not stored in the CCn0 register.

(2) If a conflict occurs between the reading of the CCn1 register and a capture operation when the CCn1 register is used in capture mode, an external trigger (INTPn1) valid edge is detected and an external interrupt request signal (INTCCn1) is generated, however, the timer value is not stored in the CCn1 register.

(3) The following bits and registers must not be rewritten during operation (TMCEn = 1).
- CSn2 to CSn0 bits of TMCn0 register
- TMCn1 register
- SESn register

(4) The TMCAEn bit of the TMCn0 register is a TMn reset signal. To use TMn, first set (1) the TMCAEn bit.

(5) The analog noise elimination time + two cycles of the input clock are required to detect the valid edge of the external interrupt request signal (INTPn0 or INTPn1) or the external clock input (TIn). Therefore, edge detection will not be performed normally for changes that are less than the analog noise elimination time + two cycles of the input clock. Only two clocks of fxx are required for detection of the internal clear input (TCLR1n) valid edge.

(6) The operation of an external interrupt request signal (INTCCn0 or INTCCn1) is automatically determined according to the operating state of the capture/compare register. When the capture/compare register is used for a capture operation, the external interrupt request signal is used for valid edge detection. When the capture/compare register is used for a compare operation, the external interrupt request signal is used for an interrupt indicating a match with the TMn register.

(7) If the ENTOn and ALVn bits are changed at the same time, a glitch (spike shaped noise) may be generated in the TOn pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTOn and ALVn bits are not changed at the same time.

Remark n = 0, 1

CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 2 TO 5

8.1 Functional Overview

8-bit timer/event counter n has the following two modes (n = 2 to 5).

- Mode using 8-bit timer/event counter alone (individual mode)
- Mode using cascade connection (16-bit resolution: cascade connection mode)

These two modes are described below.

(1) Mode using 8-bit timer/event counter alone (individual mode)
8-bit timer/event counter n operates as an 8-bit timer/event counter.
The following functions can be used.

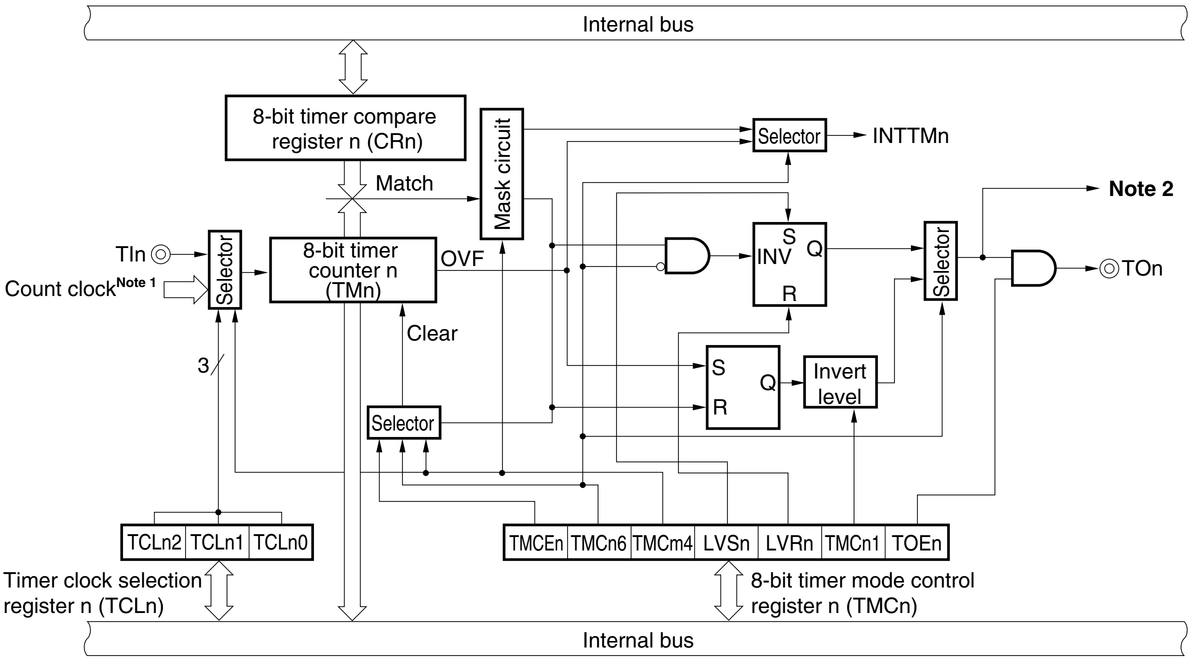
- Interval timer
- External event counter
- Square wave output
- PWM output

(2) Mode using cascade connection (16-bit resolution: cascade connection mode)
TM2 and TM3, and TM4 and TM5 can be used as 16-bit timer/event counters when they are connected in cascade. The following functions can be used.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

The block diagram of 8-bit timer/event counter n is shown next.

Figure 8-1. Block Diagram of 8-Bit Timer/Event Counter n



Notes 1. The count clock is set by the TCLn register.

2. Serial interface clock (TM4, TM5 only)

Remarks 1. “——◎” are signals that can be directly connected to ports.

2. n = 2 to 5, m = 3, 5

8.2 Configuration

8-bit timer/event counter n consists of the following hardware (n = 2 to 5).

Table 8-1. Configuration of 8-Bit Timer/Event Counter n

Item	Configuration
Timer registers	8-bit timer counters 2 to 5 (TM2 to TM5) 16-bit timer counters 23 and 45 (TM23, TM45): Only when using cascade connection
Registers	8-bit timer compare registers 2 to 5 (CR2 to CR5) 16-bit timer compare registers 23 and 45 (CR23, CR45): Only when using cascade connection
Timer output	TO2 to TO5
Control registers[Note]	Timer clock selection registers 2 to 5 (TCL2 to TCL5) Timer clock selection registers 23 and 45 (TCL23, TCL45): Only when using cascade connection 8-bit timer mode control registers 2 to 5 (TMC2 to TMC5) 16-bit timer mode control registers 23 and 45 (TMC23, TMC45): Only when using cascade connection

Note When using the functions of the TIn and TOn pins, refer to **Table 4-18 Using Alternate Function of Port Pins**.

(1) 8-bit timer counters 2 to 5 (TM2 to TM5)

The TMn register is an 8-bit read-only register that counts the count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

TM2 and TM3, and TM4 and TM5 can be used as 16-bit timers when they are connected in cascade. When these timers are used as 16-bit timers, their values can be read by using a 16-bit memory manipulation instruction. However, because these registers are connected by an internal 8-bit bus, the TMm register and TMm+1 register must be read divided into two times. Therefore, read these registers twice and compare the values, taking into consideration that the reading occurs during a count change.

In the following cases, the count value becomes 00H.

- After reset
- When the TMCEn bit of 8-bit timer mode control register n (TMCn) is cleared
- TMn register and CRn register match in the mode in which clear & start occurs on a match between the TMn register and 8-bit timer compare register n (CRn)

Caution When connected in cascade, these registers become 00H even when the TCEm bit in the lowest timer (TMm) is cleared.

Remark n = 2 to 5
m = 2, 4

(2) 8-bit timer compare registers 2 to 5 (CR2 to CR5)

The CRn register can be read and written in 8-bit units.

In a mode other than the PWM mode, the value set to the CRn register is always compared to the count value of 8-bit counter n (TMn), and if the two values match, an interrupt request signal (INTTMn) is generated.

In the PWM mode, TMn register overflow causes the TOn pin output to change to the active level, and when the values of the TMn register and the CRn register match, the TOn pin output changes to the inactive level.

The value of the CRn register can be set in the range of 00H to FFH.

When TM2 and TM3, and TM4 and TM5 are connected in cascade as 16-bit timers, the CRm register and CRm+1 register function as 16-bit timer compare registers 23 and 45 (CR23 and CR45). The counter value and register value are compared in 16-bit lengths, and if they match, an interrupt request (INTTMm) is generated.

Cautions 1. In the mode in which clear & start occurs upon a match of the TMn register and CRn register (TMCn6 =0), do not write a different value to the CRn register during the count operation.

2. In the PWM mode, set the CRn register rewrite interval to three or more count clocks (clock selected with timer clock selection register n (TCLn)).

3. Before changing the value of the CRn register when using a cascade connection, be sure to stop the timer operation.

Remark n = 2 to 5
m = 2, 4

8.3 Control Registers

The following two registers are used to control 8-bit timer/event counter n.

- Timer clock selection register n (TCLn)
- 8-bit timer mode control register n (TMCn)

Remark To use the functions of the TIn and TOn pins, refer to **Table 4-18 Using Alternate Function of Port Pins**.

(1) Timer clock selection registers 2 to 5 (TCL2 to TCL5)

These registers set the count clock of 8-bit timer/event counter n and the valid edge of the TIn pin input.
The TCLn register can be read or written in 8-bit units.
These registers are cleared to 00H after reset.

(a) Timer clock selection registers 2 and 3 (TCL2 and TCL3)

After reset: 00H R/W Address: TCL2 FFFFF644H, TCL3 FFFFF645H

	7	6	5	4	3	2	1	0
TCLn (n = 2, 3)	0	0	0	0	0	TCLn2	TCLn1	TCLn0

TCLn2	TCLn1	TCLn0	Count clock selection		
			Clock	fxx	
				20 MHz	10 MHz
0	0	0	Falling edge of TIn	–	–
0	0	1	Rising edge of TIn	–	–
0	1	0	fxx/4	200 ns	400 ns
0	1	1	fxx/8	400 ns	800 ns
1	0	0	fxx/16	800 ns	1.6 μs
1	0	1	fxx/32	1.6 μs	3.2 μs
1	1	0	fxx/128	6.4 μs	12.8 μs
1	1	1	fxx/512	25.6 μs	51.2 μs

Cautions 1. Before overwriting the TCLn register with different data, stop the timer operation.

2. TI2 and TI3 are used alternately as P01/INTP0 and P02/INTP2, respectively, so when using the TIn pin function, set the PMC01 or PMC02 bit of the PMC0 register to 1 before starting timer operation. Edge detection may not be correctly performed if the bit is manipulated after the timer starts operating.

Remark When TCL2 and TCL3 are connected in cascade, the TCL3 register settings are invalid.

(b) Timer clock selection registers 4 and 5 (TCL4 and TCL5)

After reset: 00H R/W Address: TCL4 FFFFF654H, TCL5 FFFFF655H

	7	6	5	4	3	2	1	0
TCLn (n = 4, 5)	0	0	0	0	0	TCLn2	TCLn1	TCLn0

TCLn2	TCLn1	TCLn0	Count clock selection		
			Clock	f_{XX}	
				20 MHz	10 MHz
0	0	0	Falling edge of TIn	–	–
0	0	1	Rising edge of TIn	–	–
0	1	0	$f_{XX}/4$	200 ns	400 ns
0	1	1	$f_{XX}/8$	400 ns	800 ns
1	0	0	$f_{XX}/16$	800 ns	1.6 μs
1	0	1	$f_{XX}/32$	1.6 μs	3.2 μs
1	1	0	$f_{XX}/128$	6.4 μs	12.8 μs
1	1	1	$f_{XX}/256$	12.8 μs	25.6 μs

Cautions 1. Before overwriting the TCLn register with different data, stop the timer operation.

2. TI4 and TI5 are used alternately as P03/INTP3 and P04/INTP4, respectively, so when using the TIn pin function, set the PMC03 or PMC04 bit of the PMC0 register to 1 before starting timer operation. Edge detection may not be correctly performed if the bit is manipulated after the timer starts operating.

Remark When TCL4 and TCL5 are connected in cascade, the TCL5 register settings are invalid.

(2) 8-bit timer mode control registers 2 to 5 (TMC2 to TMC5)

The TMCn register performs the following six settings.

- Controls counting by 8-bit timer counters 2 to 5 (TM2 toTM5)
- Selects the operation mode of the TMn register
- Selects the individual mode or cascade connection mode
- Sets the status of the timer output flip-flop
- Controls the timer output flip-flop or selects the active level in the PWM (free-running) mode
- Controls timer output

The TMCn register can be read or written in 8-bit or 1-bit units.
These registers are cleared to 00H after reset.

Remark n = 2 to 5

After reset: 00H R/W Address: TMC3 FFFFF646H TMC3 FFFFF647H
TMC4 FFFFF656H TMC5 FFFFF657H

	<7>	6	5	4	3	2	1	0
TMCn (n = 2 to 5, m = 3, 5)	TMCEn	TMCn6	0	TMCm4[Note]	LVSn	LVRn	TMCn1	TOEn

TMCEn	Control of count operation of 8-bit timer/event counter n
0	Counting is disabled after the counter is cleared to 0 (counter disabled)
1	Start count operation

TMCn6	Selection of operation mode of 8-bit timer/event counter n
0	Mode in which clear & start occurs on match between TMn register and CRn register
1	PWM (free-running) mode

TMCm4	Selection of individual mode or cascade connection mode
0	Individual mode
1	Cascade connection mode (connected with TM2 or TM4)

LVSn	LVRn	Setting of status of timer output F/F
0	0	Unchanged
0	1	Reset timer output F/F to 0
1	0	Set timer output F/F to 1
1	1	Setting prohibited

TMCn1	Other than PWM (free-running) mode (TMCn6 = 0)	PWM (free-running) mode (TMCn6 = 1)
	Controls timer F/F	Selects active level
0	Disable inversion operation	High active
1	Enable inversion operation	Low active

TOEn	Timer output control
0	Disable output (TOn pin is low level)
1	Enable output

Note Bit 4 of the TMC2 and TMC4 registers is fixed to 0.

Cautions 1. The LVSn and LVRn bit settings are valid in modes other than the PWM mode.

2. Do not rewrite the TMCn1 bit and TOEn bit at the same time.

3. When switching to the PWM mode, do not rewrite the TMCn6 bit and the LVSn and LVRn bits at the same time.

4. Before rewriting the TMCn6 bit or TMCm4 bit, stop the timer operation.

Remarks 1. In the PWM mode, the PWM output is set to the inactive level by TMCEn = 0.

2. When the LVSn and LVRn bits are read, 0 is read.

3. The values of the TMCn6, LVSn, LVRn, TMCn1, and TOEn bits are reflected to the TOn output regardless of the TMCEn value.

8.4 Operation

8.4.1 Operation as interval timer (8 bits)

8-bit timer/event counter n operates as an interval timer that repeatedly generates interrupts at the interval of the count value preset in 8-bit timer compare register n (CRn).

If the count value in 8-bit timer counter n (TMn) matches the value set in the CRn register, the value of the TMn register is cleared to 0 and counting is continued, and at the same time, an interrupt request signal (INTTMn) is generated.

Setting method

<1> Set each register.
- TCLn register: Selects the count clock (t).
- CRn register: Compare value (N)
- TMCn register: Stops count operation and selects the mode in which clear & start occurs on a match between the TMn register and CRn register (TMCn register = 0000xxx0B, ×: don't care).

<2> When the TMCEn bit of the TMCn register is set to 1, the count operation starts.

<3> When the values of the TMn register and CRn register match, INTTMn is generated (TMn register is cleared to 00H).

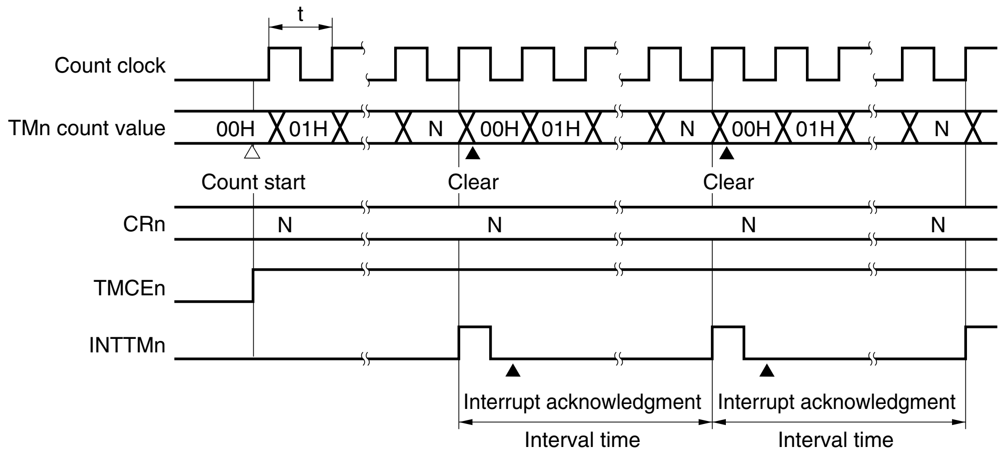
<4> Then, INTTMn is repeatedly generated at the same interval. To stop counting, set TMCEn = 0.

Interval time = (N + 1) × t: N = 00H to FFH

Caution During interval timer operation, do not rewrite the value of the CRn register.

Figure 8-2. Timing of Interval Timer Operation (1/2)

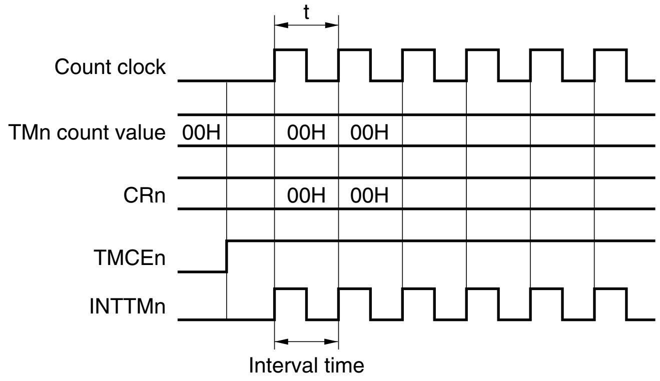
Basic operation



Remark n = 2 to 5

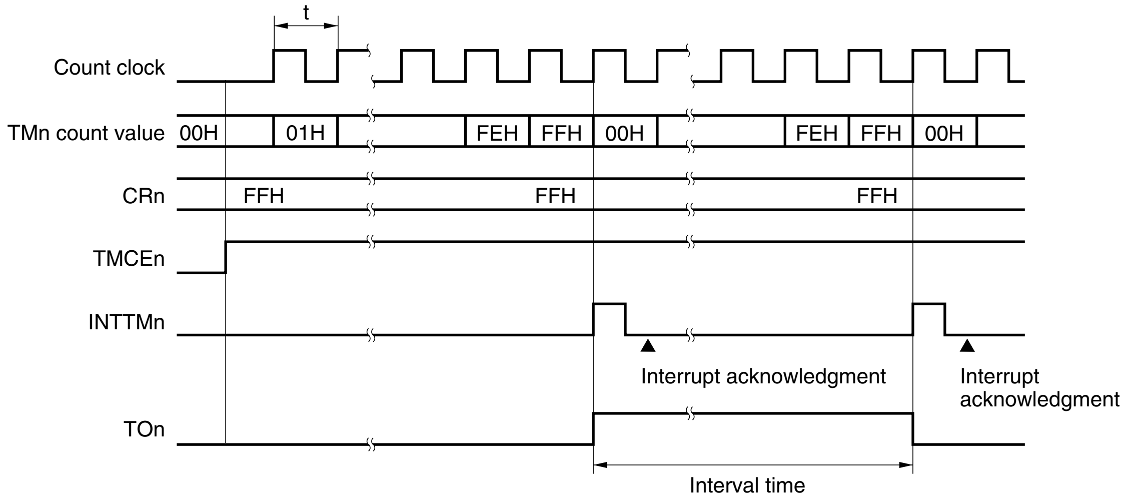
Figure 8-2. Timing of Interval Timer Operation (2/2)

When CRn register = 00H



Remark n = 2 to 5

When CRn register = FFH



Remark n = 2 to 5

8.4.2 Operation as external event counter (8 bits)

The external event counter counts the number of clock pulses input to the TIn pin from an external source by using 8-bit timer counter n (TMn).

Each time the valid edge specified by timer clock selection register n (TCLn) is input to the TIn pin, the TMn register is incremented. Either the rising edge or the falling edge can be specified as the valid edge.

When the count value of the TMn register matches the value of 8-bit timer compare register n (CRn), the TMn register is cleared to 0 and an interrupt request signal (INTTMn) is generated.

Setting method

<1> Set each register.

- TCLn register: Selects the TIn input edge.
  Falling edge of TIn pin → TLCn = 00H
  Rising edge of TIn pin → TCLn = 01H
- CRn register: Compare value (N)
- TMCn register: Stops count operation, selects the mode in which clear & start occurs on a match between the TMn register and CRn register, disables timer output F/F inversion operation, and disables timer output.
  (TMCn register = 0000xx00B, ×: don't care)

<2> When the TMCEn bit of the TMCn register is set to 1, the counter counts the number of pulses input from TIn.

<3> When the values of the TMn register and CRn register match, INTTMn is generated (TMn register is cleared to 00H).

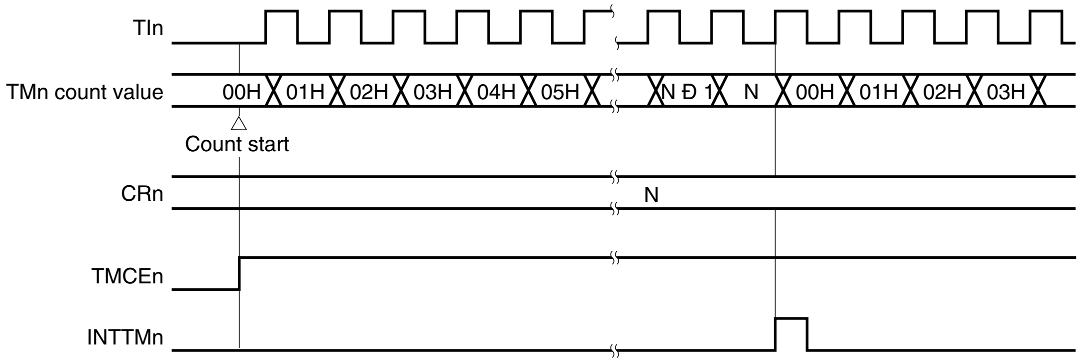
<4> Then, INTTMn is generated each time the values of the TMn register and CRn register match.

INTTMn is generated when the valid edge of TIn is input N + 1 times: N = 00 to FFH

Caution During external event counter operation, do not rewrite the value of the CRn register.

Remark n = 2 to 5

Figure 8-3. Timing of External Event Counter Operation (with Rising Edge Specified)



Remark n = 2 to 5

8.4.3 Square-wave output operation (8-bit resolution)

A square wave with any frequency can be output at an interval specified by the value preset in 8-bit timer compare register n (CRn).

By setting the TOEn bit of 8-bit timer mode control register n (TMCn) to 1, the output status of the TOn pin is inverted at an interval specified by the count value preset in the CRn register. In this way, a square wave of any frequency can be output (duty = 50%) (n = 2 to 5).

Setting method

<1> Set each register.

- TCLn register: Selects the count clock (t).
- CRn register: Compare value (N)
- TMCn register: Stops count operation, selects the mode in which clear & start occurs on a match between the TMn register and CRn register.

LVSn	LVRn	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Enables timer output F/F inversion operation, and enables timer output.
(TMCn register = 00001011B or 00000111B)

<2> When the TMCEn bit of the TMCn register is set to 1, counting starts.

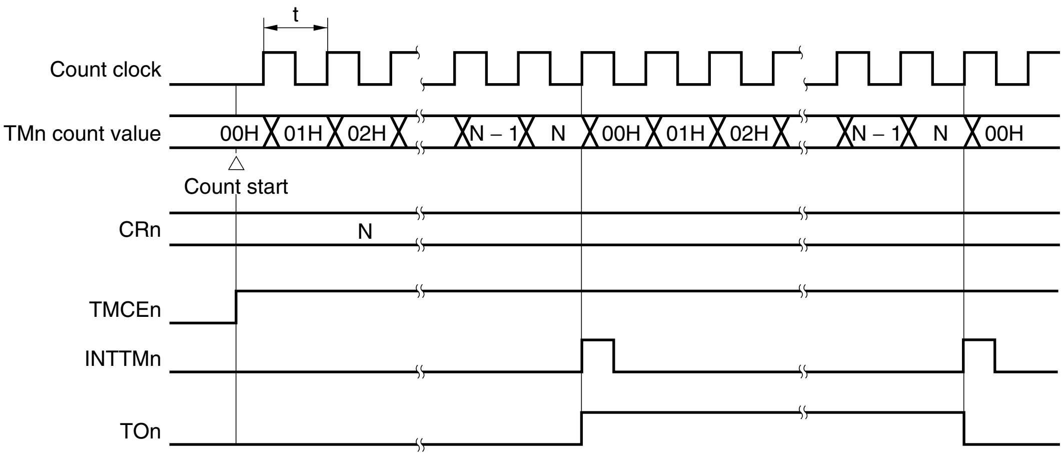
<3> When the values of the TMn register and CRn register match, the timer output F/F is inverted.
Moreover, INTTMn is generated and the TMn register is cleared to 00H.

<4> Then, the timer F/F is inverted during the same interval and a square wave is output from the TOn pin.

Frequency = 1/2t (N + 1): N = 00H to FFH

Caution Do not rewrite the value of the CRn register during square-wave output.

Figure 8-4. Timing of Square-Wave Output Operation



Note The initial value of the TOn output can be set using the LVSn and LVRn bits of the TMCn register.

Remark n = 0, 1

8.4.4 8-bit PWM output operation

By setting the TMCn6 bit of 8-bit timer mode control register n (TMCn) to 1, 8-bit timer/event counter n performs PWM output.

Pulses with the duty factor determined by the value set in 8-bit timer compare register n (CRn) are output from the TOn pin.

Set the width of the active level of the PWM pulse in the CRn register. The active level can be selected using the TMCn1 bit of the TMCn register.

The count clock can be selected using timer clock selection register n (TCLn).

PWM output can be enabled/disabled by the TOEn bit of the TMCn register.

Caution The CRn register rewrite interval must be three or more operation clocks (set by the TCLn register).

(1) Basic operation of PWM output

Setting method

<1> Set each register.

- TCLn register: Selects the count clock (t).
- CRn register: Compare value (N)
- TMCn register: Stops count operation, selects PWM mode, and leave timer output F/F unchanged.

TMCn1	Active Level Selection
0	Active high
1	Active low

Timer output enabled
(TMCn register = 01000001B or 01000011B)

<2> When the TMCEn bit of the TMCn register is set to 1, counting starts.

PWM output operation

<1> When counting starts, PWM output (output from the TOn pin) outputs the inactive level until an overflow occurs.

<2> When an overflow occurs, the active level set by setting method <1> is output. The active level is output until the value of the CRn register and the count value of 8-bit timer counter n (TMn) match.

<3> When the value of the CRn register and the count value match, the inactive level is output and continues to be output until an overflow occurs again.

<4> Then, steps <2> and <3> are repeated until counting is stopped.

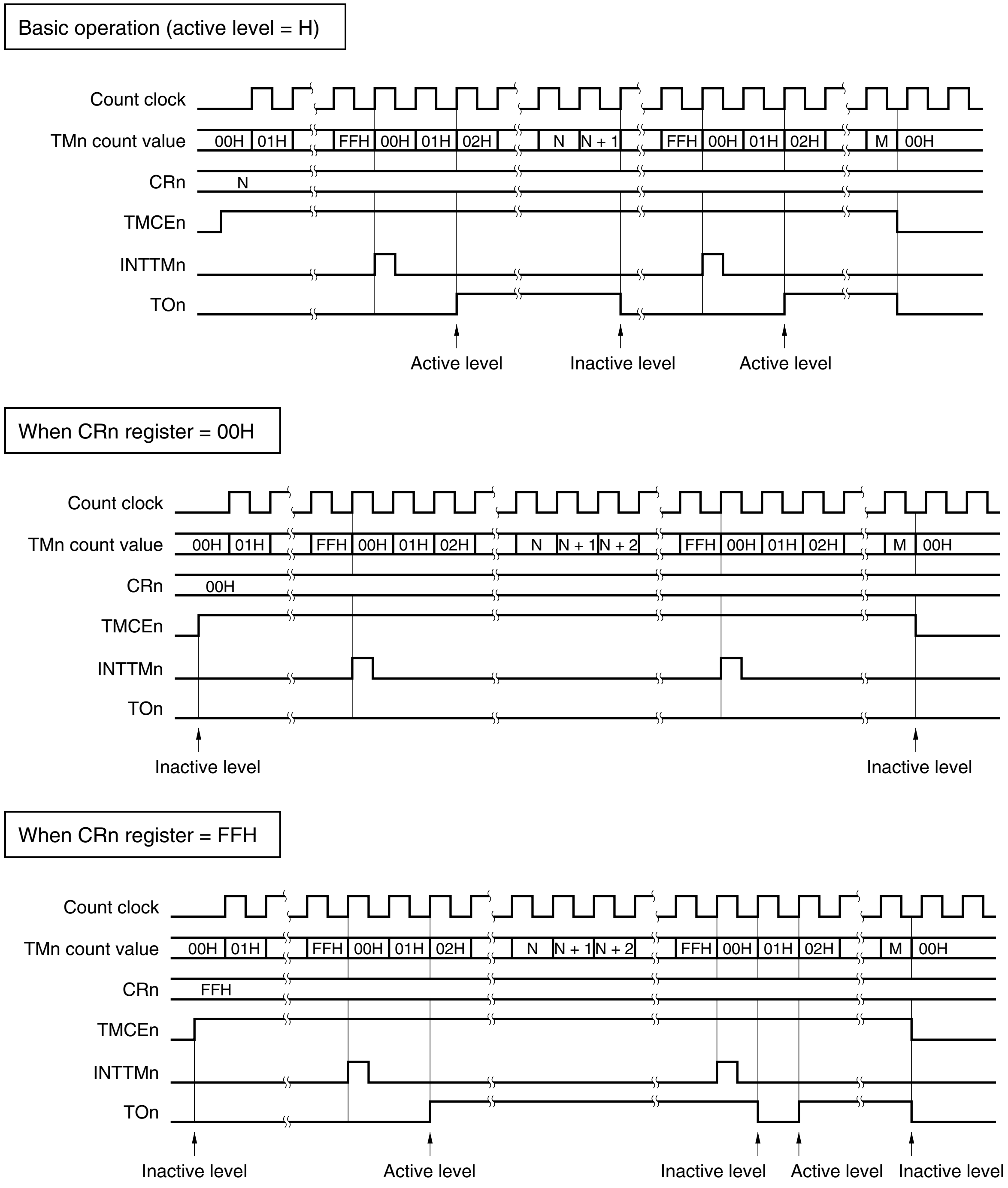
<5> When counting is stopped by setting TMCEn to 0, PWM output becomes inactive.

Cycle = 2^8t, active level width = Nt, duty = N/2^8: N = 00 to FFH

Remark n = 2 to 5

(a) Basic operation of PWM output

Figure 8-5. Timing of PWM Output Operation



Basic operation (active level = H)

When CRn register = 00H

When CRn register = FFH

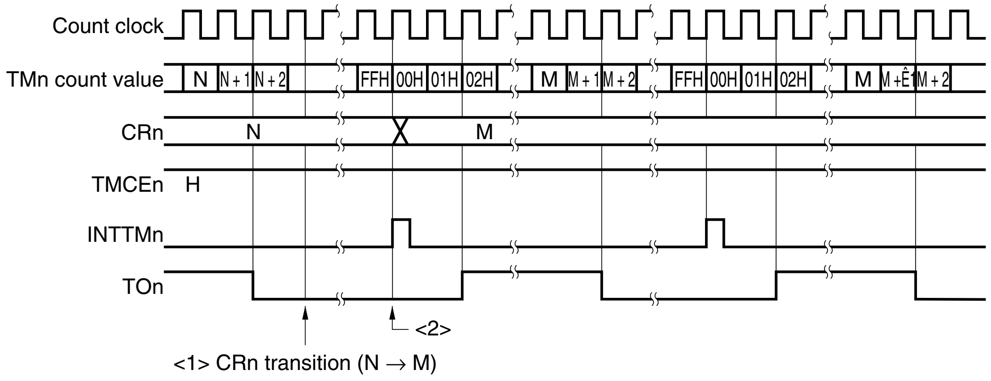
Remark n = 2 to 5

(b) Operation based on CRn register transitions

Figure 8-6. Timing of Operation Based on CRn Register Transitions

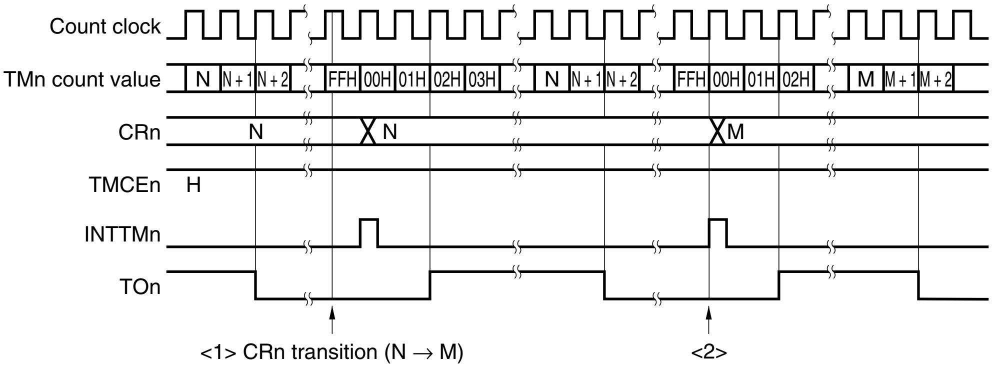
When the value of the CRn register changes from N to M before the rising edge of the FFH clock

→ The value of the CRn register is reloaded at the overflow that occurs immediately after.



When the value of the CRn register changes from N to M after the rising edge of the FFH clock

→ The value of the CRn register is reloaded at the second overflow.



Caution In the case of reload from the CRn register between <1> and <2>, the value that is actually used differs (Read value: M; Actual value of CRn register: N).

Remark n = 2 to 5

8.4.5 Operation as interval timer (16 bits)

The V850ES/SA2 and V850ES/SA3 are provided with a 16-bit register that can be used only during cascade connection.

The 16-bit resolution timer/event counter mode is selected by setting the TMC34 and TMC54 bits of 8-bit timer mode control registers 3 and 5 (TMC3 and TMC5) to 1.

8-bit timer/event counter n operates as an interval timer by repeatedly generating interrupts using the count value preset in 16-bit timer compare registers 23 and 45 (CR23 and CR45) as the interval.

In the following description, TM2 and TM3 are used. Read TM2 and TM3 as TM4 and TM5 when using TM4 and TM5.

Setting method (when TM2 and TM3 are connected in cascade)

<1> Set each register.

- TCL2 register: Selects the count clock (t)
  (The TCL3 register does not need to be set in cascade connection)
- CR2 register: Compare value (N) ... Lower 8 bits (settable from 00H to FFH)
- CR3 register: Compare value (N) ... Higher 8 bits (settable from 00H to FFH)
- TMC2, TMC3 register: Selects the mode in which clear & start occurs on a match between TM23 register and CR23 register (×: don’t care)
  TMC2 register = 0000xx00B
  TMC3 register = 0001xx00B

<2> Set the TMCE3 bit of the TMC3 register to 1. Then set the TMCE2 bit of the TMC2 register to 1 to start the count operation.

<3> When the values of the TM23 register and CR23 register connected in cascade match, INTTM2 is generated (the TM23 register is cleared to 0000H).

<4> INTTM2 is then generated repeatedly at the same interval.

Interval time = (N + 1) × t: N = 0000H to FFFFH

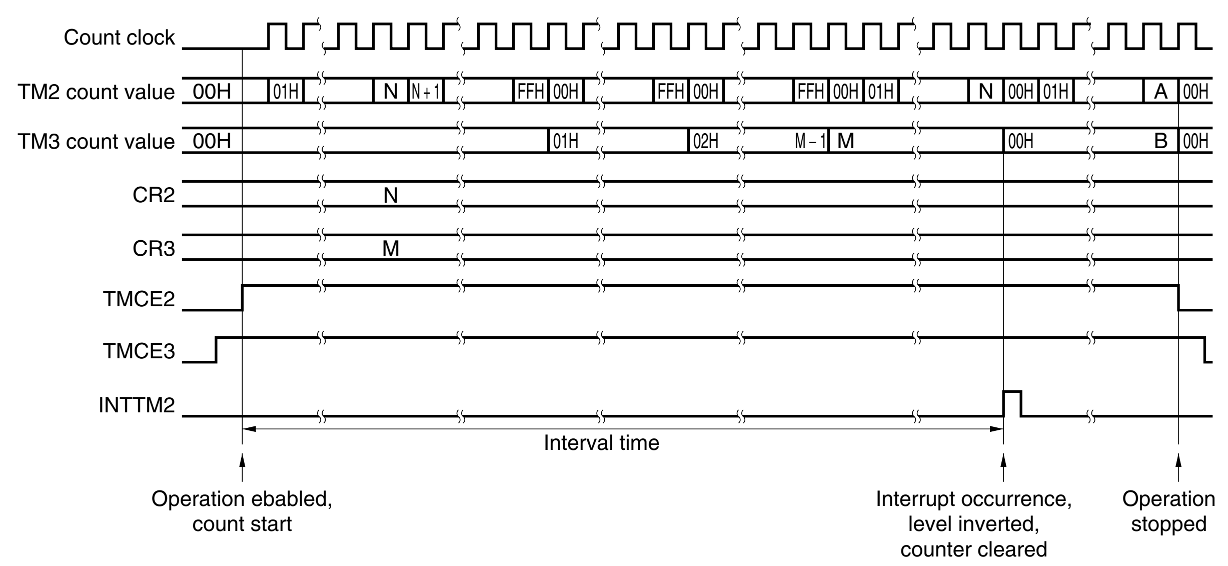
Cautions 1. To write using 8-bit access during cascade connection, set the TMCE3 bit to 1 at operation start and then set the TMCE2 bit to 1. When operation is stopped, set the TMCE2 bit to 0 and then set the TMCE3 bit to 0.

2. During cascade connection, TI2 input, TO2 output, and INTTM2 input are used while TI3 input, TO3 output, and INTTM3 input are not, so set bits LVS3, LVR3, TMC31, and TOE3 to 0.

3. Do not change the value of the CR23 register during timer operation.

Figure 8-7 shows a timing example of the cascade connection mode with 16-bit resolution.

Figure 8-7. Cascade Connection Mode with 16-Bit Resolution (When TM2 and TM3 Are Connected)



8.4.6 Operation as external event counter (16 bits)

The V850ES/SA2 and V850ES/SA3 are provided with a 16-bit register that can be used only during cascade connection.

The 16-bit resolution timer/event counter mode is selected by setting the TMC34 and TMC54 bits of 8-bit timer mode control registers 3 and 5 (TMC3 and TMC5) to 1.

The external event counter counts the number of clock pulses input to the TI2 and TI4 pins from an external source using 16-bit timer counters 23 and 45 (TM23 and TM45).

In the following description, TM2 and TM3 are used. Read TM2 and TM3 as TM4 and TM5 when using TM4 and TM5.

Setting method (when TM2 and TM3 are connected in cascade)

<1> Set each register.

- TCL2 register: Selects the TI2 input edge.
  (The TCL3 register does not have to be set during cascade connection.)
  Falling edge of TI2 → TCL2 = 00H
  Rising edge of TI2 → TCL2 = 01H
- CR2 register: Compare value (N) ... Lower 8 bits (settable from 00H to FFH)
- CR3 register: Compare value (N) ... Higher 8 bits (settable from 00H to FFH)
- TMC2, TMC3 registers: Stops count operation, selects the clear & stop mode entered on a match between the TM23 register and CR23 register, disables timer output F/F inversion, and disables timer output.
  (×: don't care)
  TMC2 register = 0000xx00B
  TMC3 register = 0001xx00B

<2> Set the TMCE3 bit of the TMC3 register to 1. Then set the TMCE2 bit of the TMC2 register to 1 and count the number of pulses input from TI2.

<3> When the values of the TM23 register and CR23 register connected in cascade match, INTTM2 is generated (the TM23 register is cleared to 0000H).

<4> INTTM2 is then generated each time the values of the TM23 register and CR23 register match.

INTTM2 is generated when the valid edge of TI2 is input N + 1 times: N = 0000 to FFFFH

Cautions 1. During external event counter operation, do not rewrite the value of the CRn register.

2. To write using 8-bit access during cascade connection, set the TMCE3 bit to 1 and then set the TMCE2 bit to 1. When operation is stopped, set the TMCE2 bit to 0 and then set the TMCE3 bit to 0.

3. During cascade connection, TI2 input and INTTM2 input are used while TI3 input, TO3 output, and INTTM3 input are not, so set bits LVS3, LVR3, TMC31, and TOE3 to 0.

4. Do not change the value of the CR23 register during external counter operation.

8.4.7 Square-wave output operation (16-bit resolution)

The V850ES/SA2 and V850ES/SA3 are provided with a 16-bit register that can be used only during cascade connection.

The 16-bit resolution timer/event counter mode is selected by setting the TMC34 and TCM54 bits of 8-bit timer mode control registers 3 and 5 (TMC3 and TMC5) to 1.

8-bit timer/event counter n outputs a square wave of any frequency using the interval preset in 16-bit timer compare registers 23 and 45 (CR23 and CR45).

In the following description, TM2 and TM3 are used. Read TM2 and TM3 as TM4 and TM5 when using TM4 and TM5.

Setting method (when TM2 and TM3 are connected in cascade)

<1> Set each register.

- TCL2 register: TCL2 selects the count clock (t)
  (The TCL3 register does not have to be set in cascade connection)
- CR2 register: Compare value (N) ... Lower 8 bits (settable from 00H to FFH)
- CR3 register: Compare value (N) ... Higher 8 bits (settable from 00H to FFH)
- TMC2, TCM3 registers: Stops count operation, selects the mode in which clear & start occurs on a match between the TM23 register and CR23 register.

LVS2	LVR2	Timer Output F/F Status Settings
1	0	High-level output
0	1	Low-level output

Enables timer output F/F inversion, and enables timer output.

TMC2 register = 00001011B or 00000111B
TMC3 register = 00010000B

<2> Set the TMCE3 bit of the TMC3 register to 1. Then set the TMCE2 bit of the TMC2 register to 1 to start the count operation.

<3> When the values of the TM23 register and the CR23 register connected in cascade match, the TO2 timer output F/F is inverted. Moreover, INTTM2 is generated and the TM23 register is cleared to 0000H.

<4> Then, the timer F/F is inverted during the same interval and a square wave is output from the TO2 pin.

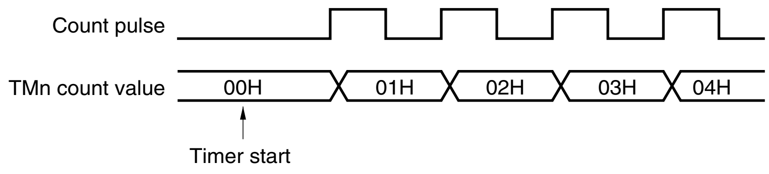
Frequency = 1/2t (N + 1): N = 0000H to FFFFH

8.4.8 Cautions

(1) Error on starting timer

An error of up to 1 clock occurs before the match signal is generated after the timer has been started. This is because 8-bit timer counter n (TMn) is started asynchronously to the count pulse.

Figure 8-8. Start Timing of Timer n



Remark n = 2 to 5

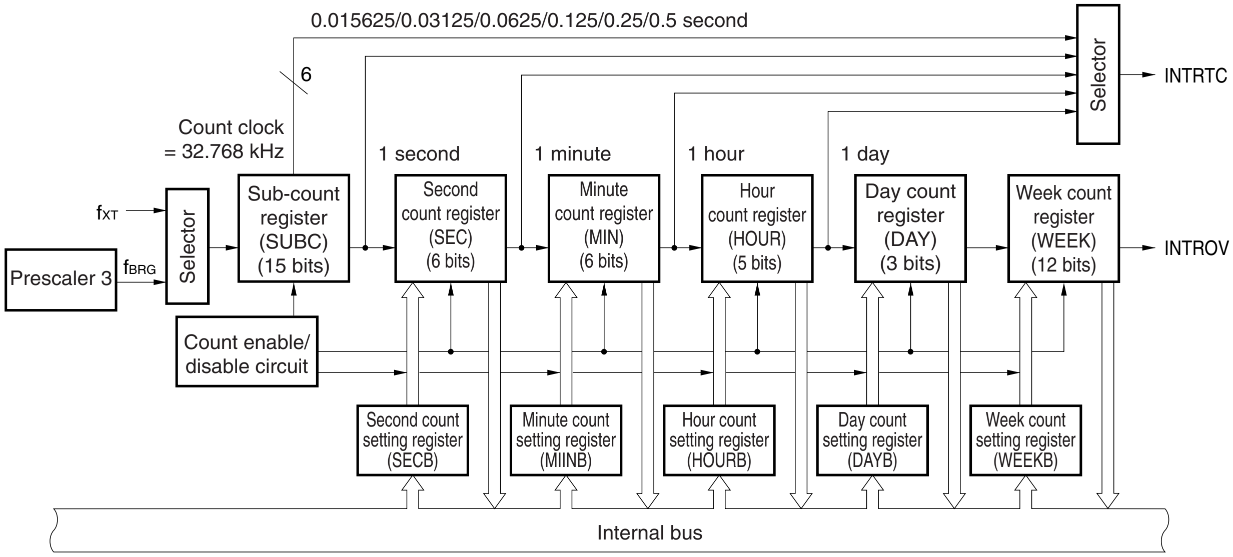
9.1 Function

The real-time counter has the following functions.

- Week, day, hour, minute, and second counters that can count up to 4,095 weeks
- Week, day, hour, minute, and second counters can be read while they are operating/stopped
- Generates overflow interrupt request signal (INTROV) from week counter.
- Generates interval interrupt request signal (INTRTC) at intervals of 0.015625, 0.03125, 0.0625, 0.125, 0.25, 0.5, or 1 second, 1 minute, 1 hour, or 1 day.

The configuration of the real-time counter function is shown below.

Figure 9-1. Block Diagram of Real-Time Counter


0.015625/0.03125/0.0625/0.125/0.25/0.5 second
6
Count clock
= 32.768 kHz
1 second
1 minute
1 hour
1 day
Selector
INTRTC
f_{XT}
f_{BRG}
Prescaler 3
Selector
Sub-count register (SUBC) (15 bits)
Second count register (SEC) (6 bits)
Minute count register (MIN) (6 bits)
Hour count register (HOUR) (5 bits)
Day count register (DAY) (3 bits)
Week count register (WEEK) (12 bits)
INTROV
Count enable/ disable circuit
Second count setting register (SECB)
Minute count setting register (MIINB)
Hour count setting register (HOURB)
Day count setting register (DAYB)
Week count setting register (WEEKB)
Internal bus

Remark f_{BRG}: Prescaler 3 clock frequency (Refer to **6.5 Prescaler 3**.)
f_{XT}: Subclock frequency

9.2 Control Registers

The registers listed in the table below control the real-time counter.

Table 9-1. Control Registers of Real-Time Counter

Register Name	Function	Instruction Unit	Reset Value	R/W
RTCC0	RTC control register 0	8/1-bit instruction	80H	R/W
RTCC1	RTC control register 1	8/1-bit instruction	8xH	R/W
SUBC	Sub-count register	8/16-bit instruction	Undefined	R
SEC	Second count register	8-bit instruction	Undefined	R
MIN	Minute count register	8-bit instruction	Undefined	R
HOUR	Hour count register	8-bit instruction	Undefined	R
DAY	Day count register	8-bit instruction	Undefined	R
WEEK	Week count register	8/16-bit instruction	Undefined	R
SECB	Second count setting register	8-bit instruction	00H	W
MINB	Minute count setting register	8-bit instruction	00H	W
HOURB	Hour count setting register	8-bit instruction	00H	W
DAYB	Day count setting register	8-bit instruction	00H	W
WEEKB	Week count setting register	8/16-bit instruction	0000H	W

(1) RTC control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that controls the operation of the real-time counter. This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 80H R/W Address: FFFFF6E0H

	<7>	6	5	4	3	2	1	0
RTCC0	RTCAE	CKS	0	0	0	0	0	0

RTCAE	Enables/disables RTC operation
0	Stops RTC clock operation and resets sub-count value.
1	Enables RTC clock operation.

CKS	Selects input clock
0	Selects subclock (f_{XT}) as input clock.
1	Selects main clock (f_X) divided by prescaler 3 (f_{BRG})[Note] as input clock.

Note Refer to **6.5 Prescaler 3**.

f_{BRG} is used alternately as the A/D converter clock. When using f_{BRG} as the clock of the real-time counter, set the FR1 and FR0 bits of the A/D converter mode register (ADM) to other than 11.

(2) RTC control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that controls the operation of the real-time counter. This register can be read or written in 8-bit or 1-bit units.

This register is set to 8xH after reset.

After reset: 8xH[Note 1] R/W Address: FFFFF6E1H

	<7>	6	5	4	3	2	1	<0>
RTCC1	RTCE	INTS3	INTS2	INTS1	INTS0	0	0	RTCF[Note 2]

RTCE	Enables/disables RTC count-up operation
0	Disables RTC count operation.
1	Enables RTC count operation.

INTS3	INTS2	INTS1	INTS0	Specifies interrupt request signal generation timing
0	0	0	0	Does not generate interrupt request signal.
0	0	0	1	Generates interrupt request signal every 0.015625 second.
0	0	1	0	Generates interrupt request signal every 0.03125 second.
0	0	1	1	Generates interrupt request signal every 0.0625 second.
0	1	0	0	Generates interrupt request signal every 0.125 second.
0	1	0	1	Generates interrupt request signal every 0.25 second.
0	1	1	0	Generates interrupt request signal every 0.5 second.
0	1	1	1	Generates interrupt request signal every 1 second.
1	0	0	0	Generates interrupt request signal every 1 minute.
1	0	0	1	Generates interrupt request signal every 1 hour.
1	0	1	0	Generates interrupt request signal every 1 day.
Other than above				Setting prohibited

RTCF	RTC operation flag
0	Count operation is stopped
1	Count-up operation is in progress.

Notes 1. 80H or 81H, depending on the value of the RTCF bit.

2. The RTCF bit is a read-only bit.

Remark The timing of generating an interrupt request set by the INTS3 to INTS0 bits is determined by the setting of the CKS bit of the RTCC0 register, as follows.

When CKS bit = 0 f_{XT} = 32.768 kHz
When CKS bit = 1 f_{BRG} = 32.768 kHz

(3) Sub-count register (SUBC)

The SUBC register is a 15-bit register that counts the reference time of the real-time counter. It counts 1 second using the 32.768 kHz clock. This register is read-only, in 16-bit or 8-bit units.

This register is not initialized after reset or when RTCE = 0.

After reset: Undefined R Address: FFFFF6E2H

	15	14 … 0
SUBC	0	SUBC14 to SUBC0

(4) Second count register (SEC)

This 8-bit register uses a value of 0 to 59 (decimal) to indicate the count value in seconds. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCE = 0.

After reset: Undefined R Address: FFFFF6E4H

	7	6	5	4	3	2	1	0
SEC	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0

(5) Second count setting register (SECB)

This is an 8-bit register for setting the second count. This register is read-only, in 8-bit units. Set a count value in a range of 0 to 59 (decimal) to this register. Do not set a count value of 60 (decimal) or greater.

This register is cleared to 00H after reset.

After reset: 00H W Address: FFFFF6EAH

	7	6	5	4	3	2	1	0
SECB	0	0	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0

(6) Minute count register (MIN)

This 8-bit uses a value of 0 to 59 (decimal) to indicate the count value in minutes. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCE = 0.

After reset: Undefined R Address: FFFFF6E5H

	7	6	5	4	3	2	1	0
MIN	0	0	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0

(7) Minute count setting register (MINB)

This is an 8-bit register for setting the minute count. This register is read-only, in 8-bit units. Set a count value in a range of 0 to 59 (decimal) to this register. Do not set a count value of 60 (decimal) or greater.

This register is cleared to 00H after reset.

After reset: 00H W Address: FFFFF6EBH

	7	6	5	4	3	2	1	0
MINB	0	0	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0

(8) Hour count register (HOUR)

This 8-bit register uses a value of 0 to 23 (decimal) to indicate the count value in hours. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCE = 0.

After reset: Undefined R Address: FFFFF6E6H

	7	6	5	4	3	2	1	0
HOUR	0	0	0	HOUR4	HOUR3	HOUR2	HOUR1	HOUR0

(9) Hour count setting register (HOURB)

This is an 8-bit register for setting the hour count. This register is read-only, in 8-bit units. Set a count value in a range of 0 to 23 (decimal) to this register. Do not set a count value of 24 (decimal) or greater.

This register is cleared to 00H after reset.

After reset: 00H W Address: FFFFF6ECH

	7	6	5	4	3	2	1	0
HOURB	0	0	0	HOUR4	HOUR3	HOUR2	HOUR1	HOUR0

(10) Day count register (DAY)

This 8-bit register used a value of 0 to 6 (decimal) to indicate the count value in days. This register is read-only, in 8-bit units.

This register is not initialized after reset or when RTCE = 0.

After reset: Undefined R Address: FFFFF6E7H

	7	6	5	4	3	2	1	0
DAY	0	0	0	0	0	DAY2	DAY1	DAY0

(11) Day count setting register (DAYB)

This is an 8-bit register for setting the day count. This register is read-only, in 8-bit units. Set a count value in a range of 0 to 6 (decimal) to this register. Do not set a count value of 7 (decimal) or greater.

This register is cleared to 00H after reset.

After reset: 00H W Address: FFFFF6EDH

	7	6	5	4	3	2	1	0
DAYB	0	0	0	0	0	DAY2	DAY1	DAY0

(12) Week count register (WEEK)

This 16-bit register uses a value of 0 to 4,095 (decimal) to indicate the count value in weeks. This register is read-only, in 8-bit or 16-bit units.

This register is not initialized after reset or when RTCE = 0.

After reset: Undefined R Address: FFFFF6E8H

	15 12	11 0
WEEK	0000	WEEK11 to WEEK0

(13) Week count setting register (WEEKB)

This is a 16-bit register for setting the week count. This register is read-only, in 8-bit or 16-bit units. Set a count value in a range of 0 to 4,095 (decimal) to this register.

This register is cleared to 0000H after reset.

After reset: 0000H W Address: FFFFF6EDH

	15 12	11 0
WEEKB	0000	WEEK11 to WEEK0

9.3 Operation

9.3.1 Initializing counter and count-up

<1> When the reset signal is asserted (0), the values of RTC control registers 0 and 1 (RTCC0 and RTCC1) are initialized. Real-time counter clock operation is enabled when RTCAE of the RTCC0 register is set to 1, and real-time counter count operation is enabled when RTCE of the RTCC1 register is set to 1.

<2> The sub-count register (SUBC) is reset if the real-time count clock operation is stopped when RTCAE is 0.

<3> The real-time counter clock operation is stopped when the CKS bit of the RTCC0 register is selected and RTCAE is set to 1.

<4> After 3 internal clocks, the values of all the count setting registers are reflected on the corresponding count registers at all once, and each count register starts counting up.

<5> Each time a count register overflows, the higher count register starts counting up.

<6> At the clock after the one at which the overflow conditions of all the count registers have been satisfied, all the count registers are cleared to “0”. The INTROV signal is asserted active for the duration of one cycle of the real-time count clock after the week count register (WEEK) overflows.

9.3.2 Rewriting counter

<1> When the reset signal is asserted (0), the values of the RTCC0 and RTCC1 registers are initialized. Real-time counter clock operation is enabled when RTCAE is set to 1, and real-time counter count operation is enabled when RTCE is set to 1.

<2> Write a value to each count setting register.

<3> The value of all the count setting registers are reflected on the corresponding count registers all at once two internal clocks after RTCE is set to 1, and the real-time counter starts counting up 3 internal clocks after that.

9.3.3 Controlling interrupt request signal output

This section explains how to control interrupt request signals, taking INTS0 to INTS3 = 0111B (every second) and INTS0 to INTS3 = 1000B (every minute) as an example.

<1> When the reset signal is asserted (0), the values of the RTCC0 and RTCC1 registers are initialized. Real-time counter clock operation is enabled when RTCAE is set to 1, and real-time counter count operation is enabled when RTCE is set to 1.

<2> Clear RTCAE to 0 and select the CKS bit.

<3> The internal clock operation is started when RTCAE = 1.

<4> After 3 internal clocks, the value of all the count setting registers are reflected on the corresponding count registers at all once, and the real-time counter starts counting up.

<5> Set the INTS0 to INTS3 bits to 0111B (1000B).

<6> Because INTS0 to INTS3 = 0111B, the INTRTC signal is asserted each time 1 second is counted (because INTS0 to INTS3 = 1000B, the INTRTC signal is asserted each time 1 minute is counted).

<7> The INTROV signal is asserted when the overflow conditions of all the count registers have been satisfied.

9.3.4 Notes

(1) When the reset signal is input, the CKS bit of RTC control register 0 (RTCC0) is cleared to 0. Therefore, the real-time counter operates with the subclock (f_{XT}). Note the following points.

- To continue the real-time counter operation even during the reset period, select f_{XT} (CKS = 0) as the count clock. If the prescaler 3 clock (f_{BRG}) (CKS = 1) is selected, the count clock is changed to f_{XT} (CKS = 0) by the reset input, in which case the operation cannot be guaranteed.
- If the real-time counter is not used, clear RTCAE of the RTCC0 register to 0 after the reset signal has been cleared.

(2) Perform initialization after clearing RTCAE to 0 when the reset signal has been cleared for the first time. For initialization, set each count setting register, count clock, and interrupt request signal generation timing using the procedure described in (4) and (5) below, and clear the OVFIF bit of the OVFIC register and the RTCIF bit of the RTCIC register to 0.

(3) Read each count register using the following procedure:

<1> Read the second, minute, hour, day, and week count registers in that order, and then read the second count register again.

<2> Compare the value of the second count register read first with the value of the second count register read second.

If the two values do not match, the chances are that the counter counted up while it was being read. If so, repeat steps <1> and <2> again.

(4) Write data to each count setting register using the following procedure:

- To clear the sub-count register (SUBC)

  <1> Using the procedure described in (3) above, read the values of all the count registers (this may be omitted), and clear RTCAE to 0.

  <2> Write a value to one of the count setting registers. Write the value read in step <1> to the other count setting registers.

  <3> Set RTCAE to 1. The values of the count setting registers will be transferred to the count registers, and the real-time counter will start counting (after 2 or 3 count clocks).

- To not clear the sub-count register (SUBC) (to hold the value)

  <1> Clear RTCE of RTC control register 1 (RTCC1) to 0, and check if RTCF of the RTCC1 register is cleared to 0 (count stops).

  <2> Read the values of all the count registers (this may be omitted).

  <3> Write a value to one of the count setting registers. Write the value read in <2> to the other count setting registers.

  <4> Set RTCE to 1. The values of the count setting registers will be transferred to the count registers, and the real-time counter will start counting (after 2 or 3 count clocks).

★ (5) Before changing the interrupt request signal generation timing or changing the RTCAE bit from 1 to 0, set the OVFMK bit of the OVFIC register and the RTCMK bit of the RTCIC register to 1, and after changing the generation timing or RTCAE bit, clear the OVFIF bit of the OVFIC register and the RTCIF bit of the RTCIC register to 0.

(6) To change the count clock, be sure to clear RTACE to 0.

★ (7) After releasing reset, clear the OVFIF bit of the OVFIC register and the RTCIF bit of the RTCIC register to 0.

CHAPTER 10 WATCHDOG TIMER FUNCTIONS

10.1 Functions

The watchdog timer has the following operation modes.

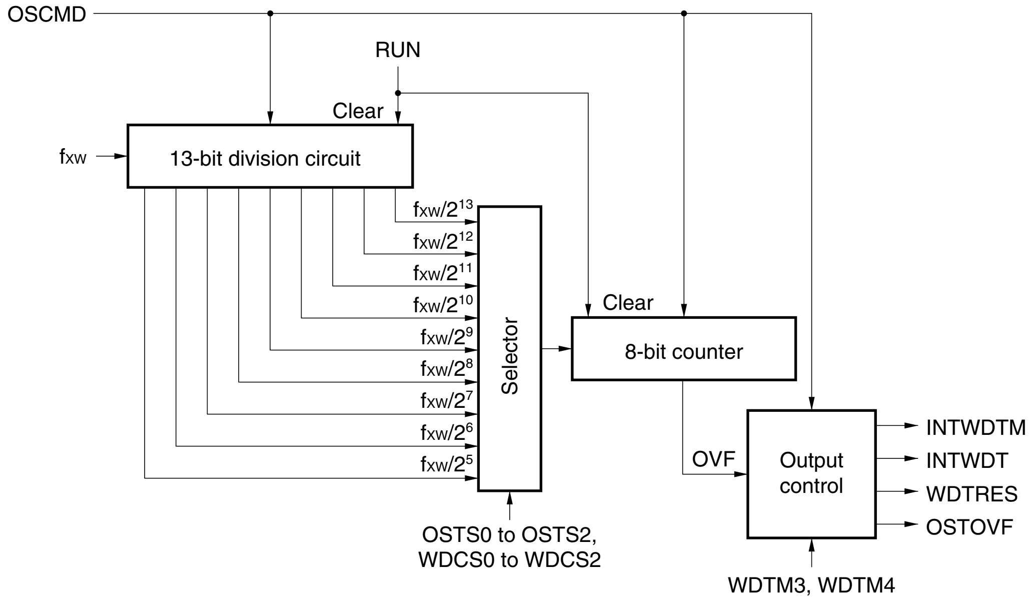
- Watchdog timer
- Interval timer
- Selecting the oscillation stabilization time

The following functions are realized from the above-listed operation modes.

- Generation of non-maskable interrupt request signal (INTWDT) upon overflow of watchdog timer
- Generation of system reset signal upon overflow of watchdog timer
- Generation of maskable interrupt request signal (INTWDTM) upon overflow of interval timer
- Securing of oscillation stabilization time for main system clock

Remark Select whether to use the watchdog timer in the watchdog timer mode or the interval timer mode using watchdog timer mode register (WDTM).

Figure 10-1. Block Diagram of Watchdog Timer



Remark INTWDTM: Request signal for maskable interrupt through WDT overflow

INTWDT: Request signal for non-maskable interrupt through WDT overflow

WDTRES: Reset signal through WDT overflow

OSTOVF: Oscillation stabilization timer overflow signal

OSCMD: Oscillation stabilization timer mode signal

fxw: Watchdog timer clock frequency

- While oscillation stabilization time is being counted: fxw = fx/2
- Other than above: fxw = fxx/2

10.2 Configuration

The watchdog timer consists of the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Oscillation stabilization time selection register (OSTS) Watchdog timer clock selection register (WDCS) Watchdog timer mode register (WDTM)

10.3 Control Registers

The registers that control the watchdog timer are as follows.

- Oscillation stabilization time selection register (OSTS)
- Watchdog timer clock selection register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Oscillation stabilization time selection register (OSTS)

This register selects the oscillation stabilization time following reset or release of the stop mode.

The OSTS register can be read or written in 8-bit units.

This register is set to 04H after reset.

After reset: 04H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time				
				f_X			
				20 MHz	17 MHz	13.5 MHz	8 MHz
0	0	0	$2^{14}/f_X$	Setting prohibited	Setting prohibited	Setting prohibited	2.048 ms
0	0	1	$2^{16}/f_X$	3.28 ms	3.855 ms	4.855 ms	8.192 ms
0	1	0	$2^{17}/f_X$	6.55 ms	7.710 ms	9.709 ms	16.38 ms
0	1	1	$2^{18}/f_X$	13.1 ms	15.42 ms	19.42 ms	32.77 ms
1	0	0	$2^{19}/f_X$	26.2 ms	30.84 ms	38.84 ms	65.54 ms
1	0	1	$2^{20}/f_X$	52.4 ms	61.68 ms	77.67 ms	131.1 ms
1	1	0	$2^{21}/f_X$	105 ms	123.4 ms	155.3 ms	262.1 ms
1	1	1	$2^{22}/f_X$	210 ms	246.7 ms	310.7 ms	524.3 ms

Caution Set the oscillation stabilization time to 1.5 ms or longer.

(2) Watchdog timer clock selection register (WDCS)

This register sets the overflow time of the watchdog timer and the interval timer.

The WDCS register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFF6C1H

	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer				
				f_X			
				20 MHz	17 MHz	13.5 MHz	8 MHz
0	0	0	$2^{14}/f_{XX}$	819 μs	964 μs	1.214 ms	2.048 ms
0	0	1	$2^{15}/f_{XX}$	1.638 ms	1.928 ms	2.427 ms	4.096 ms
0	1	0	$2^{16}/f_{XX}$	3.277 ms	3.855 ms	4.855 ms	8.192 ms
0	1	1	$2^{17}/f_{XX}$	6.554 ms	7.710 ms	9.709 ms	16.38 ms
1	0	0	$2^{18}/f_{XX}$	13.11 ms	15.42 ms	19.42 ms	32.77 ms
1	0	1	$2^{19}/f_{XX}$	26.21 ms	30.84 ms	38.84 ms	65.54 ms
1	1	0	$2^{20}/f_{XX}$	52.43 ms	61.68 ms	77.67 ms	131.1 ms
1	1	1	$2^{22}/f_{XX}$	209.7 ms	246.7 ms	310.7 ms	524.3 ms

Remark f_{XX}: Main clock frequency

(3) Watchdog timer mode register (WDTM)

This register sets the watchdog timer operation mode and enables/disables count operations.

This register is a special register that can be written only in a special sequence (refer to **3.4.8 Special registers**).

The WDTM register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF6C2H

	<7>	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Selection of watchdog timer operation mode[Note 1]
0	Stops counting
1	Clears counter and starts counting

WDTM4	WDTM3	Selection of watchdog timer operation mode[Note 2]
0	0	Interval timer mode (Upon overflow, maskable interrupt INTWDTM is generated.)
0	1	
1	0	Watchdog timer mode 1 (Upon overflow, non-maskable interrupt INTWDT is generated.)
1	1	Watchdog timer mode 2 (Upon overflow, reset operation WDTRES is started.)

Notes 1. Once the RUN bit is set (to 1), it cannot be cleared (to 0) by software. Therefore, when counting is started, it cannot be stopped except by reset ($\overline{\text{RESET}}$ input or watchdog timer overflow).

2. Once the WDTM3 and WDTM4 bits are set (to 1), they cannot be cleared (to 0) by software and can be cleared only by reset.

10.4 Operation

10.4.1 Operation as watchdog timer

Watchdog timer operation to detect a program loop is selected by setting bit 4 (WDTM4) of the watchdog timer mode register (WDTM) to 1.

The count clock (program loop detection time interval) of the watchdog timer can be selected with bits WDCS0 to WDCS2 of the watchdog timer clock selection register (WDCS). The count operation is started by setting bit 7 (RUN) of the WDTM register to 1. When, after the count operation is started, the RUN bit is again set to 1 within the set program loop detection time interval, the watchdog timer is cleared and the count operation starts again.

If the program loop detection time is exceeded without the RUN bit being set to 1, a reset (WDTRES) or a non-maskable interrupt request signal (INTWDT) is generated, depending on the value of bit WDTM3 of the WDTM register.

The count operation of the watchdog timer stops in the software STOP mode and IDLE mode. Therefore, set the RUN bit to 1 before the software STOP mode or IDLE mode is entered in order to clear the watchdog timer.

Because the watchdog timer operates in the HALT mode, do not use the watchdog timer when using the HALT mode.

Caution Once the WDTM4 bit is cleared to 0 (thereby selecting the interval timer mode), the watchdog timer mode is not entered as long as a reset does not occur.

When the subclock is selected for the CPU clock, the count operation of the watchdog timer stops (the value of the watchdog timer is maintained).

Table 10-2. Program Loop Detection Time of Watchdog Timer

Clock	Program Loop Detection Time			
	fxx = 20 MHz	fxx = 17 MHz	fxx = 13.5 MHz	fxx = 8 MHz
2^{14}/fxx	819 μs	964 μs	1.214 ms	2.048 ms
2^{15}/fxx	1.638 ms	1.928 ms	2.427 ms	4.096 ms
2^{16}/fxx	3.277 ms	3.855 ms	4.855 ms	8.192 ms
2^{17}/fxx	6.554 ms	7.710 ms	9.709 ms	16.38 ms
2^{18}/fxx	13.11 ms	15.42 ms	19.42 ms	32.77 ms
2^{19}/fxx	26.21 ms	30.84 ms	33.84 ms	65.54 ms
2^{20}/fxx	52.43 ms	61.68 ms	77.67 ms	131.1 ms
2^{22}/fxx	209.7 ms	246.7 ms	310.7 ms	524.3 ms

Remark fxx: Main clock frequency

10.4.2 Operation as interval timer

The watchdog timer can be made to operate as an interval timer that repeatedly generates interrupts using the count value set in advance as the interval, by setting bit 4 (WDTM4) of the watchdog timer mode register (WDTM) to 0.

When the watchdog timer operates as an interval timer, the interrupt mask flag (WDTMK) and priority specification flags (WDTPR0 to WDTPR2) of the WDTIC register are valid and maskable interrupt request signals (INTWDTM) can be generated. The default priority of the INTWDTM signal is set to the highest level among the maskable interrupt request signals.

The interval timer continues to operate in the HALT mode, but it stops operating in the software STOP mode and the IDLE mode. Therefore, set the RUN bit of the WDTM register to 1 before the software STOP mode or IDLE mode is entered in order to clear the interval timer.

Cautions 1. Once the WDTM4 bit is set to 1 (thereby selecting the watchdog timer mode), the interval timer mode is not entered as long as a reset does not occur.

2. When the subclock is selected for the CPU clock, the count operation of the watchdog timer stops (the value of the watchdog timer is maintained).

Table 10-3. Interval Time of Interval Timer

Clock	Interval Time			
	fxx = 20 MHz	fxx = 17 MHz	fxx = 13.5 MHz	fxx = 8 MHz
2^{14}/fxx	819 μs	964 μs	1.214 ms	2.048 ms
2^{15}/fxx	1.638 ms	1.928 ms	2.427 ms	4.096 ms
2^{16}/fxx	3.277 ms	3.855 ms	4.855 ms	8.192 ms
2^{17}/fxx	6.554 ms	7.710 ms	9.709 ms	16.38 ms
2^{18}/fxx	13.11 ms	15.42 ms	19.42 ms	32.77 ms
2^{19}/fxx	26.21 ms	30.84 ms	33.84 ms	65.54 ms
2^{20}/fxx	52.43 ms	61.68 ms	77.67 ms	131.1 ms
2^{22}/fxx	209.7 ms	246.7 ms	310.7 ms	524.3 ms

Remark fxx: Main clock frequency

10.4.3 Oscillation stabilization time selection function

The wait time until the oscillation stabilizes after the STOP mode is released is controlled by the oscillation stabilization time register (OSTS).

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

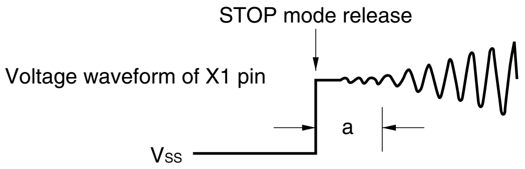
This register is set to 04H after reset.

After reset: 04H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time				
				f_X			
				20 MHz	17 MHz	13.5 MHz	8 MHz
0	0	0	$2^{14}/f_X$	Setting prohibited	Setting prohibited	Setting prohibited	2.048 ms
0	0	1	$2^{16}/f_X$	3.28 ms	3.855 ms	4.855 ms	8.192 ms
0	1	0	$2^{17}/f_X$	6.55 ms	7.710 ms	9.709 ms	16.38 ms
0	1	1	$2^{18}/f_X$	13.1 ms	15.42 ms	19.42 ms	32.77 ms
1	0	0	$2^{19}/f_X$	26.2 ms	30.84 ms	38.84 ms	65.54 ms
1	0	1	$2^{20}/f_X$	52.4 ms	61.68 ms	77.67 ms	131.1 ms
1	1	0	$2^{21}/f_X$	105 ms	123.4 ms	155.3 ms	262.1 ms
1	1	1	$2^{22}/f_X$	210 ms	246.7 ms	310.7 ms	524.3 ms

Cautions 1. The wait time following release of the software STOP mode does not include the time until the clock oscillation starts (“a” in the figure below) following release of the software STOP mode, regardless of whether the STOP mode is released by reset or the occurrence of an interrupt request signal.



2. Be sure to set bits 3 to 7 to 0.

3. Set the oscillation stabilization time to 1.5 ms or longer.

4. The oscillation stabilization time following reset release is $2^{19}/f_X$ (because the initial value of the OSTS register = 04H).

Remark f_X = Main clock oscillation frequency

CHAPTER 11 A/D CONVERTER

11.1 Function

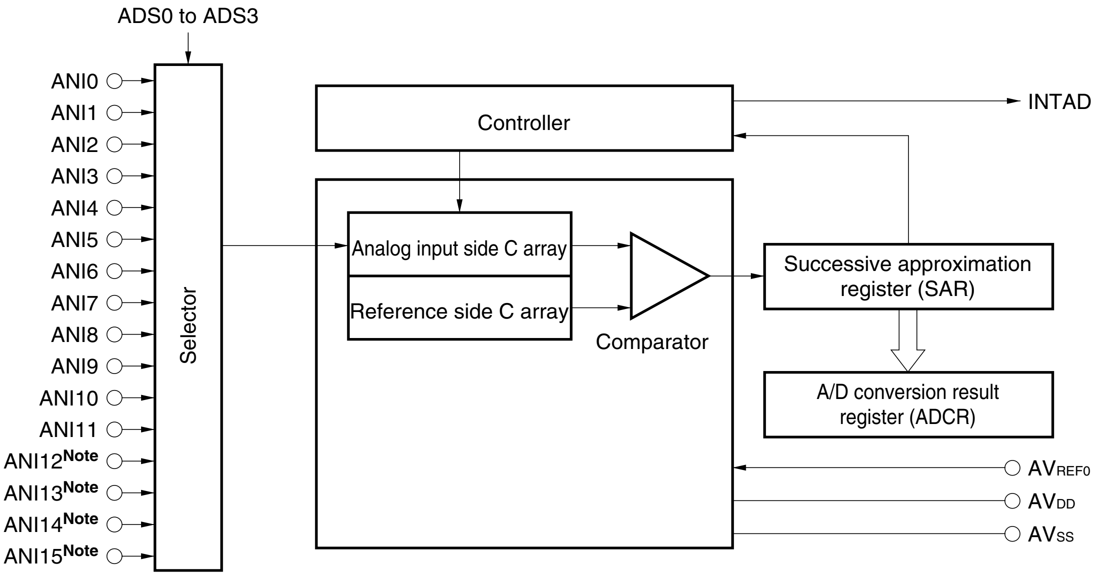
The A/D converter converts analog input signals into digital values with a resolution of 10 bits. In the V850ES/SA2, it has a 12-channel (ANI0 to ANI11) configuration, and in the V850ES/SA3, it has a 16-channel (ANI0 to ANI15) configuration.

The features of the A/D converter are shown below.

○ 10-bit resolution
○ 12 channels (V850ES/SA2)
16 channels (V850ES/SA3)
○ Successive approximation method
○ Power fail detection function
○ Operating voltage: $AV_{DD} = AV_{REF0}$ = 2.2 to 2.7 V
○ Analog input voltage: AV_{SS} to AV_{REF0}
○ Conversion rate: 8.2 to 150 μs

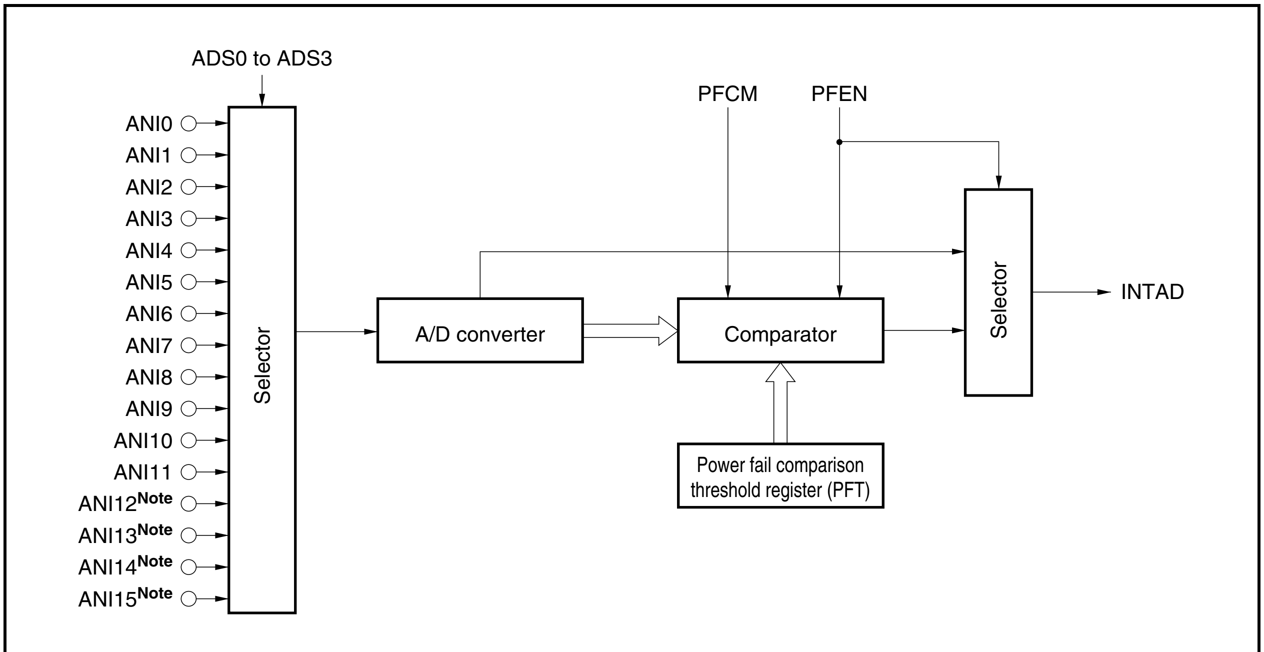
The block diagram is shown below.

Figure 11-1. Block Diagram of A/D Converter



Note V850ES/SA3 only

Figure 11-2. Block Diagram of Power Fail Detection Function



Note V850ES/SA3 only

11.2 Configuration

The A/D converter consists of the following hardware.

Table 11-1. Configuration of A/D Converter

Item	Configuration
Analog input	12 channels (ANI0 to ANI11): V850ES/SA2 16 channels (ANI0 to ANI15): V850ES/SA3
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR) A/D conversion result register H (ADCRH): Only higher 8 bits can be read Power fail comparison threshold register (PFT)
Control registers	A/D converter mode register (ADM) Analog input channel specification register (ADS) Power fail comparison mode register (PFM)

(1) Successive approximation register (SAR)

This register compares the voltage value of the analog input signal with the voltage tap (compare voltage) value from the series resistor string, and holds the comparison result starting from the most significant bit (MSB).

When the comparison result has been saved down to the least significant bit (LSB) (A/D conversion completion), the contents of the SAR are transferred to the A/D conversion result register.

(2) A/D conversion result register (ADCR), A/D conversion result register (ADCRH)

Each time A/D conversion has been completed, the result of the conversion is loaded to this register from the successive approximation register, and the higher 10 bits of this register hold the result of the A/D conversion (the lower 6 bits are fixed to 0).

The ADCR register is read-only, in 16-bit units. ADCR is undefined after reset.

When using only the higher 8 bits of the A/D conversion result, the ADCRH register is read-only, in 8-bit units.

Caution When data is written to the A/D converter mode register (ADM) or analog input channel specification register (ADS), the contents of the ADCR register may become undefined. Read the conversion result after completion of conversion and before writing data to the ADM and ADS registers. Otherwise, the correct conversion result may not be read.

(3) Power fail comparison threshold register (PFT)

This register sets the threshold when comparing with the A/D conversion result register.

The 8-bit data set in the PFT register and the higher 8 bits (ADCRH) of the A/D conversion result register are compared.

The PFT register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

(4) Sample & hold circuit

The sample & hold circuit samples each of the analog input signals sequentially sent from the input circuit and sends the sampled data to the voltage comparator. This circuit holds the sampled analog input voltage during A/D conversion.

(5) Voltage comparator

The voltage comparator compares the analog input signal with the output voltage of the series resistor string.

(6) Series resistor string

The series resistor string is connected between AV_{REF0} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(7) ANI0 to ANI15 pins[Note]

These are analog input pins for the 16 channels[Note] of the A/D converter that are used to input analog signals to be converted into digital signals. Pins other than those selected as analog input with the analog input channel specification register (ADS) can be used as input ports.

Note The V850ES/SA2 provides only 12 channels, ANI0 to ANI11.

Caution Make sure that the voltage input to ANI0 to ANI15 does not exceed the rated values. If a voltage higher than AV_{REF0} or lower than AV_{SS} (even within the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined and the conversion values of the other channels may also be affected.

(8) AV_{REF0} pin

This is the reference voltage input pin of the A/D converter. The signals input to the ANI0 to ANI15 or ANI0 to ANI11 pins are converted into digital signals based on the voltage applied across AV_{REF0} and AV_{SS}.

(9) AV_{DD} pin

This is the analog power supply pin of the A/D converter. Always use the same potential as the V_{DD} pin even when not using the A/D converter.

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use the same potential as the V_{SS} pin even when not using the A/D converter.

11.3 Control Registers

The A/D converter is controlled by the following registers.

- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- Power fail comparison mode register (PFM)

(1) A/D converter mode register (ADM)

This register sets the conversion time of the analog input signal to be converted into a digital signal as well as conversion start and stop.

The ADM register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF200H

	<7>	6	5	4	3	2	1	0
ADM	ADCSB[Note 1]	0	FR3[Note 2]	FR2[Note 2]	FR1	FR0	0	0

ADCS[Note 1]	A/D conversion control
0	Stops conversion
1	Enables conversion

FR3[Note 2]	FR2[Note 2]	Number of A/D conversion clocks
0	0	19 clocks
0	1	Setting prohibited
1	0	Setting prohibited
1	1	Setting prohibited

FR1	FR0	A/D conversion clock
0	0	$f_{XX}/16$
0	1	$f_{XX}/8$
1	0	$f_{XX}/4$
1	1	Clock of prescaler 3 (f_{BRG})

Notes 1. f_{BRG} output (refer to **6.5 Prescaler 3**) is alternated with the main clock divider of the real-time counter. To use f_{BRG} output as the conversion clock of the A/D converter, therefore, clear the CKS bit of RTC control register 0 (RTCC0) (refer to **9.2 (1)**) to 0.

An A/D conversion operation can be used to output the f_{BRG} clock in the IDLE mode. To reduce the power consumption, clear the CE bit of the prescaler mode register (PRSM) (refer to **6.5.1 (1)**) and ADCS bit of the ADM register to 0.

2. Be sure to clear the FR3 and FR2 bits to 00.

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

Remark Refer to **Tables 11-2** and **11-3** for examples of setting the A/D conversion time.

★ **Table 11-2. Example of Setting A/D Conversion Time (Immediately After Setting ADCS Bit to 1)**

FR3	FR2	FR1	FR0	A/D Conversion Time	f_{XX} = 20 MHz	f_{XX} = 17 MHz	f_{XX} = 13.5 MHz	f_{XX} = 8 MHz	f_{XX} = 2 MHz
0	0	0	0	392/f_{XX}	19.6 μs	23.1 μs	29.1 μs	49 μs	Setting prohibited
0	0	0	1	196/f_{XX}	9.8 μs	11.6 μs	14.6 μs	24.5 μs	98 μs
0	0	1	0	98/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	12.3 μs	49 μs
0	0	1	1	147/f_{XX}[Note]	Setting prohibited	8.7 μs	10.9 μs	18.4 μs	73.5 μs
Other than above				Setting prohibited					

Note When PRSM = 10H, PRSCM = 03H (Refer to **6.5 Prescaler 3**.)

Cautions 1. Set the A/D conversion time within the range of 8.2 to 150 μs. The operation is not guaranteed if this range is exceeded.

2. Clear the FR3 and FR2 bits to 00.

★ **Table 11-3. Example of Setting A/D Conversion Time (Second and Subsequent Time)**

FR3	FR2	FR1	FR0	A/D Conversion Time	f_{XX} = 20 MHz	f_{XX} = 17 MHz	f_{XX} = 13.5 MHz	f_{XX} = 8 MHz	f_{XX} = 2 MHz
0	0	0	0	320/f_{XX}	16.0 μs	18.9 μs	23.8 μs	40 μs	Setting prohibited
0	0	0	1	160/f_{XX}	8.0 μs	9.5 μs	11.9 μs	20 μs	80 μs
0	0	1	0	80/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	10 μs	40 μs
0	0	1	1	120/f_{XX}[Note]	Setting prohibited	7.1 μs	8.9 μs	15 μs	60 μs
Other than above				Setting prohibited					

Note When PRSM = 10H, PRSCM = 03H (Refer to **6.5 Prescaler 3**.)

(2) Analog input channel specification register (ADS)

This register specifies the analog voltage input port to be A/D converted.

The ADS register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF201H

	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Specification of analog input channel
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
1	0	1	0	ANI10
1	0	1	1	ANI11
1	1	0	0	ANI12[Note]
1	1	0	1	ANI13[Note]
1	1	1	0	ANI14[Note]
1	1	1	1	ANI15[Note]
Other than above				Setting prohibited

Note The ANI12 to ANI15 channels are available only in the V850ES/SA3. In the V850ES/SA2, setting these channels is prohibited.

(3) Power fail comparison mode register (PFM)

This register sets the power fail monitoring mode.

It compares the value of the power fail comparison threshold register (PFT) and the value of the A/D conversion result register (ADCRH).

The PFM register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset

After reset: 00H R/W Address: FFFFF202H

	7	6	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	Selection of power fail comparison enable/disable
0	Disables power fail comparison
1	Enables power fail comparison

PFCM	Selection of power fail comparison mode
0	Generates interrupt request signal (INTAD) when ADCR ≥ PFT
1	Generates interrupt request signal (INTAD) when ADCR < PFT

(4) Power fail comparison threshold value register (PFT)

This register sets the value to be compared in the power fail comparison mode.

The 8-bit data set in the PFT register and the higher 8 bits (ADCRH) of the A/D conversion result register are compared.

The PFT register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF203H

	7	6	5	4	3	2	1	0
PFT								

(5) A/D conversion result register, A/D conversion result register H (ADCR, ADCRH)

These registers store the A/D conversion results.

These registers are read-only, in 16-bit or 8-bit units. Specify the ADCR register when accessing in 16-bit units and specify the ADCRH register when accessing in 8-bit units. In the ADCR register, 10 bits of the conversion result are read from the higher 10 bits and 0 is read from the lower 6 bits. In the ADCRH register, the higher 8 bits of the conversion result are read.

The values of these registers are undefined after reset.

After reset: Undefined R Address: FFFFF204H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

After reset: Undefined R Address: FFFFF205H

	7	6	5	4	3	2	1	0
ADCRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

The following relationship exists between the analog input voltage input to the analog input pins (ANI0 to ANI11 or ANI0 to ANI15) and the A/D conversion results (A/D conversion result register (ADCR)).

$$\mathrm{SAR} = \mathrm{INT}\left(\frac{V_{IN}}{AV_{REF0}} \times 1024 + 0.5\right)$$

$$\mathrm{ADCR}^{\mathrm{Note}} = \mathrm{SAR} \times 64$$

Or

$$(\mathrm{SAR} - 0.5) \times \frac{AV_{REF0}}{1024} \leq V_{IN} < (\mathrm{SAR} + 0.5) \times \frac{AV_{REF0}}{1024}$$

INT ( ): Function that returns the integer portion of the value in parentheses

V_{IN}: Analog input voltage

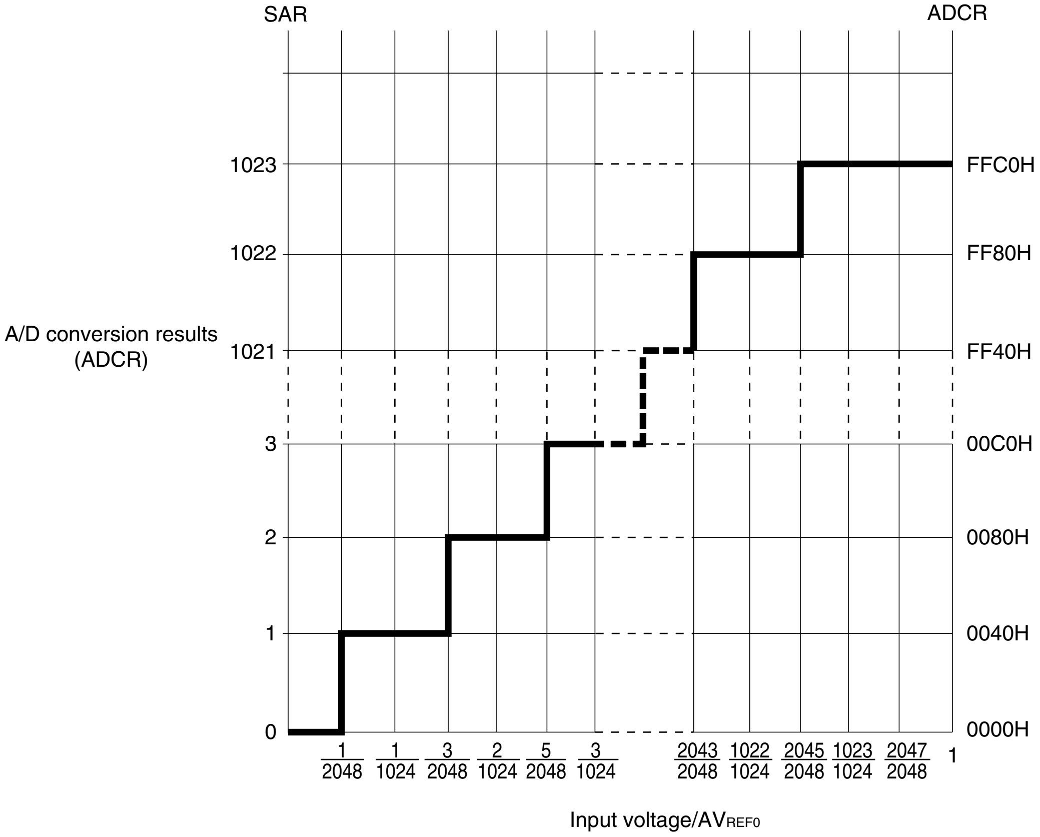
AV_{REF0}: AV_{REF0} pin voltage

ADCR: Value of A/D conversion result register n (ADCR)

Caution The lower 6 bits of ADCR are fixed to 0.

The following shows the relationship between the analog input voltage and A/D conversion results.

Figure 11-3. Relationship Between Analog Input Voltage and A/D Conversion Results



11.4 Operation

11.4.1 Conversion operation

- Setting ADCS of the A/D converter mode register (ADM) to 1 starts conversion of the signal input to the channel specified by the analog input channel specification register (ADS). Upon completion of the conversion, the conversion result is stored in the A/D conversion result register (ADCR) and a new conversion starts.
- If ADM, ADS, the power fail comparison threshold register (PFT), or the power fail comparison mode register (PFM) is written during conversion, conversion is interrupted and the conversion operation starts again from the beginning.
- If ADCS is set to 0 during conversion, conversion is interrupted and the conversion operation is stopped.
- For whether or not the conversion end interrupt request signal (INTAD) is generated, refer to **11.4.2**.

11.4.2 Conversion operation (power fail monitoring function)

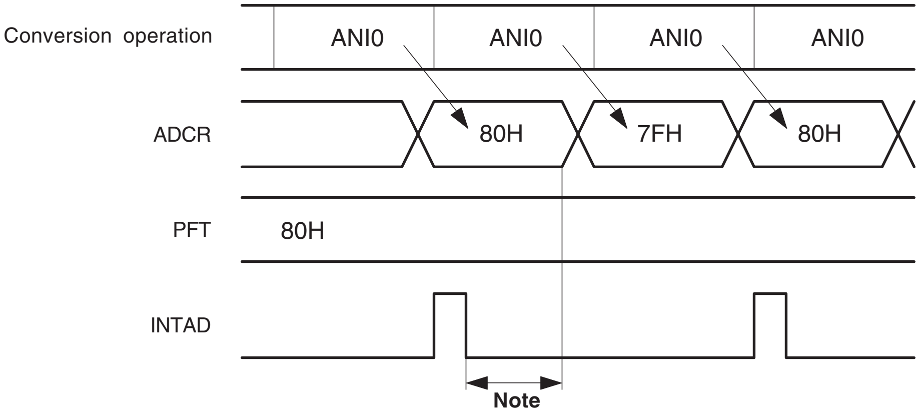
The conversion end interrupt request signal (INTAD) can be controlled as follows using the PFM and PFT registers.

Table 11-4. INTAD Signal Control

PFM Register		Operation
PFEN Bit	PFCM Bit	
0	Don’t care	INTAD signal is output each time A/D conversion ends
1	0	INTAD signal is output only if conversion result (ADCRH) ≥ PFT
1	1	INTAD signal is output only if conversion result (ADCRH) < PFT

Remark When PFEN = 1, because the conversion result is overwritten after INTAD has been output unless the conversion result is read by the time the next conversion ends, in some cases it may appear as if the actual operation differs from the operation described above (refer to **Figure 11-4**).

Figure 11-4. Power Fail Monitoring Function (PFCM = 0)



Note If reading is not performed during this interval, the conversion result changes to the next conversion result.

★ ## 11.5 Notes on Use

- Do not read the P7 register during A/D conversion.
- Do not change the set value of the ADM register during A/D conversion (ADCS bit = 1). Before changing the value, clear the ADCS bit to 0. The operation is not guaranteed when the value is changed with the ADCS bit set to 1.
- When not using the A/D converter in the standby mode, clear the ADCS bit to 0, then change to the standby mode. Connecting the AV_{REF0} pin to V_{SS} further reduces the power consumption. If the supply clock to the A/D converter stops with the ADCS bit set to 1 (i.e., in IDLE (the FR1 and FR0 bits of the ADM register are not 11B) or software STOP mode), each register stops operation holding the value immediately before the clock was stopped. The first A/D conversion operation after the clock supply is resumed cannot be guaranteed.

★ 11.6 How to Read A/D Converter Characteristics Table

This section describes the terms related to the A/D converter.

(1) Resolution

The minimum analog input voltage that can be recognized, i.e., the ratio of the analog input voltage to 1 bit of digital output, is called 1 LSB (least significant bit). The ratio of 1 LSB to the full scale is expressed as %FSR (full-scale range). %FSR is the ratio of the range of convertible analog input voltages expressed as a percentage, and can be expressed as follows, independently of the resolution.

1%FSR = (Maximum value of convertible analog input voltage – Minimum value of convertible analog input voltage)/100
= (AV_{REF} – 0)/100
= AV_{REF}/100

Where the resolution is 10 bits, 1 LSB is as follows:

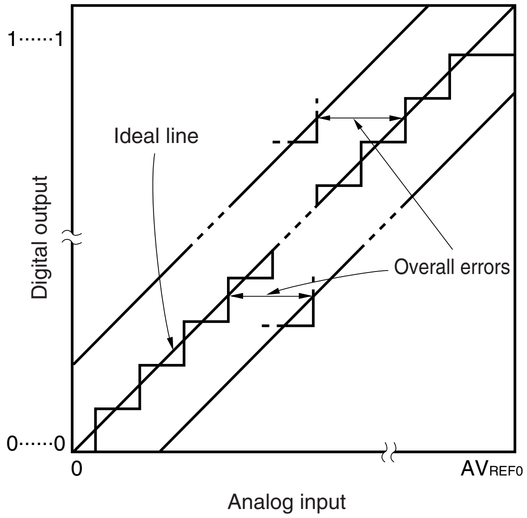
1 LSB = $1/2^{10}$ = 1/1,024
= 0.098%FSR

The accuracy is determined by the overall error, independently of the resolution.

(2) Overall error

This is the maximum value of the difference between the actually measured value and the theoretical value.
It is the total of the zero-scale error, full-scale error, linearity error, and combinations of these errors.
The overall error in the characteristics table does not include the quantization error.

Figure 11-5. Overall Error

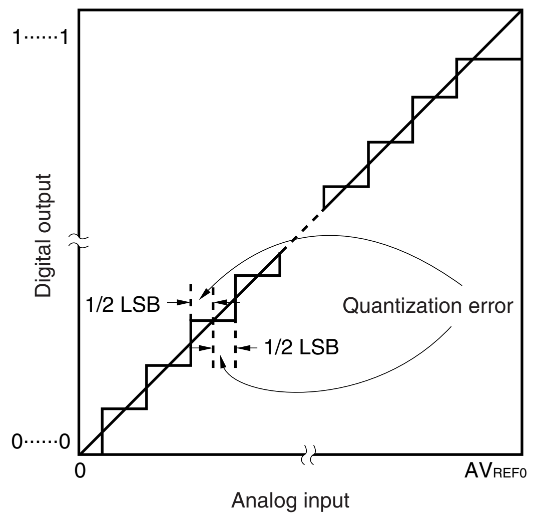


(3) Quantization error

This is the error of ±1/2 LSB that inevitably occurs when an analog value is converted into a digital value. Because the A/D converter converts analog input voltages in a range of ±1/2 LSB into the same digital codes, the quantization error is unavoidable.

This 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.

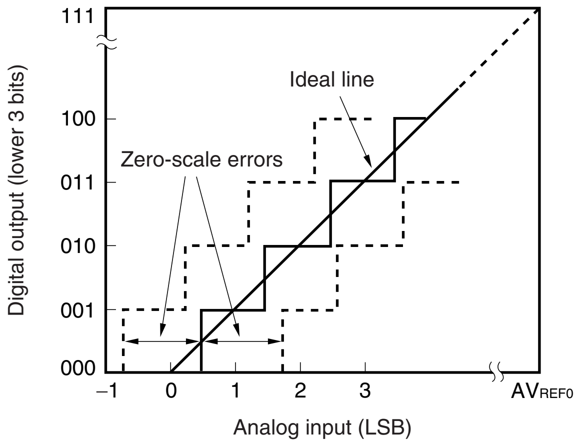
Figure 11-6. Quantization Error



(4) Zero-scale error

This is the difference between the actually measured analog input voltage and its theoretical value when the digital output changes from 0…000 to 0…001 (1/2 LSB).

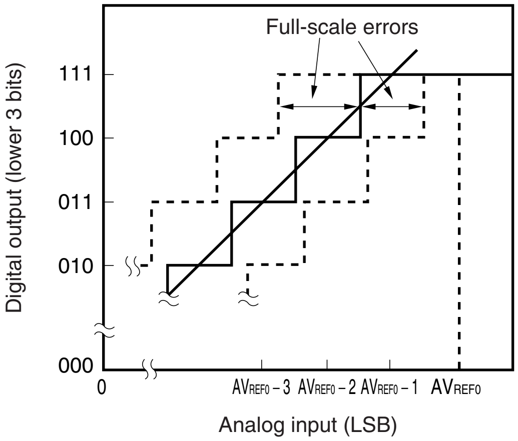
Figure 11-7. Zero-Scale Error



(5) Full-scale error

This is the difference between the actually measured analog input voltage and its theoretical value when the digital output changes from 1…110 to 0…111 (full scale – 3/2 LSB).

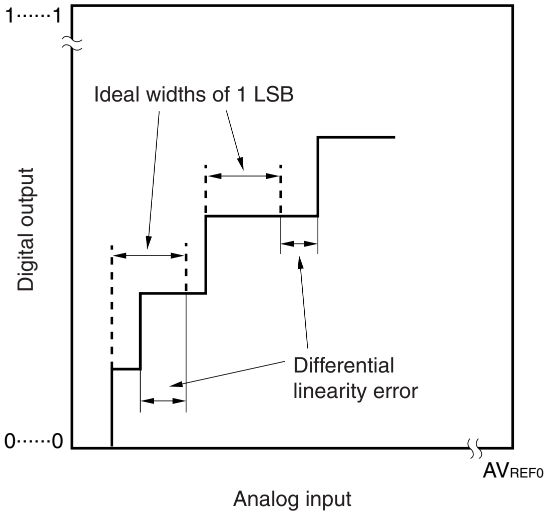
Figure 11-8. Full-Scale Error



(6) Differential linearity error

Ideally, the width to output a specific code is 1 LSB. This error indicates the difference between the actually measured value and its theoretical value when a specific code is output.

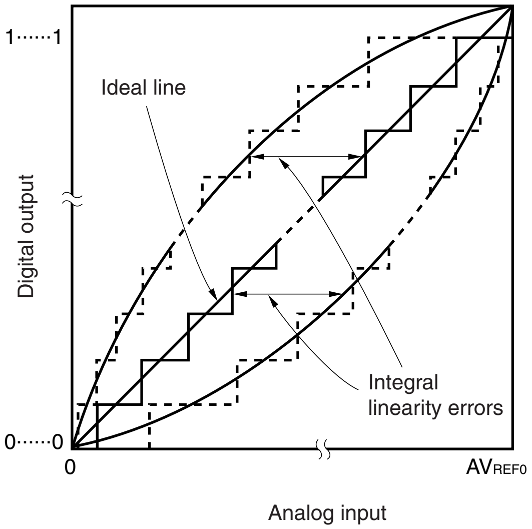
Figure 11-9. Differential Linearity Error



(7) Integral linearity error

This error indicates the extent to which the conversion characteristics differ from the ideal linear relations. It indicates the maximum value of the difference between the actually measured value and its theoretical value where the zero-scale error and full-scale error are 0.

Figure 11-10. Integral Linearity Error



(8) Conversion time

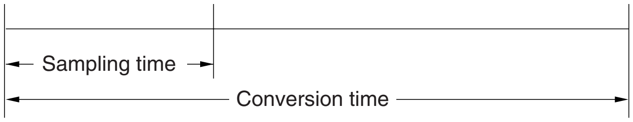
This is the time required to obtain a digital output after an analog input voltage has been assigned.

The conversion time in the characteristics table includes sampling time.

(9) Sampling time

This is the time during which the analog switch is ON to load an analog voltage to the sample & hold circuit.

Figure 11-11. Sampling Time



CHAPTER 12 D/A CONVERTER

12.1 Functions

The D/A converter has the following functions.

○ 8-bit resolution × 2 channels (DAC0, DAC1)

○ R string method

○ Conversion time: 20 μs max. (AV_{REF1} = 2.2 to 2.7 V)

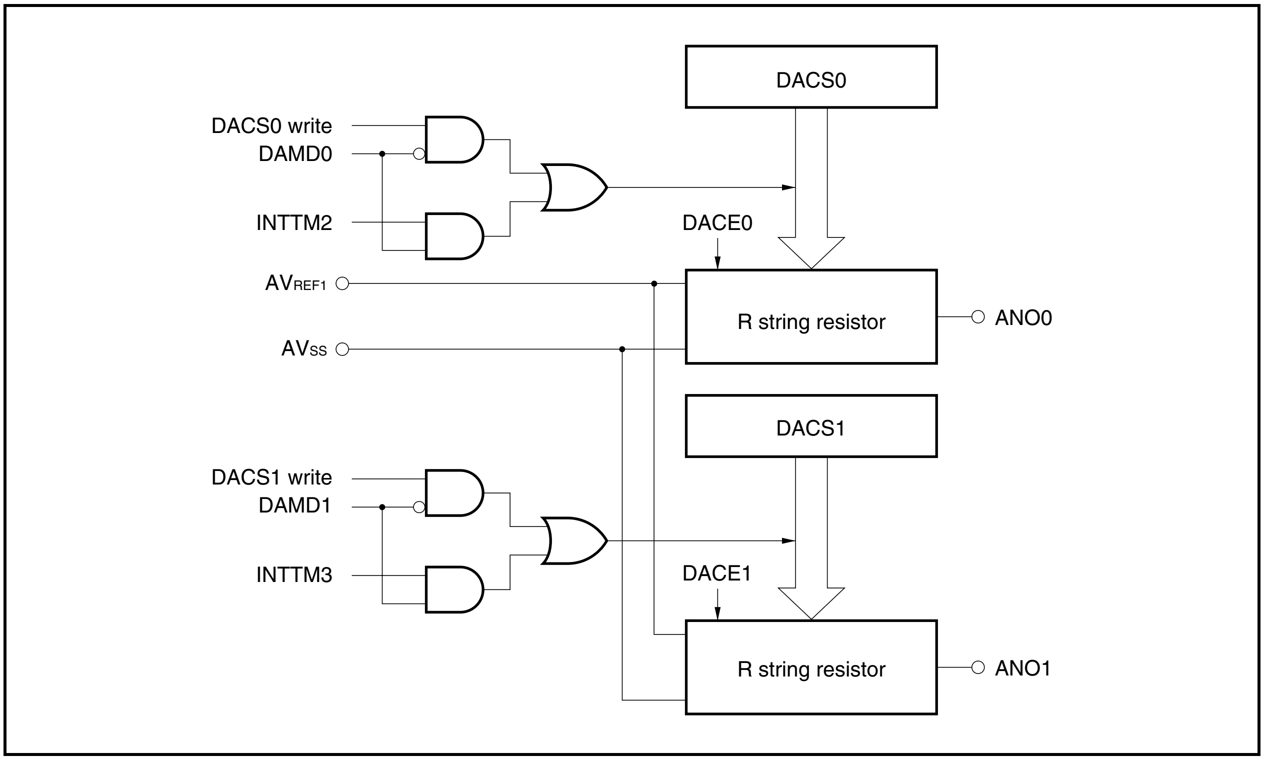
○ Analog output voltage: AV_{REF1} × m/256 (m = 0 to 255; value set to DACSn register)

○ Operation modes: Normal mode, real-time output mode

Remark n = 0, 1

The D/A converter configuration is shown below.

Figure 12-1. Block Diagram of D/A Converter



12.2 Configuration

The D/A converter consists of the following hardware.

Table 12-1. Configuration of D/A Converter

Item	Configuration
Control registers	D/A converter mode register (DAM) D/A conversion value setting registers 0 and 1 (DACS0 and DACS1)

12.3 Control Registers

The registers that control the D/A converter are as follows.

- D/A converter mode register (DAM)
- D/A conversion value setting registers 0 and 1 (DACS0 and DACS1)

(1) D/A converter mode register (DAM)

This register controls the operation of the D/A converter.
DAM can be read or written in 8-bit or 1-bit units.
This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF284H

	7	6	5	4	3	<2>	1	<0>
DAM	0	0	0	0	DAMD1	DACE1	DAMD0	DACE0

DAMDn	Selection of D/A converter operation mode (n = 0, 1)
0	Normal mode
1	Real-time output mode[Note]

DACEn	D/A converter operation enable/disable control (n = 0, 1)
0	Disables operation
1	Enables operation

Note The output trigger in the real-time output mode (DAMDn bit = 1) is as follows.
- When n = 0: INTTM2 signal (refer to **CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 2 TO 5**)
- When n = 1: INTTM3 signal (refer to **CHAPTER 8 8-BIT TIMER/EVENT COUNTERS 2 TO 5**)

(2) D/A conversion value setting registers 0 and 1 (DACS0 and DACS1)

These registers set the analog voltage value output to the ANO0 and ANO1 pins.

These registers can be read or written in 8-bit units.

These registers are cleared to 00H after reset.

After reset: 00H R/W Address: DACS0 FFFFF280H, DACS1 FFFFF282H

	7	6	5	4	3	2	1	0
DACSn	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

Caution In the real-time output mode (DAMDn bit = 1), set the DACSn register before the INTTM2/INTTM3 signals are generated. D/A conversion starts when the INTTM2/INTTM3 signals are generated.

Remark n = 0, 1

12.4 Operation

12.4.1 Operation in normal mode

D/A conversion is performed using a write operation to D/A conversion value setting register n (DACSn) as the trigger.

The setting method is described below.

<1> Set the DAMDn bit of the D/A converter mode register (DAM) to 0 (normal mode).
<2> Set the analog voltage value to be output to the ANOn pin to the DACSn register.
Steps <1> and <2> above constitute the initial settings.
<3> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts when this setting is performed.
<4> To perform subsequent D/A conversions, write to the DACSn register.
The previous D/A conversion result is held until the next D/A conversion is performed.

Remark n = 0, 1

12.4.2 Operation in real-time output mode

D/A conversion is performed using the interrupt request signals (INTTM2 and INTTM3) of 8-bit timer/event counters 2 and 3 (TM2 and TM3) as triggers.

The setting method is described below.

<1> Set the DAMDn bit of the DAM register to 1 (real-time output mode).
<2> Set the analog voltage value to be output to the ANOn pin to the DACSn register.
<3> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).
Steps <1> to <3> above constitute the initial settings.
<4> Operate 8-bit timer/event counters 2 and 3 (TM2 and TM3).
<5> D/A conversion starts when the INTTM2 and INTTM3 signals are generated.
<6> The INTTM2 and INTTM3 signals are generated when subsequent D/A conversions are performed.
Before performing the next D/A conversion (generation of INTTM2 and INTTM3 signals), set the analog voltage value to be output to the ANOn pin to the DACSn register.

12.4.3 Cautions

Observe the following cautions when using the D/A converter of the V850ES/SA2 and V850ES/SA3.

(1) Do not change the set value of the DACSn register while the trigger signal is being issued in the real-time output mode.

(2) Before changing the operation mode, be sure to clear DACEn to 0.

(3) When using only one channel of the D/A converter, connect the pins that are not used as analog outputs to V_{SS}.

(4) When using the P80/ANO0 and P81/ANO1 pins as port pins (when not using the D/A converter), make sure that their input level does not change much.

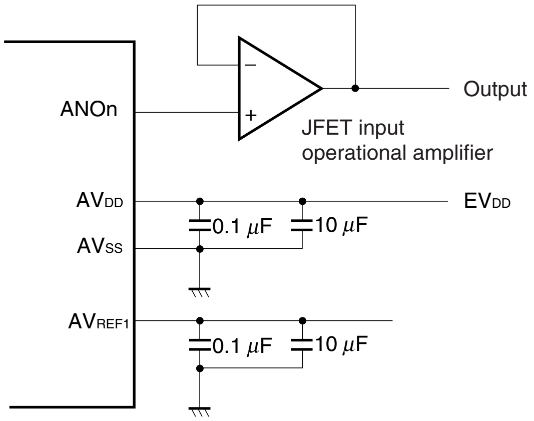
(5) When not using the D/A converter in the standby mode, clear the DACS bit to 0, and then change to the standby mode. Connecting the AV_{REF1} pin to V_{SS} further reduces the power consumption.

(6) Make sure that $V_{DD} = EV_{DD} = AV_{DD} = AV_{REF1} = 2.2$ to 2.7 V. If this range is exceeded, the operation is not guaranteed.

(7) Apply power to AV_{DD} at the same timing as V_{DD}.

(8) No current can be output from the ANOn pin (n = 0, 1) because the output impedance of the D/A converter is high. When connecting a resistor of 5 MΩ or less, insert a JFET input operational amplifier between the resistor and the ANOn pin.

Figure 12-2. External Pin Connection Example



CHAPTER 13 ASYNCHRONOUS SERIAL INTERFACE n (UARTn)

13.1 Features

- Transfer rate: 300 bps to 312.5 kbps (using a dedicated baud rate generator and an internal system clock of 20 MHz)
- Full-duplex communications

  On-chip receive buffer register n (RXBn)

  On-chip transmit buffer register n (TXBn)
- Two-pin configuration

  TXDn: Transmit data output pin

  RXDn: Receive data input pin
- Reception error detection functions
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3 types
  - Reception error interrupt (INTSREn): Interrupt is generated according to the logical OR of the three types of reception errors
  - Reception completion interrupt (INTSRn): Interrupt is generated when receive data is transferred from the shift register to receive buffer register n after serial transfer is completed during a reception enabled state
  - Transmission completion interrupt (INTSTn): Interrupt is generated when the serial transmission of transmit data (8 or 7 bits) from the shift register is completed
- The character length of transmit/receive data is specified by the ASIMn register
- Character length: 7 or 8 bits
- Parity functions: Odd, even, 0, or none
- Transmission stop bits: 1 or 2 bits
- On-chip dedicated baud rate generator

Remark n = 0, 1

13.1.1 Switching modes between UART0 and CSI1

CSI1 and UART0 of the V850ES/SA2 and V850ES/SA3 share pins, and therefore these interfaces cannot be used at the same time. Select CSI1 or UART0 in advance by using port mode control register 3 (PMC3) and port function control register 3 (PFC3) (refer to **4.3.3 Port 3**).

Caution CSI1 or UART0 transmission/reception operations are not guaranteed if the mode is changed during transmission or reception. Be sure to disable the operation of the unit that is not used.

Figure 13-1. Selecting CSI1 or UART0 Mode

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	0	0	0	0	0	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	0	0	0	0	PFC31	PFC30

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	CSI1 mode
1	1	UART0 mode

Remarks 1. n = 0, 1

2. × = Don't care

13.2 Configuration

UARTn is controlled by asynchronous serial interface mode register n (ASIMn), asynchronous serial interface status register n (ASISn), and asynchronous serial interface transmission status register n (ASIFn). Receive data is maintained in receive buffer register n (RXBn), and transmit data is written to transmit buffer register n (TXBn).

Figure 13-3 shows the configuration of asynchronous serial interface n (UARTn).

(1) Asynchronous serial interface mode register n (ASIMn)

The ASIMn register is an 8-bit register for specifying the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status register n (ASISn)

The ASISn register consists of a set of flags that indicate the error contents when a reception error occurs. The various reception error flags are set (1) when a reception error occurs and are reset (0) when the ASISn register is read.

(3) Asynchronous serial interface transmission status register n (ASIFn)

The ASIFn register is an 8-bit register that indicates the status when a transmit operation is performed.

This register consists of a transmit buffer data flag, which indicates the hold status of TXBn data, and the transmit shift register data flag, which indicates whether transmission is in progress.

(4) Reception control parity check

The receive operation is controlled according to the contents set in the ASIMn register. A check for parity errors is also performed during a receive operation, and if an error is detected, a value corresponding to the error contents is set in the ASISn register.

(5) Receive shift register

This is a shift register that converts the serial data that was input to the RXDn pin into parallel data. One byte of data is received, and if a stop bit is detected, the receive data is transferred to receive buffer register n (RXBn).

This register cannot be directly manipulated.

(6) Receive buffer register n (RXBn)

RXBn is an 8-bit buffer register for holding receive data. When 7 characters are received, 0 is stored in the MSB.

During a reception enabled state, receive data is transferred from the receive shift register to RXBn, synchronized with the end of the shift-in processing of one frame.

Also, the reception completion interrupt request (INTSRn) is generated by the transfer of data to RXBn.

(7) Transmit shift register

This is a shift register that converts the parallel data that was transferred from transmit buffer register n (TXBn) into serial data.

When one byte of data is transferred from TXBn, the shift register data is output from the TXDn pin.

This register cannot be directly manipulated.

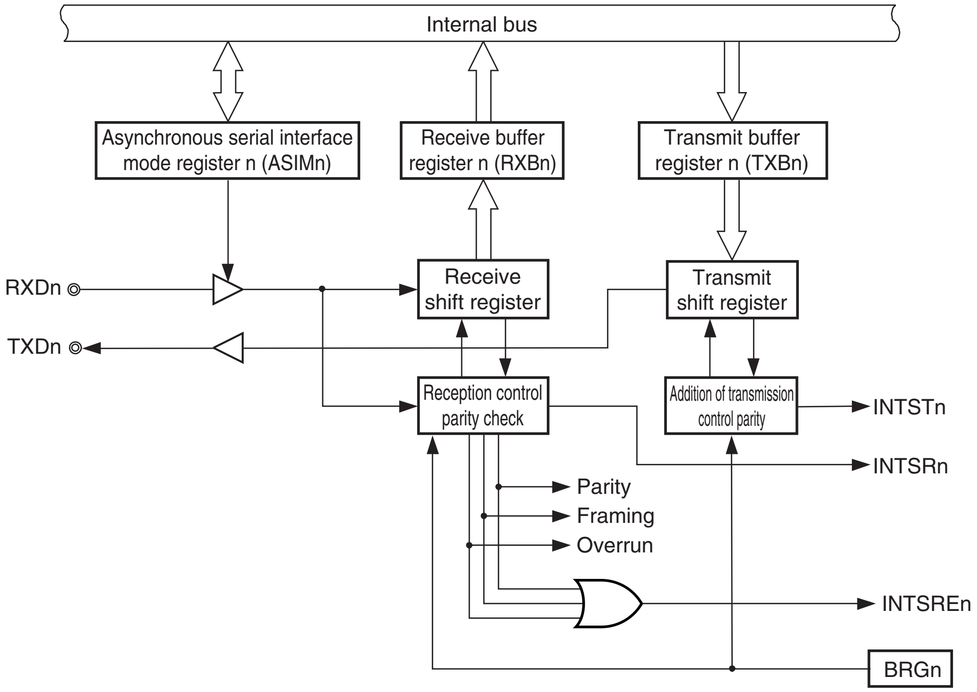
(8) Transmit buffer register n (TXBn)

TXBn is an 8-bit buffer for transmit data. A transmit operation is started by writing transmit data to TXBn. The transmission completion interrupt request (INTSTn) is generated synchronized with the completion of transmission of one frame.

(9) Addition of transmission control parity

A transmit operation is controlled by adding a start bit, parity bit, or stop bit to the data that is written to the TXBn register, according to the contents that were set in the ASIMn register.

Figure 13-2. Block Diagram of Asynchronous Serial Interface n


Internal bus
Asynchronous serial interface mode register n (ASIMn)
Receive buffer register n (RXBn)
Transmit buffer register n (TXBn)
RXDn
TXDn
Receive shift register
Transmit shift register
Reception control parity check
Addition of transmission control parity
INTSTn
INTSRn
Parity
Framing
Overrun
INTSREn
BRGn

Remark n = 0, 1

13.3 Control Registers

(1) Asynchronous serial interface mode register n (ASIMn)

The ASIMn register is an 8-bit register that controls the UARTn transfer operation.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 01H after reset.

Caution When using UARTn, be sure to set the external pins related to UARTn functions to the control mode before setting clock select register n (CKSRn) and baud rate generator control register n (BRGCn), and then set the UARTCAEn bit to 1. Then set the other bits.

(1/3)

After reset: 01H R/W Address: ASIM0 FFFFFA00H, ASIM1 FFFFFA10H

	<7>	<6>	<5>	4	3	2	1	0
ASIMn (n = 0, 1)	UARTCAEn	TXEn	RXEn	PSn1	PSn0	CLn	SLn	ISRMn

UARTCAEn	Controls the operating clock
0	Stops clock supply to UARTn.
1	Supplies clock to UARTn.

- If UARTCAEn = 0, UARTn is asynchronously reset.
- If UARTCAEn = 0, UARTn is reset. To operate UARTn, first set UARTCAEn to 1.
- If the UARTCAEn bit is changed from 1 to 0, all the registers of UARTn are initialized. To set UARTCAEn to 1 again, be sure to re-set the registers of UARTn.
- The output of the TXDn pin goes high when transmission is disabled, regardless of the setting of the UARTCAEn bit.

TXEn	Enables/disables transmission
0	Disables transmission
1	Enables transmission

- Set the TXEn bit to 1 after setting the UARTCAEn bit to 1 at startup. Set the UARTCAEn bit to 0 after setting the TXEn bit to 0 to stop.
- To initialize the transmission unit, clear (0) the TXEn bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the TXEn bit again. If the TXEn bit is not set again, initialization may not be successful. (For details of the base clock, refer to **13.6 (1) (a) Base clock (Clock)**.)

(2/3)

RXEn	Enables/disables reception
0	Disables reception
1	Enables reception

- Set the RXEn bit to 1 after setting the UARTCAEn bit to 1 at startup. Set the UARTCAEn bit to 0 after setting the RXEn bit to 0 to stop.
- To initialize the reception unit status, clear (0) the RXEn bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the RXEn bit again. If the RXEn bit is not set again, initialization may not be successful. (For details of the base clock, refer to **13.6 (1) (a) Base clock (Clock)**.)

PSn1	PSn0	Transmit operation	Receive operation
0	0	Parity bit not output	Receive with no parity
0	1	Output 0 parity	Receive as 0 parity
1	0	Output odd parity	Judge as odd parity
1	1	Output even parity	Judge as even parity

- To overwrite the PS1 and PS0 bits, first clear (0) the TXEn and RXEn bits.
- If “0 parity” is selected for reception, no parity judgment is performed. Therefore, no error interrupt is generated because the PE bit of the ASISn register is not set.

Remarks 1. When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to receive buffer register n (RXBn) is performed, and the contents of the RXBn register are retained.

When reception is enabled, the receive shift operation starts, synchronized with the detection of the start bit, and when the reception of one frame is completed, the contents of the receive shift register are transferred to the RXBn register. A reception completion interrupt (INTSRn) is also generated in synchronization with the transfer to the RXBn register.

2. (Even parity)

If the transmit data contains an odd number of bits with the value “1”, the parity bit is set (1). If it contains an even number of bits with the value “1”, the parity bit is cleared (0). This controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an even number.

During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is odd, a parity error is generated.

(Odd parity)

In contrast to even parity, odd parity controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an odd number.

During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is even, a parity error is generated.

(0 parity)

During transmission, the parity bit is cleared (0) regardless of the transmit data.

During reception, no parity error is generated because no parity bit is checked.

(No parity)

No parity bit is added to transmit data.

During reception, the receive data is considered to have no parity bit. No parity error is generated because there is no parity bit.

(3/3)

CLn	Specifies character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits
• To overwrite the CL bit, first clear (0) the TXEn and RXEn bits.	

SLn	Specifies stop bit length of transmit data
0	1 bit
1	2 bits
• To overwrite the SL bit, first clear (0) the TXEn bit. • Since reception is always performed with a stop bit length of 1, the SL bit setting does not affect receive operations.	

ISRMn	Enables/disables generation of reception completion interrupt requests when an error occurs
0	Generate a reception error interrupt request (INTSREn) as an interrupt when an error occurs. In this case, no reception completion interrupt request (INTSRn) is generated.
1	Generate a reception completion interrupt request (INTSRn) as an interrupt when an error occurs. In this case, no reception error interrupt request (INTSREn) is generated.
• To overwrite the ISRM bit, first clear (0) the RXEn bit.	

(2) Asynchronous serial interface status register n (ASISn)

The ASISn register, which consists of 3 error flag bits (PEn, FEn and OVEn), indicates the error status when UARTn reception is complete.

The status flag, which indicates a reception error, always indicates the status of the error that occurred most recently. That is, if the same error occurred several times before the receive data was read, this flag would hold only the status of the error that occurred last.

The ASISn register is cleared to 00H by a read operation. When a reception error occurs, receive buffer register n (RXBn) should be read and the error flag should be cleared after the ASISn register is read.

This register is read-only, in 8-bit units.

This register is cleared to 00H after reset.

Caution When the UARTCAEn bit or RXEn bit of the ASIMn register is set to 0, or when the ASISn register is read, the PEn, FEn, and OVEn bits of the ASISn register are cleared (0).

After reset: 00H R Address: ASIS0 FFFFFA03H, ASIS1 FFFFFA13H

	7	6	5	4	3	2	1	0
ASISn (n = 0, 1)	0	0	0	0	0	PEn	FEn	OVEn

PEn	Status flag that indicates a parity error
0	When the ASIMn register's UARTCAEn and RXEn bits are both set to 0, or when the ASISn register has been read
1	When reception was completed, the transmit data parity did not match the parity bit
• The operation of the PEn bit differs according to the settings of the PS1 and PS0 bits of the ASIMn register.	

FEn	Status flag that indicates a framing error
0	When the ASIMn register's UARTCAEn and RXEn bits are both set to 0, or when the ASISn register has been read
1	When reception was completed, no stop bit was detected
• For receive data stop bits, only the first bit is checked regardless of the stop bit length.	

OVEn	Status flag that indicates an overrun error
0	When the ASIMn register's UARTCAEn and RXEn bits are both 0, or when the ASISn register has been read.
1	UARTn completed the next receive operation before reading the RXBn receive data.
• When an overrun error occurs, the next receive data value is not written to the RXBn register and the data is discarded.	

(3) Asynchronous serial interface transmission status register n (ASIFn)

The ASIFn register, which consists of 2 status flag bits, indicates the status during transmission.

By writing the next data to the TXBn register after data is transferred from the TXBn register to the transmit shift register, transmit operations can be performed continuously without suspension even during an interrupt interval. When transmission is performed continuously, data should be written after referencing the TXBFn bit of the ASIFn register to prevent writing to the TXBn register by mistake.

This register is read-only, in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R Address: ASIF0 FFFFFA05H, ASIF1 FFFFFA15H

	7	6	5	4	3	2	<1>	<0>
ASIFn (n = 0, 1)	0	0	0	0	0	0	TXBFn	TXSFn

TXBFn	Transmit buffer data flag
0	Data to be transferred next to the TXBn register does not exist (when the ASIMn register’s POWERn or TXEn bit is 0, or when data has been transferred to the transmit shift register)
1	Data to be transferred next exists in the TXBn register (data exists in the TXBn register when the TXBn register has been written to)
• When transmission is performed continuously, data should be written to the TXBn register after confirming that this flag is 0. If writing to the TXBn register is performed when this flag is 1, transmit data cannot be guaranteed.	

TXSFn	Transmit shift register data flag (indicating the transmission status of UARTn)
0	Initial status or a waiting transmission (when the ASIMn register’s UARTCAEn or TXEn bit is set to 0, or following transfer completion, the next data transfer from the TXBn register is not performed)
1	Transmission in progress (when data has been transferred from the TXBn register)
• When the transmission unit is initialized, initialization should be executed after confirming that this flag is 0 following the occurrence of a transmission completion interrupt. If initialization is performed when this flag is 1, transmit data cannot be guaranteed.	

(4) Receive buffer register n (RXBn)

The RXBn register is an 8-bit buffer register for storing parallel data that had been converted by the receive shift register.

When reception is enabled (RXEn bit = 1 in the ASIMn register), receive data is transferred from the receive shift register to the RXBn register, synchronized with the completion of the shift-in processing of one frame. Also, a reception completion interrupt request (INTSRn) is generated by the transfer to the RXBn register. For information about the timing for generating this interrupt request, refer to **13.5 (4) Receive operation**.

If reception is disabled (RXEn bit = 0 in the ASIMn register), the contents of the RXBn register are retained, and no processing is performed for transferring data to the RXBn register even when the shift-in processing of one frame is completed. Also, no reception completion interrupt is generated.

When 7 bits is specified for the data length, bits 6 to 0 of the RXBn register are transferred for the receive data and the MSB (bit 7) is always 0. However, if an overrun error (OVEn) occurs, the receive data at that time is not transferred to the RXBn register.

Except when a reset is input, the RXBn register becomes FFH even when UARTCAEn = 0 in the ASIMn register.

This register is read-only, in 8-bit units.

This register is set to FFH after reset.

After reset: FFH R Address: RXB0 FFFFFA02H, RXB1 FFFFFA12H

	7	6	5	4	3	2	1	0
RXBn (n = 0, 1)	RXBn7	RXBn6	RXBn5	RXBn4	RXBn3	RXBn2	RXBn1	RXBn0

(5) Transmit buffer register n (TXBn)

The TXBn register is an 8-bit buffer register for setting transmit data.

When transmission is enabled (TXEn bit = 1 in the ASIMn register), the transmit operation is started by writing data to TXBn register.

When transmission is disabled (TXEn bit = 0 in the ASIMn register), even if data is written to TXBn register, the value is ignored.

The TXBn register data is transferred to the transmit shift register, and a transmission completion interrupt request (INTSTn) is generated, synchronized with the completion of the transmission of one frame from the transmit shift register. For information about the timing for generating this interrupt request, refer to **13.5 (2) Transmit operation**.

When TXBFn bit = 1 in the ASIFn register, the TXBn register must not be written.

This register can be read or written in 8-bit units.

This register is set to FFH after reset.

After reset: FFH R/W Address: TXB0 FFFFFA04H, TXB1 FFFFFA14H

	7	6	5	4	3	2	1	0
TXBn (n = 0, 1)	TXBn7	TXBn6	TXBn5	TXBn4	TXBn3	TXBn2	TXBn1	TXBn0

13.4 Interrupt Requests

The following three types of interrupt requests are generated from UARTn.

- Reception error interrupt (INTSREn)
- Reception completion interrupt (INTSRn)
- Transmission completion interrupt (INTSTn)

The default priorities among these three types of interrupt requests is, from high to low, reception error interrupt, reception completion interrupt, and transmission completion interrupt.

Table 13-1. Generated Interrupts and Default Priorities

Interrupt	Priority
Reception error	1
Reception completion	2
Transmission completion	3

(1) Reception error interrupt (INTSREn)

When reception is enabled, a reception error interrupt is generated according to the logical OR of the three types of reception errors explained for the ASISn register. Whether a reception error interrupt (INTSREn) or a reception completion interrupt (INTSRn) is generated when an error occurs can be specified using the ISRMn bit of the ASIMn register.

When reception is disabled, no reception error interrupt is generated.

(2) Reception completion interrupt (INTSRn)

When reception is enabled, a reception completion interrupt is generated when data is shifted in to the receive shift register and transferred to the receive buffer register (RXBn).

A reception completion interrupt request can be specified to be generated in place of a reception error interrupt using the ISRMn bit of the ASIMn register even when a reception error has occurred.

When reception is disabled, no reception completion interrupt is generated.

(3) Transmission completion interrupt (INTSTn)

A transmission completion interrupt is generated when one frame of transmit data containing 7-bit or 8-bit characters is shifted out from the transmit shift register.

13.5 Operation

(1) Data format

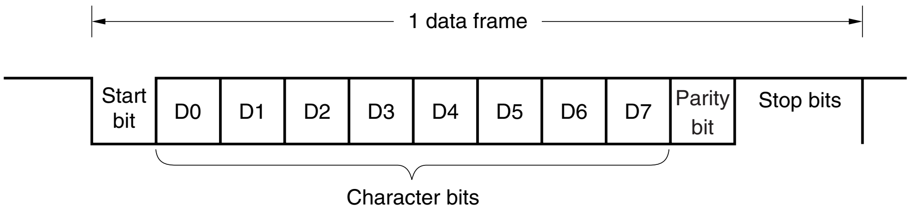
Full-duplex serial data transmission and reception can be performed.

The transmit/receive data format consists of one data frame containing a start bit, character bits, a parity bit, and stop bits as shown in Figure 13-3.

The character bit length within one data frame, the type of parity, and the stop bit length are specified by asynchronous serial interface mode register n (ASIMn).

Also, data is transferred LSB first.

Figure 13-3. Asynchronous Serial Interface Transmit/Receive Data Format



- Start bit .... 1 bit
- Character bits .... 7 bits or 8 bits
- Parity bit .... Even parity, odd parity, 0 parity, or no parity
- Stop bits .... 1 bit or 2 bits

(2) Transmit operation

When the UARTCAEn bit is set to 1 in the ASIMn register, a high level is output from the TXDn pin.

Then, when the TXEn bit is set to 1 in the ASIMn register, transmission is enabled, and the transmit operation is started by writing transmit data to transmit buffer register n (TXBn).

(a) Transmission enabled state

This state is set by the TXEn bit in the ASIMn register.

- TXEn = 1: Transmission enabled state
- TXEn = 0: Transmission disabled state

Since UARTn does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in a reception enabled state.

(b) Starting a transmit operation

In the transmission enabled state, a transmit operation is started by writing transmit data to transmit buffer register n (TXBn). When a transmit operation is started, the data in TXBn is transferred to the transmit shift register. Then, the transmit shift register outputs data to the TXDn pin (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

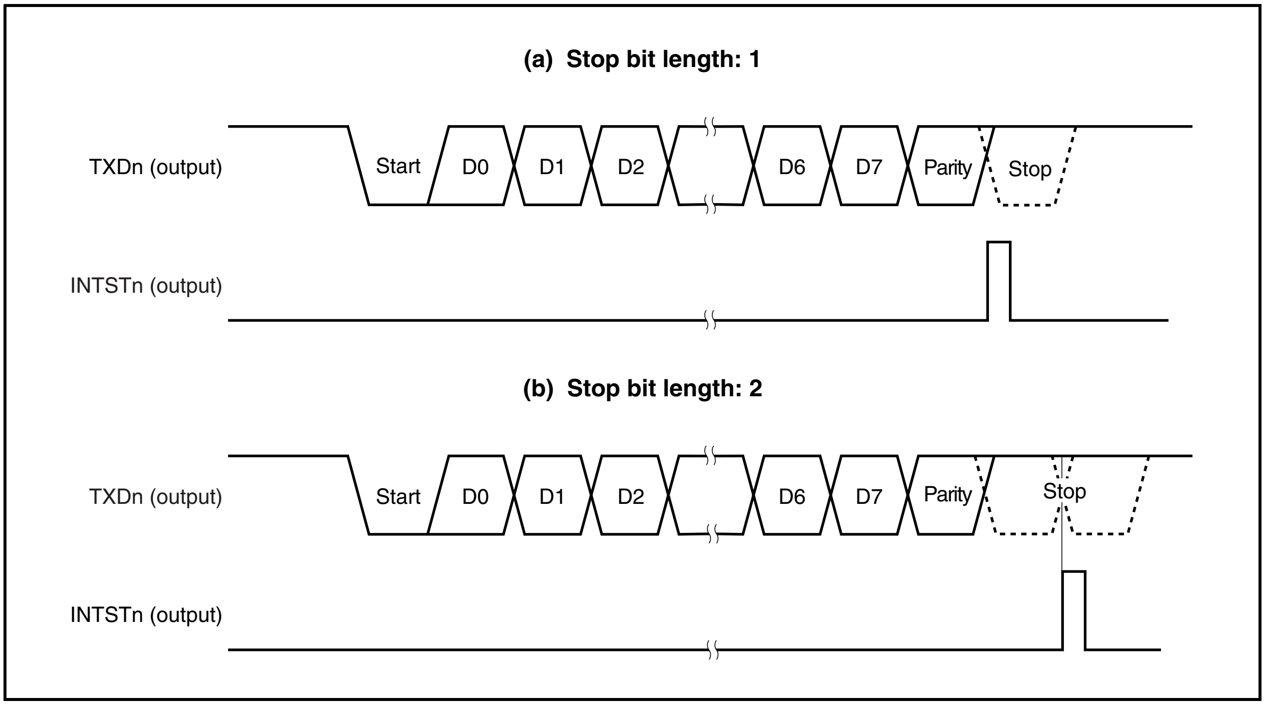
(c) Transmission interrupt request

When the transmit shift register becomes empty, a transmission completion interrupt request (INTSTn) is generated. The timing for generating the INTSTn interrupt differs according to the specification of the stop bit length. The INTSTn interrupt is generated at the same time that the last stop bit is output.

If the data to be transmitted next has not been written to the TXBn register, the transmit operation is suspended.

Caution Normally, when the transmit shift register becomes empty, a transmission completion interrupt (INTSTn) is generated. However, no transmission completion interrupt (INTSTn) is generated if the transmit shift register becomes empty due to reset.

Figure 13-4. Asynchronous Serial Interface Transmission Completion Interrupt Timing



(3) Continuous transmission operation

UARTn can write the next transmit data to the TXBn register at the timing that the transmit shift register starts the shift operation. This enables an efficient transmission rate to be realized by continuously transmitting data even during the INTSTn interrupt servicing after the transmission of one data frame. In addition, reading the TXSFn bit of the ASIFn register after the occurrence of a transmission completion interrupt enables the TXBn register to be efficiently written twice (2 bytes) without waiting for the transmission of 1 data frame.

When continuous transmission is performed, data should be written after referencing the ASIFn register to confirm the transmission status and whether or not data can be written to the TXBn register.

TXBFn	Whether or Not Writing to TXBn Register Is Enabled
0	Writing is enabled
1	Writing is not enabled

Caution When transmission is performed continuously, write the first transmit data (first byte) to the TXBn register and confirm that the TXBFn bit is 0, and then write the next transmit data (second byte) to TXBn register. If writing to the TXBn register is performed when the TXBFn bit is 1, transmit data cannot be guaranteed.

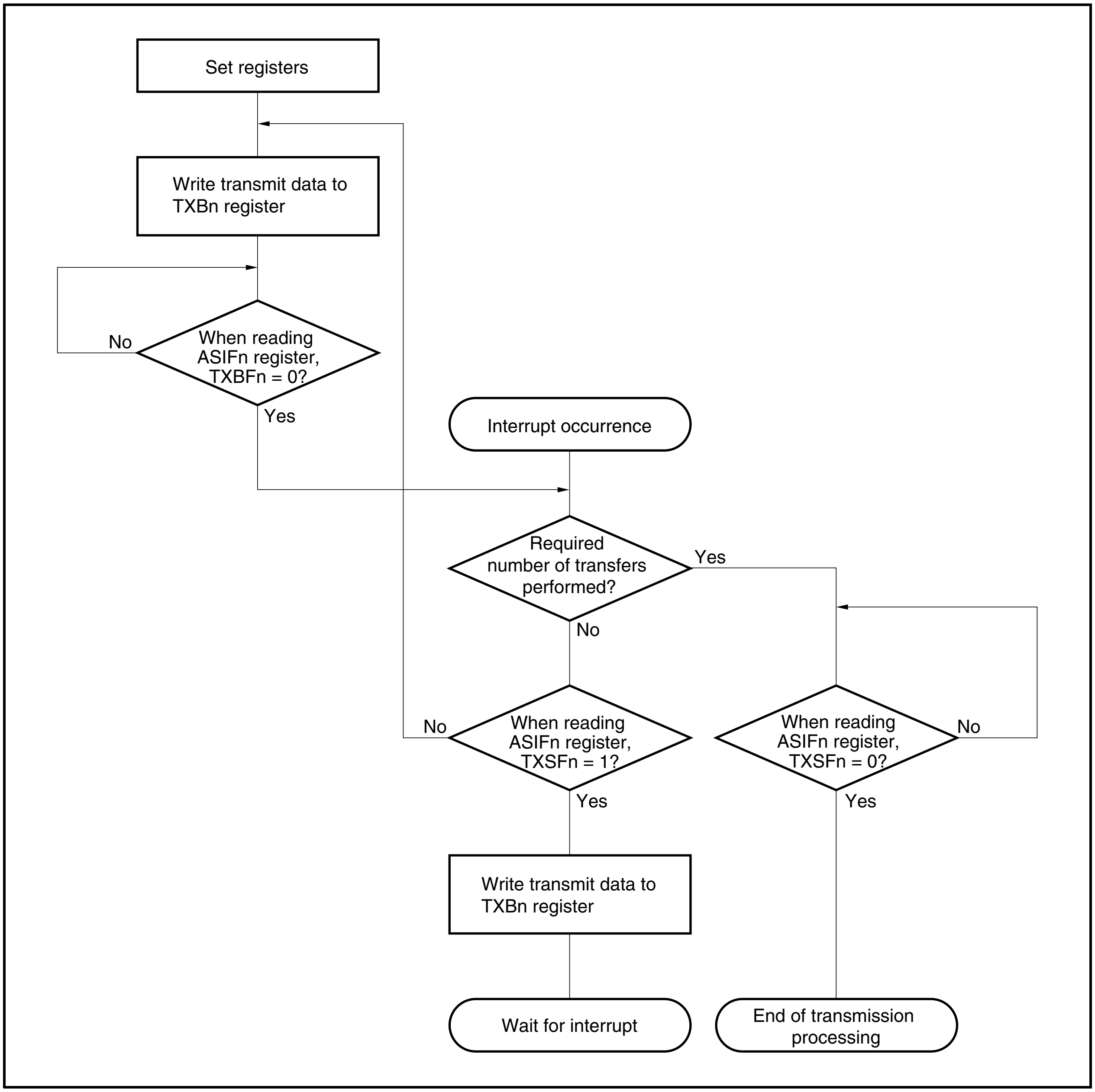
While transmission is being performed continuously, whether writing to the TXBn register later is enabled can be judged by confirming the TXSFn bit after the occurrence of a transmission completion interrupt.

TXSFn	Transmission Status
0	Transmission is completed. However, the cautions concerning the TXBFn bit must be observed. Writing transmit data can be performed twice (2 bytes).
1	Under transmission. Transmit data can be written once (1 byte).

Cautions 1. When initializing the transmission unit when continuous transmission is completed, confirm that the TXBFn bit is 0 after the occurrence of the transmission completion interrupt, and then execute initialization. If initialization is performed when the TXBFn bit is 1, transmit data cannot be guaranteed.

2. While transmission is being performed continuously, an overrun error may occur if the next transmission is completed before the INTSTn interrupt servicing following the transmission of 1 data frame is executed. An overrun error can be detected by embedding a program that can count the number of transmit data and referencing the TXSFn bit.

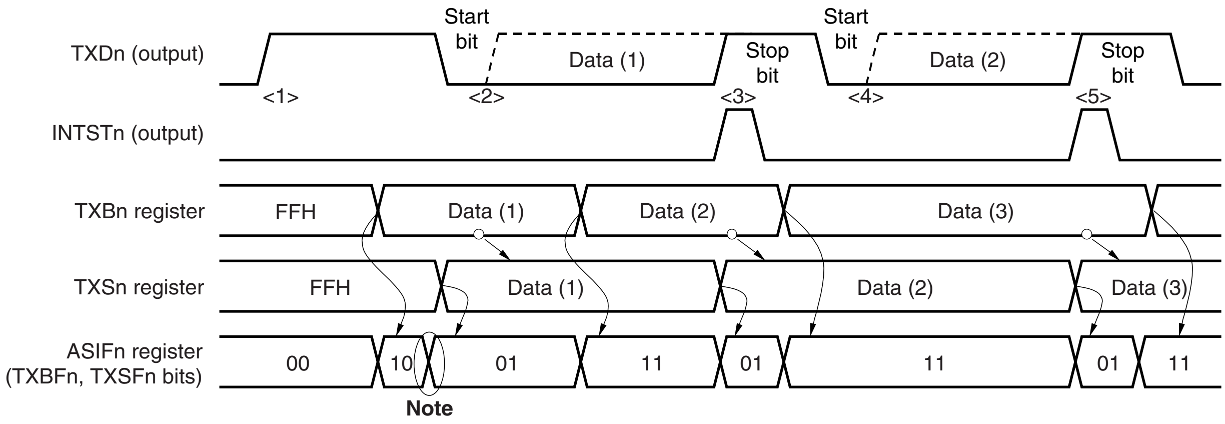
Figure 13-5. Continuous Transmission Processing Flow


Set registers
Write transmit data to TXBn register
No
When reading ASIFn register, TXBFn = 0?
Yes
Interrupt occurrence
Required number of transfers performed?
Yes
No
No
When reading ASIFn register, TXSFn = 1?
Yes
When reading ASIFn register, TXSFn = 0?
No
Yes
Write transmit data to TXBn register
Wait for interrupt
End of transmission processing

(a) Starting procedure

The procedure to start continuous transmission is shown below.

Figure 13-6. Continuous Transmission Starting Procedure



Note Since this period is a transition period from 10 to 01, when reading the TXBFn and TXSFn bits of the ASIFn register simultaneously, 11 or 00 may be read. Thus, whether writing to the TXBn register is enabled or not should be judged only for the TXBFn bit.

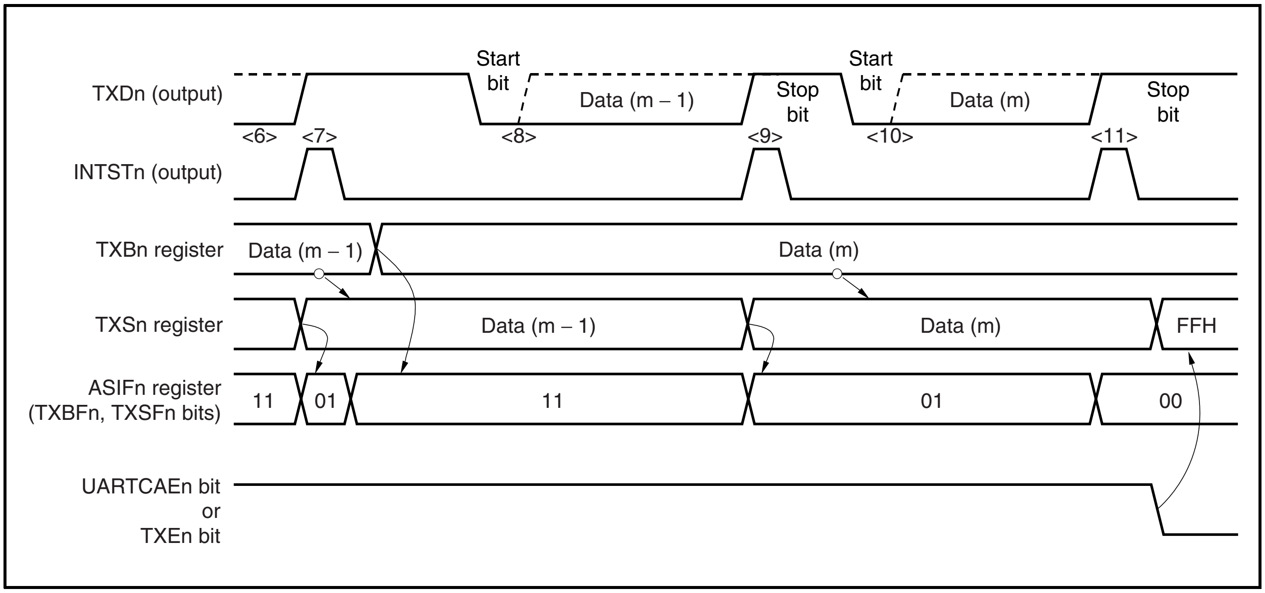
Transmission Starting Procedure	Internal Operation	ASIFn Register	
		TXBFn	TXSFn
• Set transmission mode	<1> Start transmission unit	0	0
• Write data (1) →		1	0
	<2> Generate start bit	1	1/0[Note]
		0	1/0[Note]
	Start data (1) transmission →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0) ←		0	1
• Write data (2) →		1	1
	<<Transmission in progress>>		
	<3> INTSTn interrupt occurs →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0) ←		0	1
• Write data (3) →		1	1
	<4> Generate start bit		
	Start data (2) transmission		
	<<Transmission in progress>>		
	<5> INTSTn interrupt occurs →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0) ←		0	1
• Write data (4) →		1	1

Note Transition period

(b) Ending procedure

The procedure for ending continuous transmission is shown below.

Figure 13-7. Continuous Transmission End Procedure



Transmission End Procedure	Internal Operation	ASIFn Register	
		TXBFn	TXSFn
	<6> Transmission of data (m – 2) is in progress	1	1
	<7> INTSTn interrupt occurs →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0) ←		0	1
• Write data (m) →		1	1
	<8> Generate start bit Start data (m – 1) transmission <<Transmission in progress>>		
	<9> INTSTn interrupt occurs →	0	1
• Read ASIFn register (confirm that TXSFn bit = 1) ←		0	1
There is no write data			
	<10> Generate start bit Start data (m) transmission <<Transmission in progress>>		
	<11> Generate INTSTn interrupt →	0	0
• Read ASIFn register (confirm that TXSFn bit = 0) ←		0	0
• Clear (0) the UARTCAEn bit or TXEn bit	Initialize internal circuits		

(4) Receive operation

The awaiting reception state is set by setting the UARTCAEn bit to 1 in the ASIMn register and then setting the RXEn bit to 1 in the ASIMn register. To start the receive operation, first perform start bit detection. The start bit is detected by sampling the RXDn pin. When the receive operation begins, serial data is stored sequentially in the receive shift register according to the baud rate that was set. A reception completion interrupt (INTSRn) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from receive buffer register n (RXBn) to memory by this interrupt servicing.

(a) Reception enabled state

The receive operation is set to the reception enabled state by setting the RXEn bit in the ASIMn register to 1.

- RXEn bit = 1: Reception enabled state
- RXEn bit = 0: Reception disabled state

In reception disabled state, the reception hardware stands by in the initial state. At this time, the contents of receive buffer register n (RXBn) are retained, and no reception completion interrupt or reception error interrupt is generated.

(b) Starting a receive operation

A receive operation is started by the detection of a start bit.

The RXDn pin is sampled using the serial clock from baud rate generator n (BRGn).

(c) Reception completion interrupt

When RXEn = 1 in the ASIMn register and the reception of one frame of data is completed (the stop bit is detected), a reception completion interrupt (INTSRn) is generated and the receive data within the receive shift register is transferred to RXBn at the same time.

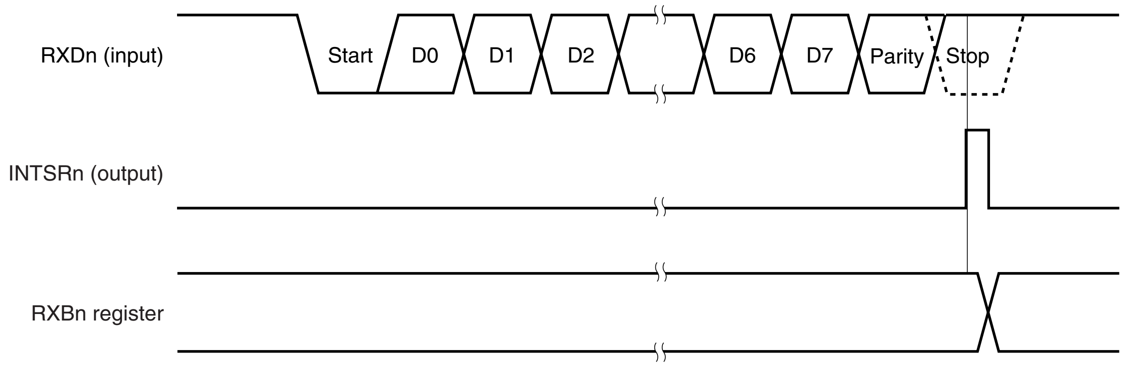
Also, if an overrun error (OVEn) occurs, the receive data at that time is not transferred to receive buffer register n (RXBn), and either a reception completion interrupt (INTSRn) or a reception error interrupt (INTSREn) is generated (the receive data within the receive shift register is transferred to RXBn) according to the ISRMn bit setting in the ASIMn register.

Even if a parity error (PEn) or framing error (FEn) occurs during a reception operation, the receive operation continues until stop bit is received, and after reception is completed, either a reception completion interrupt (INTSRn) or a reception error interrupt (INTSREn) is generated according to the ISRM bit setting in the ASIMn register.

If the RXEn bit is reset (0) during a receive operation, the receive operation is immediately stopped. The contents of receive buffer register n (RXBn) and of the asynchronous serial interface status register (ASISn) at this time do not change, and no reception completion interrupt (INTSRn) or reception error interrupt (INTSREn) is generated.

No reception completion interrupt is generated when RXEn = 0 (reception is disabled).

Figure 13-8. Asynchronous Serial Interface Reception Completion Interrupt Timing



Cautions 1. Be sure to read receive buffer register n (RXBn) even when a reception error occurs. If RXBn is not read, an overrun error will occur at the next data reception and the reception error status will continue infinitely.

**2. Reception is always performed assuming a stop bit length of 1.
A second stop bit is ignored.**

(5) Reception error

The three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. As a result of data reception, the various flags of the ASISn register are set (1), and a reception error interrupt (INTSREn) or a reception completion interrupt (INTSRn) is generated at the same time. The ISRMn bit of the ASIMn register specifies whether INTSREn or INTSRn is generated.

The type of error that occurred during reception can be detected by reading the contents of the ASISn register during the INTSREn or INTSRn interrupt servicing.

The contents of the ASISn register are reset (0) by reading the ASISn register.

Table 13-2. Reception Error Causes

Error Flag	Reception Error	Cause
PEn	Parity error	The parity specification during transmission did not match the parity of the reception data
FEn	Framing error	No stop bit was detected
OVEn	Overrun error	The reception of the next data was completed before data was read from receive buffer register n (RXBn)

(a) Separation of reception error interrupt

A reception error interrupt can be separated from the INTSRn interrupt and generated as the INTSREn interrupt by clearing the ISRMn bit of the ASIMn register to 0.

Figure 13-9. When Reception Error Interrupt Is Separated from Reception Completion Interrupt (INTSRn) (ISRMn Bit = 0)

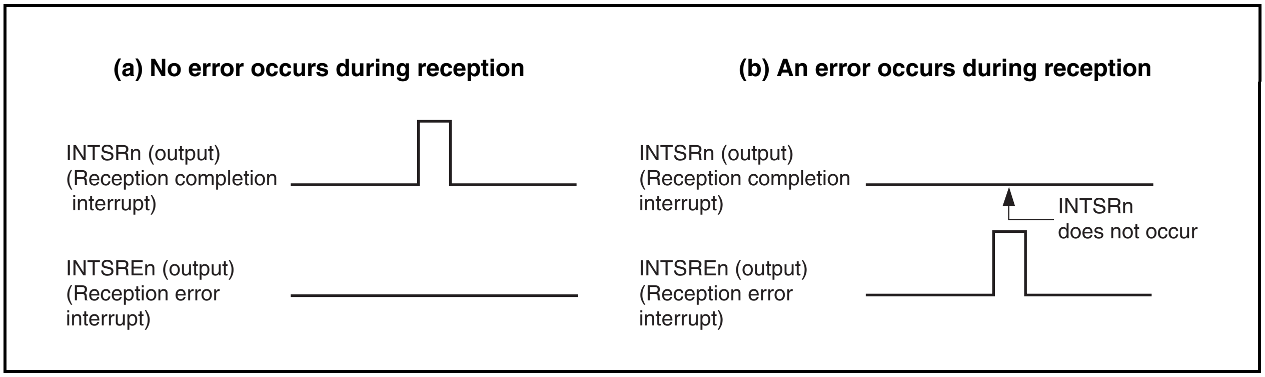
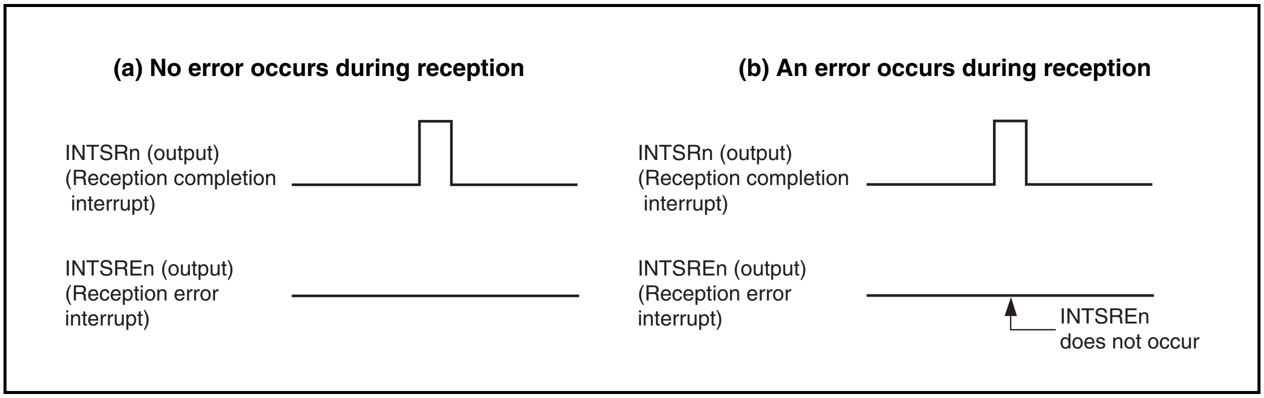

(a) No error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSREn (output)
(Reception error interrupt)
(b) An error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSRn does not occur
INTSREn (output)
(Reception error interrupt)

Figure 13-10. When Reception Error Interrupt Is Included in Reception Completion Interrupt (INTSRn) (ISRMn Bit = 1)


(a) No error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSREn (output)
(Reception error interrupt)
(b) An error occurs during reception
INTSRn (output)
(Reception completion interrupt)
INTSREn (output)
(Reception error interrupt)
INTSREn does not occur

(6) Parity types and corresponding operation

A parity bit is used to detect a bit error in communication data. Normally, the same type of parity bit is used on the transmission and reception sides.

(a) Even parity

(i) During transmission

The parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is even. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 1
- If the number of bits with the value “1” within the transmit data is even: 0

(ii) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is odd.

(b) Odd parity

(i) During transmission

In contrast to even parity, the parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is odd. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 0
- If the number of bits with the value “1” within the transmit data is even: 1

(ii) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is even.

(c) 0 parity

During transmission the parity bit is set to “0” regardless of the transmit data.

During reception, no parity bit check is performed. Therefore, no parity error is generated regardless of whether the parity bit is “0” or “1”.

(d) No parity

No parity bit is added to the transmit data.

During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

(7) Receive data noise filter

The RXDn signal is sampled at the rising edge of the prescaler output base clock (Clock). If the same sampling value is obtained twice, the match detector output changes, and this output is sampled as input data. Therefore, data not exceeding one clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 13-12**). Refer to **13.6 (1) (a) Base clock (Clock)** regarding the base clock.

Also, since the circuit is configured as shown in Figure 13-11, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 13-11. Noise Filter Circuit

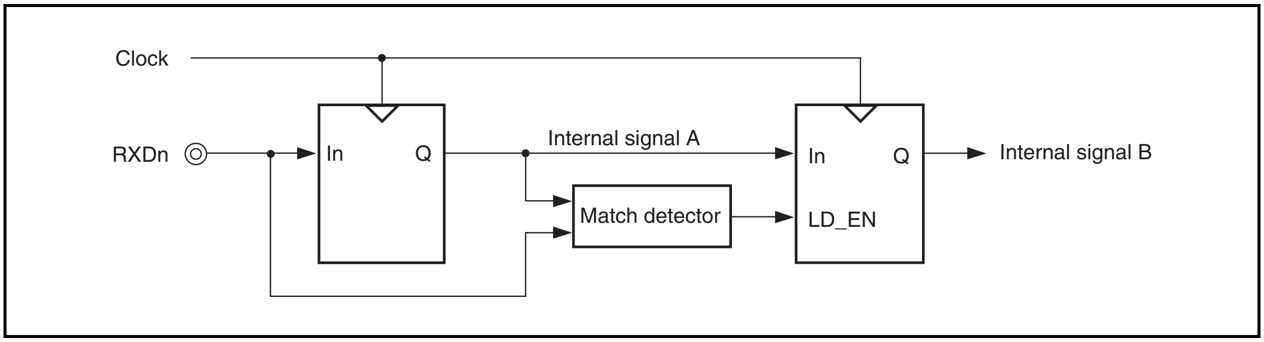
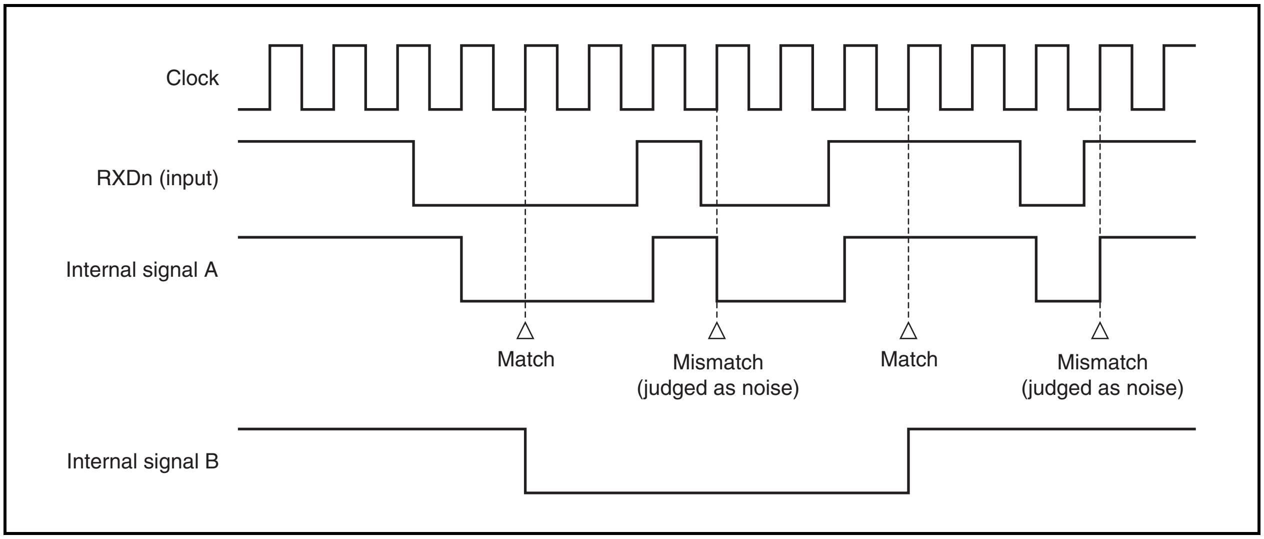


Figure 13-12. Timing of RXDn Signal Judged as Noise



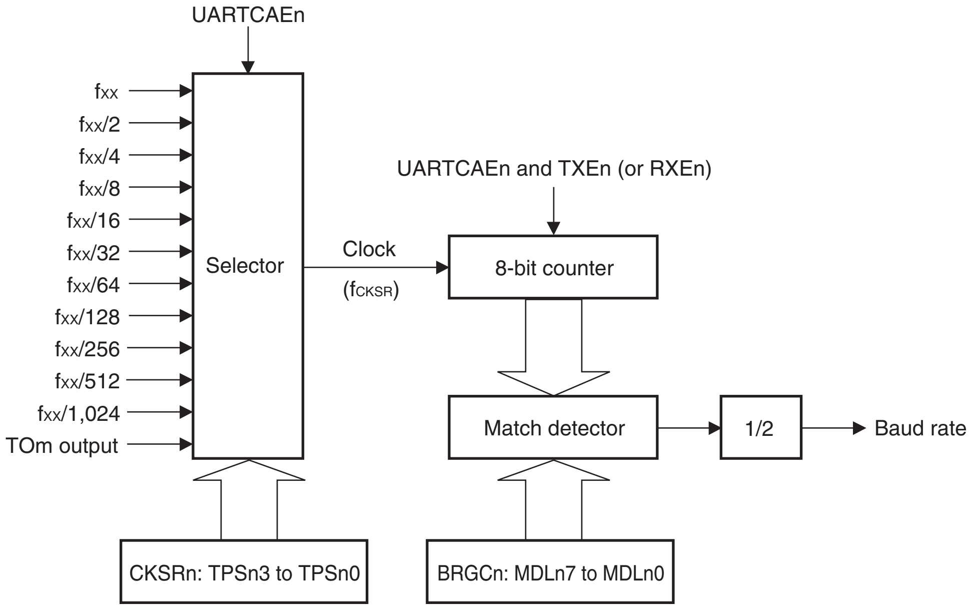
13.6 Dedicated Baud Rate Generator n (BRGn)

A dedicated baud rate generator, which consists of a source clock selector and an 8-bit programmable counter, generates serial clocks during transmission/reception by UARTn. The dedicated baud rate generator output can be selected as the serial clock for each channel.

Separate 8-bit counters exist for transmission and for reception.

(1) Baud rate generator n (BRGn) configuration

Figure 13-13. Configuration of Baud Rate Generator n (BRGn)



Remark fXX: Main clock
n = 0, 1
m = 4, 5 (m = 4 when n = 0, m = 5 when n = 1)

(a) Base clock (Clock)

When the UARTCAEn bit = 1 in the ASIMn register, the clock selected according to the TPSn3 to TPSn0 bits of the CKSRn register is supplied to the transmission/reception unit. This clock is called the base clock (Clock), and its frequency is referred to as f_{CKSR}. When UARTCAEn = 0, Clock is fixed to low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the CKSRn and BRGCn registers.
The base clock to the 8-bit counter is selected by the TPSn3 to TPSn0 bits of the CKSRn register.
The 8-bit counter divisor value can be set by the MDLn7 to MDLn0 bits of the BRGCn register.

(a) Clock select register n (CKSRn)

The CKSRn register is an 8-bit register for selecting the basic block using the TPSn3 to TPSn0 bits. The clock selected by the TPSn3 to TPSn0 bits becomes the base clock (Clock) of the transmission/reception module. Its frequency is referred to as f_{CKSR}.
This register can be read or written in 8-bit units.
This register is cleared to 00H after reset.

Caution Set the UARTCAEn bit of the ASIMn register to 0 before rewriting the TPSn3 to TPSn0 bits.

After reset: 00H   R/W   Address: CKSR0 FFFFFA06H, CKSR1 FFFFFA16H

	7	6	5	4	3	2	1	0
CKSRn	0	0	0	0	TPSn3	TPSn2	TPSn1	TPSn0

TPSn3	TPSn2	TPSn1	TPSn0	Receive operation
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	0	0	1	$f_{XX}/512$
1	0	1	0	$f_{XX}/1,024$
1	0	1	1	TOm output
Other than above				Setting prohibited

Remark n = 0, 1
m = 4, 5 (m = 4 when n = 0, m = 5 when n = 1)

(b) Baud rate generator control register n (BRGCn)

The BRGCn register is an 8-bit register that controls the baud rate (serial transfer speed) of UARTn.

This register can be read or written in 8-bit units.

This register is set to FFH after reset.

Caution If the MDLn7 to MDLn0 bits are to be overwritten, the TXEn and RXEn bits should be set to 0 in the ASIMn register first.

After reset: FFH R/W Address: BRGC0 FFFFFA07H, BRGC1 FFFFFA17H

	7	6	5	4	3	2	1	0
BRGCn (n = 0, 1)	MDLn7	MDLn6	MDLn5	MDLn4	MDLn3	MDLn2	MDLn1	MDLn0

MD Ln7	MD Ln6	MD Ln5	MD Ln4	MD Ln3	MD Ln2	MD Ln1	MD Ln0	Division value (k)	Serial clock
0	0	0	0	0	×	×	×	–	Setting prohibited
0	0	0	0	1	0	0	0	8	f_{CKSR}/8
0	0	0	0	1	0	0	1	9	f_{CKSR}/9
0	0	0	0	1	0	1	0	10	f_{CKSR}/10
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	0	1	0	250	f_{CKSR}/250
1	1	1	1	1	0	1	1	251	f_{CKSR}/251
1	1	1	1	1	1	0	0	252	f_{CKSR}/252
1	1	1	1	1	1	0	1	253	f_{CKSR}/253
1	1	1	1	1	1	1	0	254	f_{CKSR}/254
1	1	1	1	1	1	1	1	255	f_{CKSR}/255

Remarks 1. f_{CKSR}: Frequency [Hz] of base clock (Clock) selected by TPSn3 to TPSn0 bits of CKSRn register

2. k: Value set by MDLn7 to MDLn0 bits (k = 8, 9, 10, ..., 255)

3. The baud rate is the output clock for the 8-bit counter divided by 2

4. ×: Don’t care

(c) Baud rate

The baud rate is the value obtained by the following formula.

$$\text{Baud rate} = \frac{f_{\text{CKSR}}}{2 \times k} \text{ [bps]}$$

f_{CKSR} = Frequency [Hz] of base clock (Clock) selected by TPSn3 to TPSn0 bits of CKSRn register.
k = Value set by MDLn7 to MDLn0 bits of BRGCn register (k = 8, 9, 10, ..., 255)

(d) Baud rate error

The baud rate error is obtained by the following formula.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. Make sure that the baud rate error during transmission does not exceed the allowable error of the reception destination.

2. Make sure that the baud rate error during reception is within the allowable baud rate range described in (4) Allowable baud rate range during reception.

Example Base clock frequency = 20 MHz = 20,000,000 Hz
Setting of MDLn7 to MDLn0 bits in BRGC0 register = 01000001B (k = 65)
Target baud rate = 153,600 bps

Baud rate = 20M/(2 × 65)
= 20,000,000/(2 × 65) = 153,846 [bps]

Error = (153,846/153,600 − 1) × 100
= 0.160 [%]

(3) Baud rate setting example

Table 13-3. Baud Rate Generator Setting Data

Baud Rate [bps]	f_{XX} = 20 MHz			f_{XX} = 17 MHz			f_{XX} = 13.5 MHz			f_{XX} = 8 MHz		
	f_{CKSR}	k	ERR	f_{CKSR}	k	ERR	f_{CKSR}	k	ERR	f_{CKSR}	k	ERR
300	f_{XX}/512	65	0.16	f_{XX}/128	221	0.16	f_{XX}/1024	22	–0.12	f_{XX}/1024	13	0.16
600	f_{XX}/256	65	0.16	f_{XX}/64	221	0.16	f_{XX}/1024	11	–0.12	f_{XX}/512	13	0.16
1,200	f_{XX}/128	65	0.16	f_{XX}/32	221	0.16	f_{XX}/512	11	–0.12	f_{XX}/256	13	0.16
2,400	f_{XX}/64	65	0.16	f_{XX}/16	221	0.16	f_{XX}/256	11	–0.12	f_{XX}/128	13	0.16
4,800	f_{XX}/32	65	0.16	f_{XX}/8	221	0.16	f_{XX}/128	11	–0.12	f_{XX}/64	13	0.16
9,600	f_{XX}/16	65	0.16	f_{XX}/4	221	0.16	f_{XX}/64	11	–0.12	f_{XX}/32	13	0.16
19,200	f_{XX}/8	65	0.16	f_{XX}/2	221	0.16	f_{XX}/32	11	–0.12	f_{XX}/16	13	0.16
31,250	f_{XX}/32	10	0.00	f_{XX}/16	17	0.00	f_{XX}/8	27	0.00	f_{XX}/16	8	0.00
38,400	f_{XX}/4	65	0.16	f_{XX}	221	0.16	f_{XX}/16	11	–0.12	f_{XX}/8	13	0.16
76,800	f_{XX}/2	65	0.16	f_{XX}	111	–0.29	f_{XX}/8	11	–0.12	f_{XX}/4	13	0.16
153,600	f_{XX}	65	0.16	f_{XX}	55	0.62	f_{XX}/4	11	–0.12	f_{XX}/2	13	0.16
312,500	f_{XX}/4	8	0.00	f_{XX}	27	0.74	f_{XX}/2	11	–1.82	f_{XX}	13	–1.54

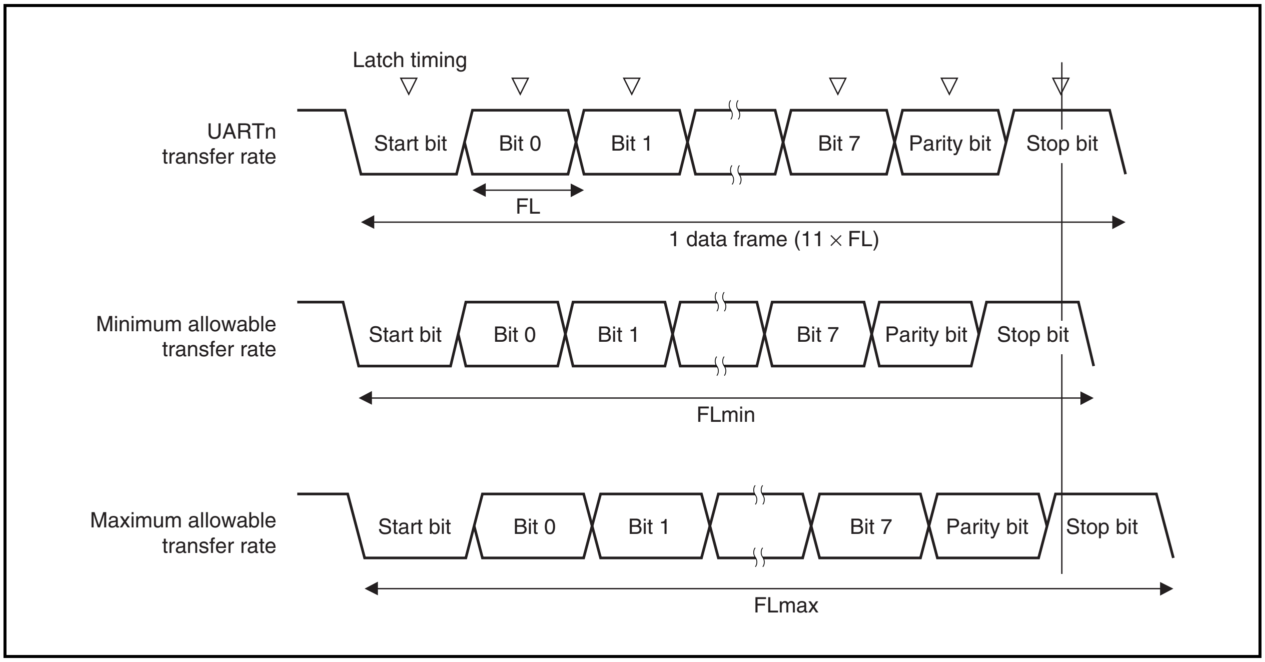
Remark f_{XX}: Main clock frequency
f_{CKSR}: Base clock frequency
k: Setting values of MDLn7 to MDLn0 bits in BRGCn register
ERR: Baud rate error [%]

(4) Allowable baud rate range during reception

The degree to which a discrepancy from the transmission destination's baud rate is allowed during reception is shown below.

Caution The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

Figure 13-14. Allowable Baud Rate Range During Reception



As shown in Figure 13-13, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the BRGCn register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.

If this is applied to 11-bit reception, the following is theoretically true.

$$FL = (Brate)^{-1}$$

Brate:	UARTn baud rate
k:	BRGCn register setting value
FL:	1-bit data length

When the latch timing margin is 2 base clocks (Clock), the minimum allowable transfer rate (FLmin) is as follows.

$$FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the transfer destination's maximum receivable baud rate (BRmax) is as follows.

$$\boxed{\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\ \text{Brate}}$$

Similarly, the maximum allowable transfer rate (FLmax) can be obtained as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\ \text{FL} \times 11$$

Therefore, the transfer destination's minimum receivable baud rate (BRmin) is as follows.

$$\boxed{\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\ \text{Brate}}$$

The allowable baud rate error of UARTn and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 13-4. Maximum and Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

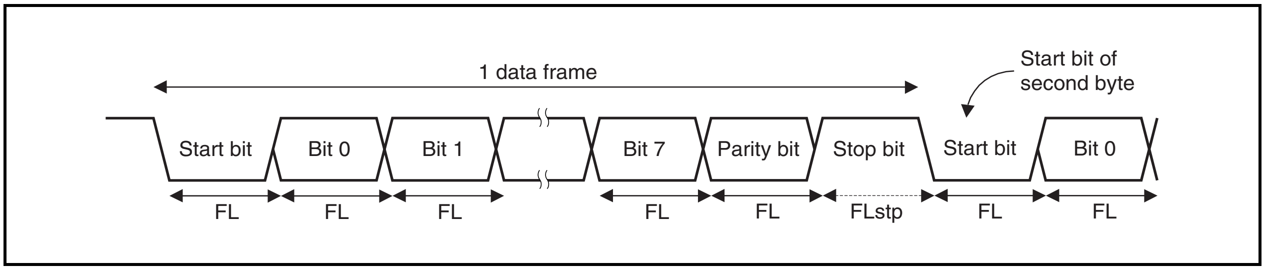
Remarks **1.** The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: BRGCn setting value

(5) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks of the base clock (Clock) longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

Figure 13-15. Transfer Rate During Continuous Transmission



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the base clock frequency by f_{CKSR} yields the following equation.

$$FLstp = FL + 2/f_{CKSR}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/f_{CKSR}$$

13.7 Cautions

★

Cautions to be observed when using UARTn are shown below.

(1) When the supply of clocks to UARTn is stopped (for example, in IDLE or STOP mode), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXDn pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting UARTCAEn = 0, RXEn = 0, and TXEn = 0 in the ASIMn register.

(2) When the UARTCAEn bit is set to 0, the UARTn unit is asynchronously reset. The output of the TXDn pin goes to high level.
To operate the UARTn unit, set the UARTCAEn bit to 1, and then set the other bits (TXEn bit = 1, RXEn bit = 1).
To stop the UARTn unit, clear the TXEn and RXEn bits to 0, and then clear the UARTCAEn bit to 0.

(3) Do not change the values of the following control registers when the TXEn bit or RXEn bit is 1.

- PSn1, PSn0, CLn, SLn, and ISRMn bits of ASIMn register
- BRGCn register

Before changing the values of the above registers, clear the TXEn or RXEn bit to 0.
The operation when the above values are changed with the TXEn or RXEn bit set to 1 is prohibited.

(4) To initialize the transmission or reception status, time corresponding to two cycles of the source clock (f_{CKSR}) is required after clearing the TXEn bit or RXEn bit to 0.

(5) To successively transmit data, confirm the value of the TXBFn bit and then write data to the TXBn register. Writing data to the TXBn register is prohibited when the TXBFn bit is 1 (write disabled state).

(6) Clear the UARTCAEn bit to 0 before rewriting the CKSRn register.

(7) Always read the RXBn register, even when a reception error has occurred.
Unless read, the reception error status (OVEn bit = 1) continues indefinitely.

(8) UARTn has a 2-stage buffer configuration consisting of transmit buffer register n (TXBn) and the transmit shift register, and has status flags (the TXBFn and TXSFn bits of the ASIFn register) that indicate the status of each buffer. If the TXBFn and TXSFn bits are read in continuous transmission, the value changes from 10 to 01, but since this change timing is in the period in which data is shifted from TXBn to the transmit shift register, 11 or 00 may be read, depending on the timing. Thus, read only the TXBFn bit during continuous transmission.

14.1 Features

- Transfer rate: Master mode: Maximum 5 Mbps
  Slave mode: Maximum 5 Mbps
- Half-duplex communications
- Master mode and slave mode can be selected
- Transmission data length: 8 bits
- Transfer data direction can be switched between MSB first and LSB first
- Eight clock signals can be selected (7 master clocks and 1 slave clock)
- 3-wire method
  - SOn: Serial data output
  - SIn: Serial data input
  - $\overline{\text{SCKn}}$: Serial clock input/output
- Interrupt sources: 1 type
  - Transmission/reception completion interrupt (INTCSIn)
- Transmission/reception mode or reception-only mode can be specified
- On-chip transmit buffer (SOTBn)

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

14.1.1 Switching modes between CSI0 and I^{2}C

CSI0 and I^{2}C of the V850ES/SA2 and V850ES/SA3 share pins, and therefore these interfaces cannot be used at the same time. Select CSI0 or I^{2}C in advance by using port mode control register 4 (PMC4) and port function control register 4 (PFC4) (refer to **4.3.4 Port 4**).

Cautions 1. CSI0 or I^{2}C transmission/reception operations are not guaranteed if the mode is changed during transmission or reception. Be sure to disable the operation of the unit that is not used.

2. I^{2}C: μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Figure 14-1. Selecting CSI0 or I^{2}C Mode

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	PFC46	0	PFC44	0	PFC42	PFC41	0

PMC4n	PFC4n	Operation mode
0	×	Port I/O mode
1	0	CSI0 mode
1	1	I^{2}C mode

Remarks 1. n = 1, 2

2. × = Don't care

14.1.2 Switching modes between CSI1 and UART0

CSI1 and UART0 of the V850ES/SA2 and V850ES/SA3 share pins, and therefore these interfaces cannot be used at the same time. Select CSI1 or UART0 in advance by using port mode control register 3 (PMC3) and port function control register 3 (PFC3) (refer to **4.3.3 Port 3**).

Caution CSI1 or UART0 transmission/reception operations are not guaranteed if the mode is changed during transmission or reception. Be sure to disable the operation of the unit that is not used.

Figure 14-2. Selecting CSI1 or UART0 Mode

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	0	0	0	0	0	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	0	0	0	0	PFC31	PFC30

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	CSI1 mode
1	1	UART0 mode

Remarks 1. n = 0, 1

2. × = Don’t care

14.2 Configuration

CSIn is controlled by the clocked serial interface mode register (CSIMn). Transmit/receive data can be written to or read from the SIOn register.

(1) Clocked serial interface mode register n (CSIMn)

The CSIMn register is an 8-bit register for specifying the operation of CSIn.

(2) Clocked serial interface clock selection register n (CSICn)

The CSICn register is an 8-bit register for controlling the transmit operation of CSIn.

(3) Serial I/O shift register n (SIOn)

The SIOn register is an 8-bit register for converting between serial data and parallel data. SIOn is used for both transmission and reception.

Data is shifted in (reception) or shifted out (transmission) beginning at either the MSB side or the LSB side.

Actual transmit/receive operations are controlled by reading or writing SIOn.

(4) Clocked serial interface transmit buffer register n (SOTBn)

The SOTBn register is an 8-bit buffer register for storing transmit data.

(5) Selector

The selector selects the serial clock to be used.

(6) Serial clock controller

The serial clock controller controls the supply of serial clocks to the shift register. When an internal clock is used, it also controls the clocks that are output to the $\overline{\text{SCKn}}$ pin.

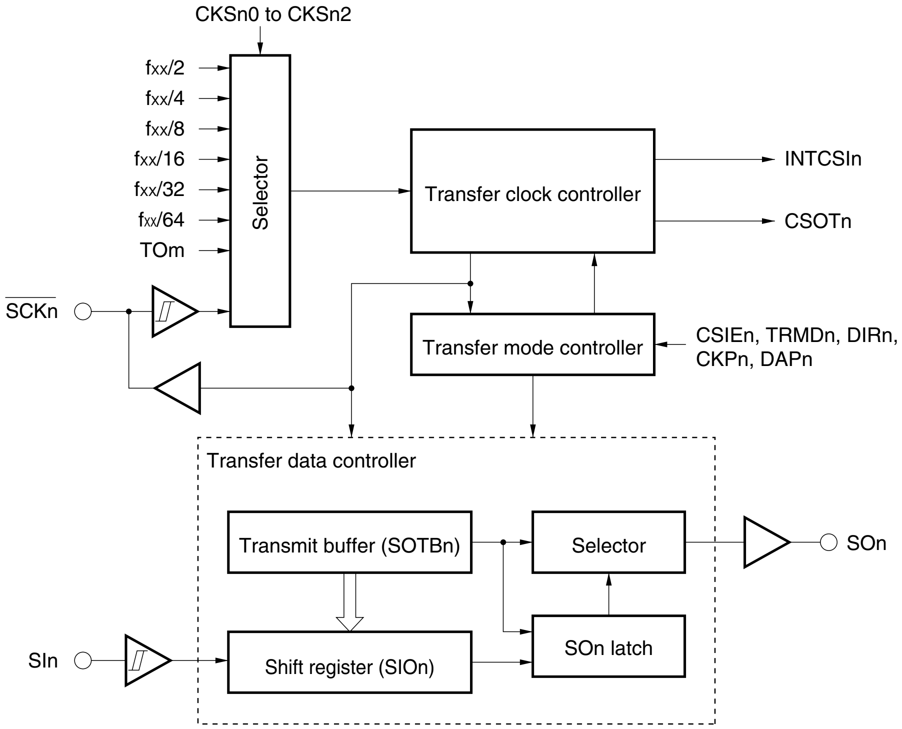
(7) Serial clock counter

The serial clock counter counts serial clocks that are output or input during transmit and receive operations and checks that 8-bit data has been transmitted or received.

(8) Interrupt controller

The interrupt controller controls whether or not an interrupt request is generated when the serial clock counter has counted eight serial clocks.

Figure 14-3. Clocked Serial Interface Block Diagram



Remarks 1. m = 2 when n = 0
m = 3 when n = 1
m = 4 when n = 2
m = 5 when n = 3
m = 5 when n = 4 (V850ES/SA3 only)

2. fxx: Main clock frequency

14.3 Control Registers

(1) Clocked serial interface mode register n (CSIMn)

The CSIn register controls the operation of CSIn.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

Caution To use CSIn, be sure to set the external pins related to the CSIn function to control mode and set the CSICn register. Then set the CSIEn bit to 1 before setting the other bits.

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

After reset: 00H R/W Address: CSIM0 FFFFFD00H, CSIM1 FFFFFD10H, CSIM2 FFFFFD20H
CSIM3 FFFFFD30H, CSIM4[Note 1] FFFFFD40H

	<7>	<6>	5	<4>	3	2	1	<0>
CSIMn	CSIEn	TRMDn	0	DIRn	0	0	0	CSOTn[Note 2]

CSIEn	CSIn operation enable/disable specification
0	CSIn operation is disabled (SOn = low level, $\overline{\text{SCKn}}$ = high level)
1	CSIn operation is enabled

- If CSIEn is set to 0, the CSIn unit can be reset asynchronously.
- If CSIEn = 0, the CSIn unit is in a reset state. Therefore, to operate CSIn, CSIEn must be set to 1.
- If the CSIEn bit is changed from 1 to 0, all registers of the CSIn unit are initialized. To set CSIEn to 1 again, the registers of the CSIn unit must be set again.

TRMDn	Transmission mode specification
0	Reception-only mode
1	Transmission/reception mode

- If TRMDn = 0, reception-only transfers are performed. In addition, the SOn pin output is fixed at low level. Data reception is started by reading the SIOn register. If TRMDn = 1, transmission/reception is started by writing data to the SOTBn register.
- The TRMDn bit can be overwritten only when CSOTn = 0.

DIRn	Transfer direction mode (MSB/LSB specification)
0	MSB first
1	LSB first

- The DIRn bit can be overwritten only when CSOTn = 0.

CSOTn[Note 2]	Transfer status display flag
0	Idle status
1	Transfer execution status

- This flag is used to judge whether writing to the shift register (SIOn) is enabled or not when starting serial data transmission in transmission/reception mode (TRMDn = 1)
- The CSOTn bit is reset when the CSIE bit is cleared (0).

Notes 1. CSIM4: V850ES/SA3 only

2. The CSOTn bit is read-only.

Caution Be sure to set bits 5 and 3 to 1 to 0.

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

(2) Clocked serial interface clock selection register n (CSICn)

The CSICn register is an 8-bit register that controls the transmit operation of CSIn.

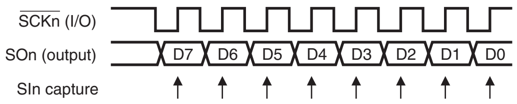
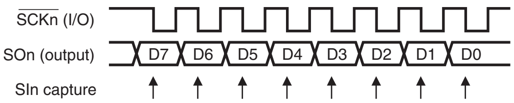
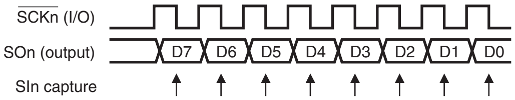
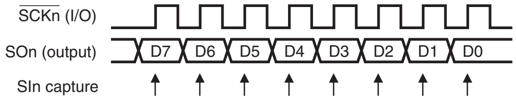
This register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

Caution The CSICn register can only be overwritten after CSIEn is cleared to 0 in the CSIMn register.

After reset: 00H R/W Address: CSIC0 FFFFFD01H, CSIC1 FFFFFD11H, CSIC2 FFFFFD21H
CSIC3 FFFFFD31H, CSIC4[Note 1] FFFFFD41H

	7	6	5	4	3	2	1	0
CSICn	0	0	0	CKPn	DAPn	CKSn2	CKSn1	CKSn0

CKPn	DAPn	Specification of data transmission/reception timing for SCKn
0	0	 $\overline{\text{SCKn}}$ (I/O) SOn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIn capture
0	1	 $\overline{\text{SCKn}}$ (I/O) SOn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIn capture
1	0	 $\overline{\text{SCKn}}$ (I/O) SOn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIn capture
1	1	 $\overline{\text{SCKn}}$ (I/O) SOn (output) D7 D6 D5 D4 D3 D2 D1 D0 SIn capture

CKSn2	CKSn1	CKSn0	Input clock	Mode
0	0	0	fxx/2	Master mode
0	0	1	fxx/4	Master mode
0	1	0	fxx/8	Master mode
0	1	1	fxx/16	Master mode
1	0	0	fxx/32	Master mode
1	0	1	fxx/64	Master mode
1	1	0	TOm output[Note 2]	Master mode
1	1	1	External clock ($\overline{\text{SCKn}}$)	Slave mode

Notes 1. CSIC4: V850ES/SA3 only

2. m = 2 when n = 0
m = 3 when n = 1
m = 4 when n = 2
m = 5 when n = 3
m = 5 when n = 4 (V850ES/SA3 only)

Caution Set the input clock to 5 MHz or lower.

(a) Transfer rate selection example

CKSn2	CKSn1	CKSn0	Baud Rate (bps)					
			20 MHz Operation	17 MHz Operation	13.5 MHz Operation	10 MHz Operation	8 MHz Operation	4 MHz Operation
0	0	0	Setting prohibited	Setting prohibited	Setting prohibited	5,000,000	4,000,000	2,000,000
0	0	1	5,000,000	4,250,000	3,375,000	2,500,000	2,000,000	1,000,000
0	1	0	2,500,000	2,125,000	1,687,500	1,250,000	1,000,000	500,000
0	1	1	1,250,000	1,062,500	843,750	625,000	500,000	250,000
1	0	0	625,000	531,250	421,875	312,500	250,000	125,000
1	0	1	312,500	265,625	210,938	156,250	125,000	62,500

(3) Serial I/O shift register n (SIOn)

The SIOn register is an 8-bit shift register that converts parallel data to serial data. If TRMDn = 0 in the CSIMn register, the transfer is started by reading SIOn.

Except when a reset is input, the SIOn register becomes 00H even when the CSIEn bit of the CSIMn register is cleared (0).

SIOn shifts data in (reception) or shifts data out (transmission) beginning at the MSB or the LSB side.

This register is read-only, in 8-bit units.

Caution The SIOn register can be accessed only when the system is in an idle state (CSOTn bit = 0 in the CSIMn register).

After reset: 00H R Address: SIO0: FFFFFD02H, SIO1: FFFFFD12H, SIO2: FFFFFD22H
SIO3: FFFFFD32H, SIO4[Note]: FFFFFD42H

	7	6	5	4	3	2	1	0
SIOn	SIOn7	SIOn6	SIOn5	SIOn4	SIOn3	SIOn2	SIOn1	SIOn0

Note SIO4: V850ES/SA3 only

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

(4) Receive-only serial I/O shift register n (SIOEn)

The SIOEn register is an 8-bit shift register that converts parallel data into serial data. A receive operation does not start even if the SIOEn register is read while the TRMDn bit of the CSIMn register is 0. Therefore this register is used to read the value of the SIOn register (receive data) without starting a receive operation.

SIOEn shifts data in (reception) beginning at the MSB or the LSB side.

Except when a reset is input, the SIOEn register becomes 00H even when the CSIEn bit of the CSIMn register is cleared (0).

This register is read-only, in 8-bit units.

Caution The SIOEn register can be accessed only when the system is in an idle state (CSOTn bit = 0 in the CSIMn register).

After reset: 00H R Address: SIOE0: FFFFFD03H, SIOE1: FFFFFD13H, SIOE2: FFFFFD23H
SIOE3: FFFFFD33H, SIOE4[Note]: FFFFFD43H

	7	6	5	4	3	2	1	0
SIOEn	SIOEn7	SIOEn6	SIOEn5	SIOEn4	SIOEn3	SIOEn2	SIOEn1	SIOEn0

Note SIOE4: V850ES/SA3 only

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

(5) Clocked serial interface transmit buffer register n (SOTBn)

The SOTBn register is an 8-bit buffer register for storing transmit data.

If transmission/reception mode is set (TRMDn = 1 in the CSIMn register), a transmit operation is started by writing data to the SOTBn register.

This register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

Caution The SOTBn register can be accessed only when the system is in an idle state (CSOTn bit = 0 in the CSIMn register).

After reset: 00H R/W Address: SOTB0: FFFFFD04H, SOTB1: FFFFFD14H, SOTB2: FFFFFD24H
SOTB3: FFFFFD34H, SOTB4[Note]: FFFFFD44H

	7	6	5	4	3	2	1	0
SOTBn	SOTBn7	SOTBn6	SOTBn5	SOTBn4	SOTBn3	SOTBn2	SOTBn1	SOTBn0

Note SOTB4: V850ES/SA3 only

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

14.4 Operation

(1) Transfer mode

CSIn transmits and receives data using three lines: 1 clock line and 2 data lines.

In reception-only mode (TRMDn = 0 in the CSIMn register), the transfer is started by reading the SIOn register. To read the value of the SIOn register without starting reception, read the SIOEn register.

In transmission/reception mode (TRMDn = 1 in the CSIMn register), the transfer is started by writing data to the SOTBn register.

When an 8-bit transfer of CSIn ends, the CSOTn bit of the CSIMn register becomes 0, and transfer stops automatically. Also, when the transfer ends, a transmission/reception completion interrupt (INTCSIn) is generated.

Cautions 1. When CSOTn bit = 1 in the CSIMn register, the control registers and data registers should not be accessed.

2. If transmit data is written to the SOTBn register and the TRMDn bit of the CSIMn register is changed from 0 to 1, serial transfer is not performed.

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

(2) Serial clock

(a) When internal clock is selected as the serial clock

If reception or transmission is started, a serial clock is output from the $\overline{\text{SCKn}}$ pin, and the data of the SIn pin is taken into the SIOn register sequentially or data is output to the SOn pin sequentially from the SIOn register when the data is synchronized with the serial clock in accordance with the setting of the CKPn and DAPn bits of the CSICn register.

(b) When external clock is selected as the serial clock

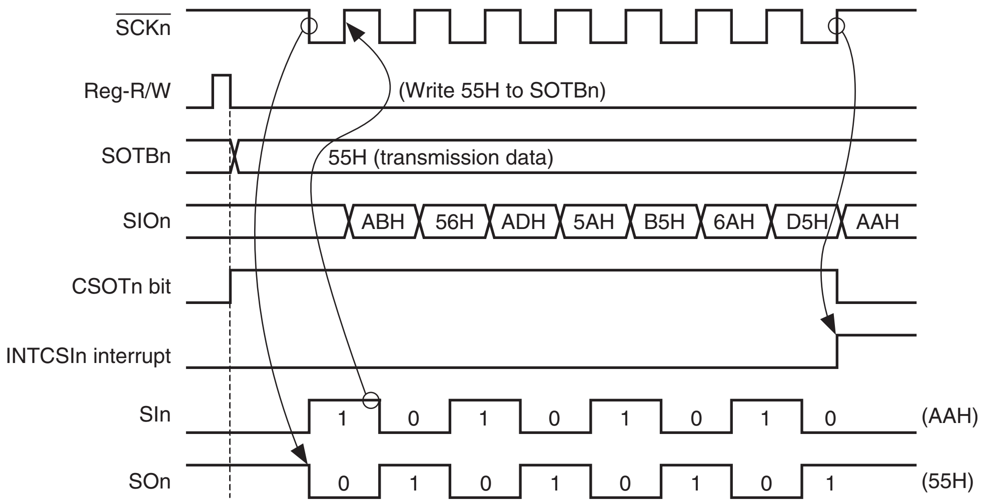
If reception or transmission is started, the data of the SIn pin is taken into the SIOn register sequentially or output to the SOn pin sequentially in synchronization with the serial clock that has been input to the $\overline{\text{SCKn}}$ pin following transmission/reception startup in accordance with the setting of the CKPn and DAPn bits of the CSICn register.

If serial clock is input to the $\overline{\text{SCKn}}$ pin when neither reception nor transmission is started, a shift operation will not be executed.

Remark n = 0 to 3 (V850ES/SA2)
n = 0 to 4 (V850ES/SA3)

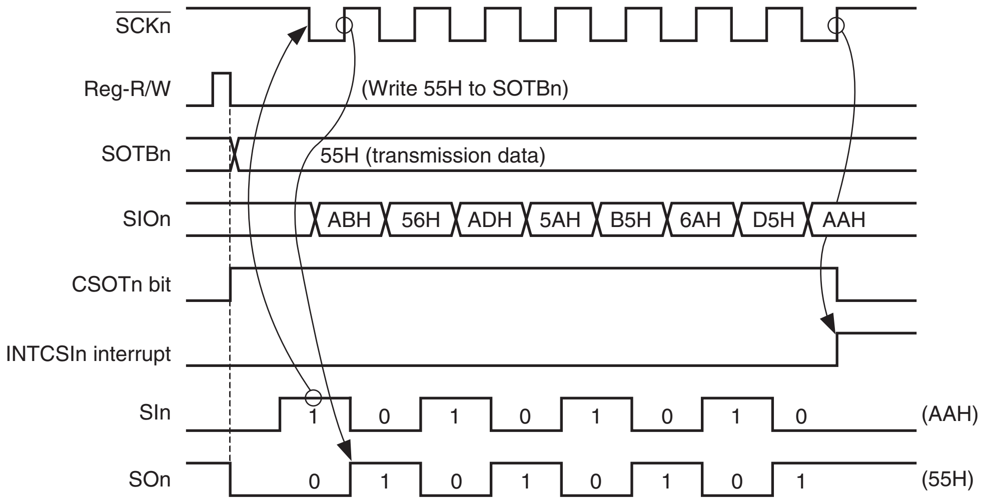
Figure 14-4. Transfer Timing

(a) When TRMDn = 1, DIRn = 0, CKPn = 0, and DAPn = 0



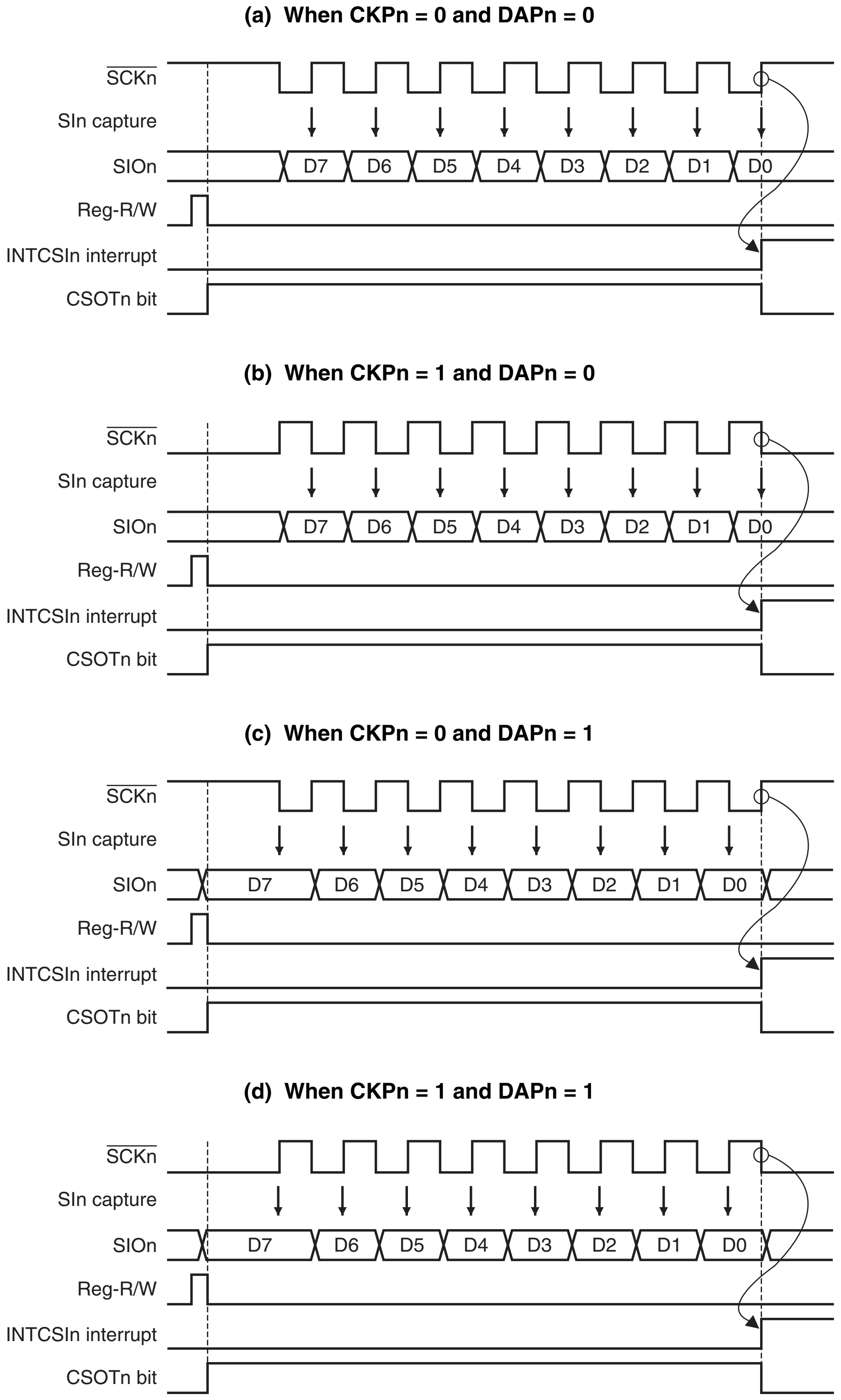
Remark n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

(b) When TRMDn = 1, DIRn = 0, CKPn = 0, and DAPn = 1



Remark n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

Figure 14-5. Clock Timing



(a) When CKPn = 0 and DAPn = 0

(b) When CKPn = 1 and DAPn = 0

(c) When CKPn = 0 and DAPn = 1

(d) When CKPn = 1 and DAPn = 1

Remark n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

14.5 Output Pins

(1) $\overline{\text{SCKn}}$ pin

When CSIn operation is disabled (CSIEn = 0), the $\overline{\text{SCKn}}$ pin output state is as follows.

CKPn	$\overline{\text{SCKn}}$ Pin Output
0	Fixed to high level
1	Fixed to low level

Remarks 1. When the CKPn bit is overwritten, the $\overline{\text{SCKn}}$ pin output changes.

2. n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

(2) SOn pin

When CSIn operation is disabled (CSIEn = 0), the SOn pin output state is as follows.

TRMDn	DAPn	DIRn	SOn Pin Output
0	×	×	Fixed to low level
1	0	×	SOn latch value (low level)
	1	0	SOTBn7 value
		1	SOTBn0 value

Remarks 1. If any of the TRMDn, DAPn, and DIRn bits is overwritten, the SOn pin output changes.

2. n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

3. ×: Don’t care

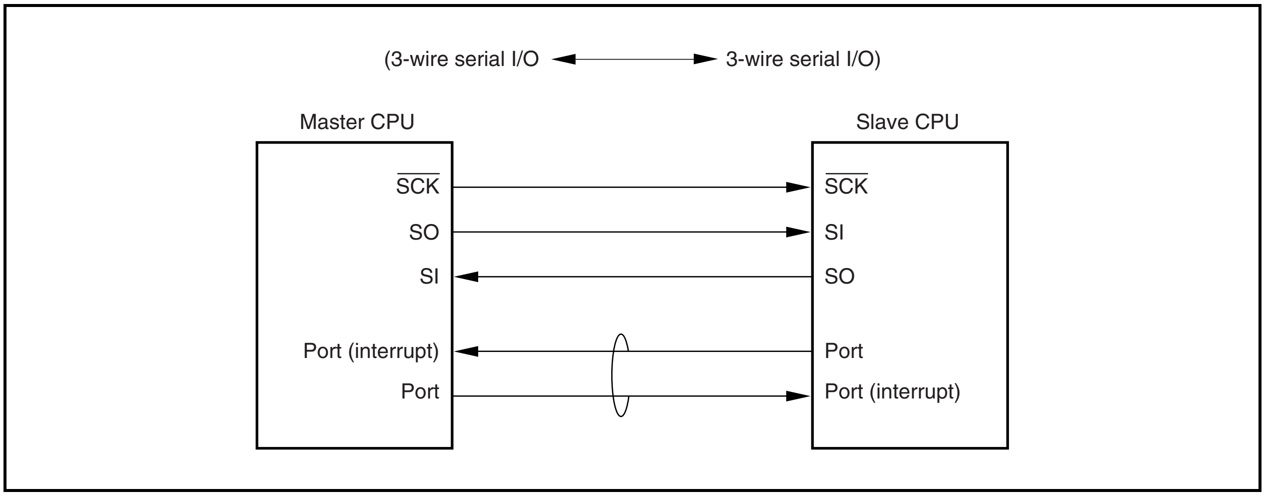
14.6 System Configuration Example

CSIn performs 8-bit length data transfer using three signal lines: a serial clock ($\overline{\text{SCKn}}$), serial input (SIn), and serial output (SOn). This is effective when connecting peripheral I/O that incorporate a conventional clocked serial interface, or a display controller to the V850ES/SA2 or V850ES/SA3 (n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)).

When connecting the V850ES/SA2 or V850ES/SA3 to several devices, lines for handshake are required.

Since the first communication bit can be selected as MSB or LSB, communication with various devices can be achieved.

Figure 14-6. System Configuration Example of CSI



CHAPTER 15 I²C BUS

To use the I²C bus function, set the P41/SO0/SDA and P42/$\overline{\text{SCK0}}$/SCL pins to N-ch open drain output.

15.1 Features

I²C has the following two modes.

- Operation stopped mode
- I²C (Inter IC) bus mode (multiple masters supported)

(1) Operation stopped mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I²C bus mode (multiple masters supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock line (SCL) and a serial data bus line (SDA).

This mode complies with the I²C bus format and the master device can output “start condition”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received data by hardware. This function can simplify the part of application program that controls the I²C bus.

Since SCL and SDA are open-drain outputs, the I²C bus requires pull-up resistors for the serial clock line and the serial data bus line.

Caution Follow the procedure below to set N-ch open drain.

<1> Set the P4n bit to 1.
<2> Set the PF4n bit to 1.
<3> Set the IICE bit to 1.
<4> Set the PFC4n bit to 1.
<5> Set the PMC4n bit to 1.

15.1.1 Switching modes between I^{2}C and CSI0

CSI0 and I^{2}C of the V850ES/SA2 and V850ES/SA3 share pins, and therefore these interfaces cannot be used at the same time. Select CSI0 or I^{2}C in advance by using port mode control register 4 (PMC4) and port function control register 4 (PFC4) (refer to **4.3.4 Port 4**).

Cautions 1. CSI0 or I^{2}C transmission/reception operations are not guaranteed if the mode is changed during transmission or reception. Be sure to disable the operation of the unit that is not used.

2. I^{2}C: μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

Figure 15-1. Selecting CSI0 or I^{2}C Mode

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

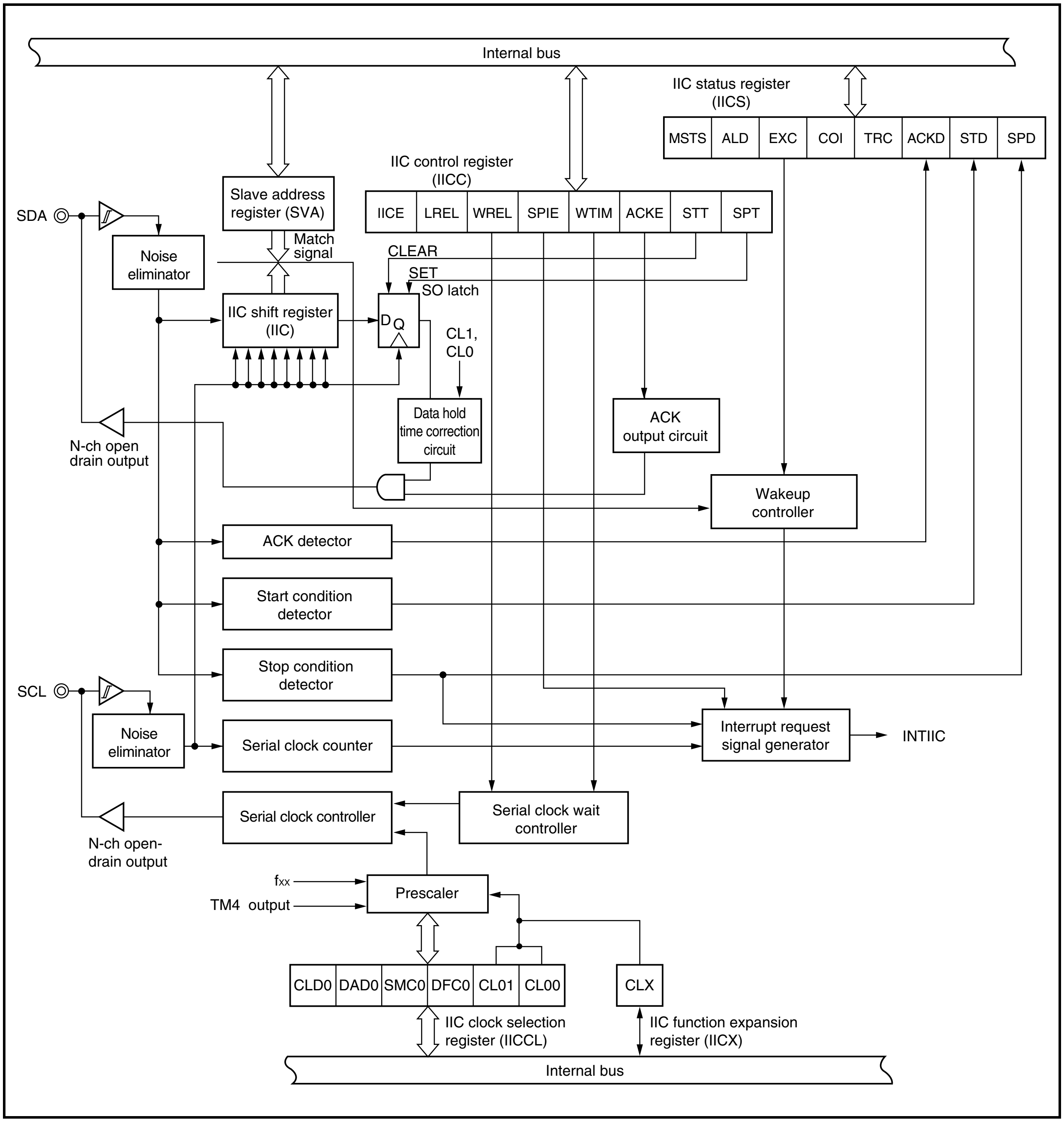
	7	6	5	4	3	2	1	0
PFC4	0	PFC46	0	PFC44	0	PFC42	PFC41	0

PMC4n	PFC4n	Operation mode
0	×	Port I/O mode
1	0	CSI0 mode
1	1	I^{2}C mode

Remarks 1. n = 1, 2

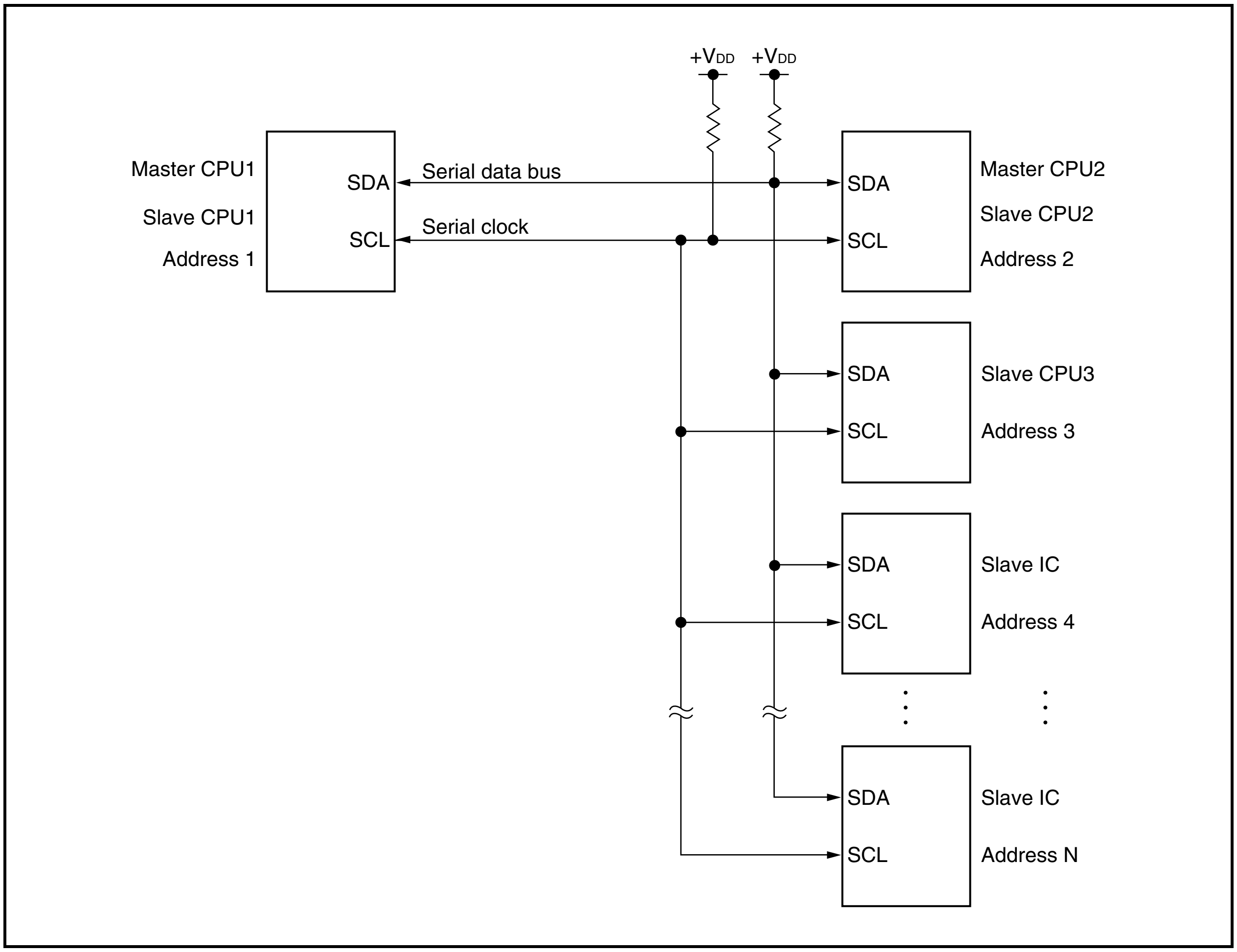
2. × = Don’t care

Figure 15-2. Block Diagram of I²C



A serial bus configuration example is shown below.

Figure 15-3. Example of Serial Bus Configuration Using I^{2}C Bus



15.2 Configuration

I^{2}C includes the following hardware.

Table 15-1. Configuration of I^{2}C

Item	Configuration
Registers	IIC shift register (IIC) Slave address register (SVA)
Control registers	IIC control register (IICC) IIC status register (IICS) IIC clock selection register (IICCL) IIC function expansion register (IICX)

(1) IIC shift register (IIC)

The IIC register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data. The IIC register can be used for both transmission and reception.

Write and read operations to the IIC register are used to control the actual transmit and receive operations.

The IIC register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

(2) Slave address register (SVA)

The SVA register sets local addresses when in slave mode.

The SVA register can be read or written in 8-bit units.

This register is cleared to 00H after reset.

(3) SO latch

The SO latch is used to retain the SDA pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request when the address received by this register matches the address value set to slave address register (SVA) or when an extension code is received.

(5) Clock selector

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC).
An I^{2}C interrupt is generated following either of two triggers.

- Eighth or ninth clock of the serial clock (set by WTIM bit[Note])
- Interrupt request generated when a stop condition is detected (set by SPIE bit[Note])

Note WTIM bit: Bit 3 of IIC control register (IICC)
SPIE bit: Bit 4 of IIC control register (IICC)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) ACK output circuit, stop condition detector, start condition detector, and ACK detector

These circuits are used to output and detect various control signals.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

15.3 Control Registers

I^{2}C is controlled by the following registers.

- IIC control register (IICC)
- IIC status register (IICS)
- IIC clock selection register (IICCL)
- IIC function expansion register (IICX)

The following registers are also used.

- IIC shift register (IIC)
- Slave address register (SVA)

(1) IIC control register (IICC)

The IICC register is used to enable/disable I^{2}C operations, set wait timing, and set other I^{2}C operations.
The IICC register can be read or written in 8-bit or 1-bit units.
This register is cleared to 00H after reset.

Caution When using the I^{2}C bus mode, set the port in the control mode (refer to 15.1.1 Switching modes between I^{2}C and CSI0).
Also set the N-ch open-drain output mode (refer to 4.3.4 (2) (e) Port function register 4 (PF4)).

(1/4)

After reset: 00H R/W Address: FFFFFD82H

	<7>	<6>	5	4	3	2	1	0
IICC	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I^2C operation enable/disable specification
0	Operation stopped. IIC status register (IICS) preset. Internal operation stopped.
1	Operation enabled.

Condition for clearing (IICE= 0)	Condition for setting (IICE = 1)
• Cleared by instruction • After reset	• Set by instruction

LREL	Exit from communications
0	Normal operation
1	This exits from the current communication operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL and SDA lines are set to high impedance. The following flags are cleared. • STD • ACKD • TRC • COI • EXC • MSTS • STT • SPT

The standby mode following exit from communications remains in effect until the following communication entry conditions are met.

- After a stop condition is detected, restart is in master mode.
- An address match or extension code reception occurs after the start condition.

Condition for clearing (LREL = 0)[Note]	Condition for setting (LREL = 1)
• Automatically cleared after execution • After reset	• Set by instruction

Note This flag's signal is invalid when IICE = 0.

Remark

STD:	Bit 1 of IIC status register (IICS)
ACKD:	Bit 2 of IIC status register (IICS)
TRC:	Bit 3 of IIC status register (IICS)
COI:	Bit 4 of IIC status register (IICS)
EXC:	Bit 5 of IIC status register (IICS)
MSTS:	Bit 7 of IIC status register (IICS)

(2/4)

WREL	Wait cancellation control
0	Wait not canceled
1	Wait canceled. This setting is automatically cleared after wait is canceled.

Condition for clearing (WREL = 0)[Note]	Condition for setting (WREL = 1)
• Automatically cleared after execution • After reset	• Set by instruction

SPIE	Enable/disable generation of interrupt request when stop condition is detected
0	Disabled
1	Enabled

Condition for clearing (SPIE = 0)[Note]	Condition for setting (SPIE = 1)
• Cleared by instruction • After reset	• Set by instruction

WTIM	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clockÕs falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the eighth clockÕs falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.

This bitÕs setting is invalid during an address transfer and is valid as the transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an $\overline{\text{ACK}}$ signal is issued. When the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.

Condition for clearing (WTIM = 0)[Note]	Condition for setting (WTIM = 1)
• Cleared by instruction • After reset	• Set by instruction

Note This flag’s signal is invalid when IICE = 0.

(3/4)

ACKE	Acknowledge control
0	Acknowledgment disable.
1	Acknowledgment enabled. During the ninth clock period, the SDA line is set to low level. However, $\overline{\text{ACK}}$ is invalid during address transfers and is valid when EXC = 1.

Condition for clearing (ACKE = 0)Note	Condition for setting (ACKE = 1)
• Cleared by instruction • After reset	• Set by instruction

STT	Start condition trigger
0	Start condition not generated.
1	When bus is released (in STOP mode): Generates a start condition (for starting as master). The SDA line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL is changed to low level. When bus is not used: This trigger functions as a start condition reserve flag. When set, it releases the bus and then automatically generates a start condition. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing

- For master reception: Cannot be set during transfer. Can be set only when ACKE has been set to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the ACK period. Set during the wait period.
- Cannot be set at the same time as SPT

Condition for clearing (STT = 0)	Condition for setting (STT = 1)
• Cleared by instruction • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When LREL = 1 • When IICE = 0 • After reset	• Set by instruction

Note This flag's signal is invalid when IICE = 0.

Remark Bit 1 (STT) is 0 if it is read immediately after data setting.

(4/4)

SPT	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master deviceÕs transfer). After the SDA line goes to low level, either set the SCL line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA line is changed from low l evel to high level and a stop condition is generated.

Cautions concerning set timing

- For master reception: Cannot be set during transfer.
  Can be set only when ACKE has been set to 0 and during the wait period after slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the ACK period. Set during the wait period.
- Cannot be set at the same time as STT.
- SPT can be set only when in master mode[Note].
- When WTIM has been set to 0, if SPT is set during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock.
  When a ninth clock must be output, WTIM should be changed from 0 to 1 during the wait period following output of eight clocks, and SPT should be set during the wait period that follows output of the ninth clock.

Condition for clearing (SPT = 0)	Condition for setting (SPT = 1)
• Cleared by instruction • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When LREL = 1 • When IICE = 0 • After reset	• Set by instruction

Note Set SPT only in master mode. However, SPT must be set and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status. For details, see **15.14 Cautions**.

Caution When bit 3 (TRC) of the IIC status register (IICS) is set to 1, WREL is set during the ninth clock and wait is canceled, after which TRC is cleared and the SDA line is set to high impedance.

Remark Bit 0 (SPT) is 0 if it is read immediately after data setting.

(2) IIC status register (IICS)

The IICS register is used to indicate the status of I^{2}C.

The IICS register is read-only, in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

(1/3)

After reset: 00H R Address: FFFFFD86H

	7	6	5	4	3	2	1	0
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS = 0)	Condition for setting (MSTS = 1)
• When a stop condition is detected • When ALD = 1 • Cleared by LREL = 1 • When IICE changes from 1 to 0 • After reset	• When a start condition is generated

ALD	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS is cleared.

Condition for clearing (ALD = 0)	Condition for setting (ALD = 1)
• Automatically cleared after IICS is read[Note] • When IICE changes from 1 to 0 • After reset	• When the arbitration result is a “loss”.

EXC	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • After reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This register is also cleared when a bit manipulation instruction is executed for bits other than IICS.

Remark LREL: Bit 6 of IIC control register (IICC)
IICE: Bit 7 of IIC control register (IICC)

(2/3)

COI	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • After reset	• When the received address matches the local address (SVA) (set at the rising edge of the eighth clock).

TRC	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA line is set to high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA line (valid starting at the falling edge of the first byte's ninth clock).

Condition for clearing (TRC = 0)	Condition for setting (TRC = 1)
• When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • Cleared by WREL = 1[Note] • When ALD changes from 0 to 1 • After reset Master • When "1" is output to the first byte's LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated Slave • When "1" is input by the first byte's LSB (transfer direction specification bit)

Note TRC is cleared and the SDA line becomes high impedance when bit 5 (WREL) of the IIC control register (IICC) is set and the wait state is released at ninth clock when bit 3 (TRC) of the IIC status register (IICS) = 1.

Remark WREL: Bit 5 of IIC control register (IICC)
LREL: Bit 6 of IIC control register (IICC)
IICE: Bit 7 of IIC control register (IICC)

(3/3)

ACKD	Detection of ACK
0	ACK was not detected.
1	ACK was detected.
Condition for clearing (ACKD = 0)	Condition for setting (ACKD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL = 1 • When IICE changes from 1 to 0 • After reset	• After the SDA line is set to low level at the rising edge of the SCL's ninth clock

STD	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect
Condition for clearing (STD = 0)	Condition for setting (STD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL = 1 • When IICE changes from 1 to 0 • After reset	When a start condition is detected

SPD	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.
Condition for clearing (SPD = 0)	Condition for setting (SPD = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When IICE changes from 1 to 0 • After reset	When a stop condition is detected

Remark LREL: Bit 6 of IIC control register (IICC)
IICE: Bit 7 of IIC control register (IICC)

(3) IIC clock selection register (IICCL)

The IICCL register is used to set the transfer clock for I^{2}C.

The IICCL register can be read or written in 8-bit or 1-bit units. Set the SMC, CL1, and CL0 bits in combination with the CLX bit of the IIC function expansion register (IICX) (see **Table 15-2 Transfer Clock Setting**).

This register is cleared to 00H after reset.

After reset: 00H R/W[Note] Address: FFFFFD84H

	7	6	5	4	3	2	1	0
IICCL	0	0	CLD	DAD	SMC	DFC	CL1	CL0

CLD	Detection of SCL line level (valid only when IICE = 1)
0	SCL line was detected at low level.
1	SCL line was detected at high level.

Condition for clearing (CLD = 0)	Condition for setting (CLD = 1)
• When the SCL line is at low level • When IICE = 0 • After reset	• When the SCL0 line is at high level

DAD	Detection of SDA line level (valid only when IICE = 1)
0	SDA line was detected at low level.
1	SDA line was detected at high level.

Condition for clearing (DAD = 0)	Condition for setting (DAD = 1)
• When the SDA line is at low level • When IICE = 0 • After reset	• When the SDA line is at high level

SMC	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFC	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
The digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFC switching (on/off).	

Note Bits 4 and 5 are read-only bits.

Remark IICE: Bit 7 of IIC control register (IICC)

(4) IIC function expansion register (IICX)

The IICX register is used to set the function expansion of I²C (valid only in high-speed mode).

The IICX register can be read or written in 8-bit or 1-bit units. Set the CLX bit in combination with the SMC, CL1, and CL0 bits of the IIC clock selection register (IICCL) (see **Table 15-2 Transfer Clock Setting**).

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFFD85H

	7	6	5	4	3	2	1	0
IICX	0	0	0	0	0	0	0	CLX

(5) I²C transfer clock setting method

The I²C transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 18, 24, 48, 86, 88, 172 (see **Table 15-2 Transfer Clock Setting**.)

T: $1/f_{XX}$

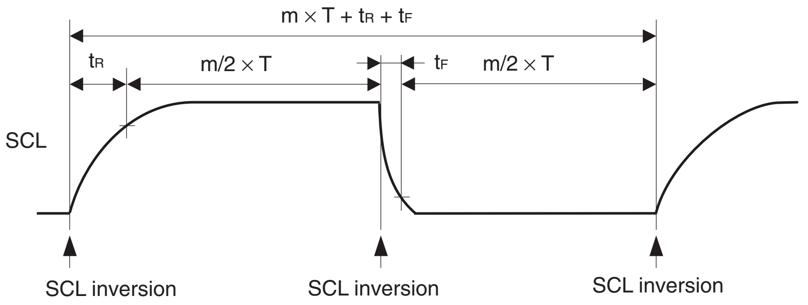
t_R: SCL rise time

t_F: SCL fall time

For example, the I²C transfer clock frequency (f_{SCL}) when f_{XX} = 16 MHz, m = 172, t_R = 200 ns, and t_F = 50 ns is calculated using the following expression.

$$f_{SCL} = 1/(172 \times 62.5 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 90.9 \text{ kHz}$$

Figure 15-4. I²C Transfer Clock Frequency (f_{SCL})



The transfer clock is set using a combination of the SMC, CL1, and CL0 bits of IIC clock select register (IICCL), the CLX bit of IIC function expansion register (IICX).

Table 15-2. Transfer Clock Setting

IICX	IICCL			Transfer Clock	Settable Main Clock Frequency (f_{XX}) Range	Operation Mode
Bit 0	Bit 3	Bit1	Bit0			
CLX	SMC	CL1	CL0			
0	0	0	0	f_{XX}/88	4.0 MHz to 8.38 MHz	Normal mode (SMC = 0)
0	0	0	1	f_{XX}/172	8.38 MHz to 16.76 MHz	
0	0	1	0	f_{XX}/86	4.19 MHz to 8.38 MHz	
0	0	1	1	TO4 output/66	TM4 Setting[Note]	
0	1	0	×	f_{XX}/48	8 MHz to 16.76 MHz	High-speed mode (SMC = 1)
0	1	1	0	f_{XX}/24	4 MHz to 8.38 MHz	
0	1	1	1	TO4 output/18	TM4 Setting[Note]	
1	0	×	×	Setting prohibited		
1	1	0	×	f_{XX}/24	8.00 MHz to 8.38 MHz	Normal mode (SMC = 0)
1	1	1	0	f_{XX}/12	4.00 MHz to 4.19 MHz	
1	1	1	1	Setting prohibited		

Remarks 1. ×: Don't care

2. When the transfer clock is set to timer output, the P96/TO4/A6 pin does not need to be set in timer output mode.

(6) IIC shift register (IIC)

The IIC register is used for serial transmission/reception (shift operations) that are synchronized with the serial clock.

The IIC register can be read or written in 8-bit units, but data should not be written to IIC during a data transfer.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFFD80H

	7	6	5	4	3	2	1	0
IIC								

(7) Slave address register (SVA)

The SVA register holds the I²C bus's slave addresses.

The SVA register can be read or written in 8-bit units, but bit 0 is fixed to 0.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFFD83H

	7	6	5	4	3	2	1	0
SVA								0

15.4 I^{2}C Bus Mode Functions

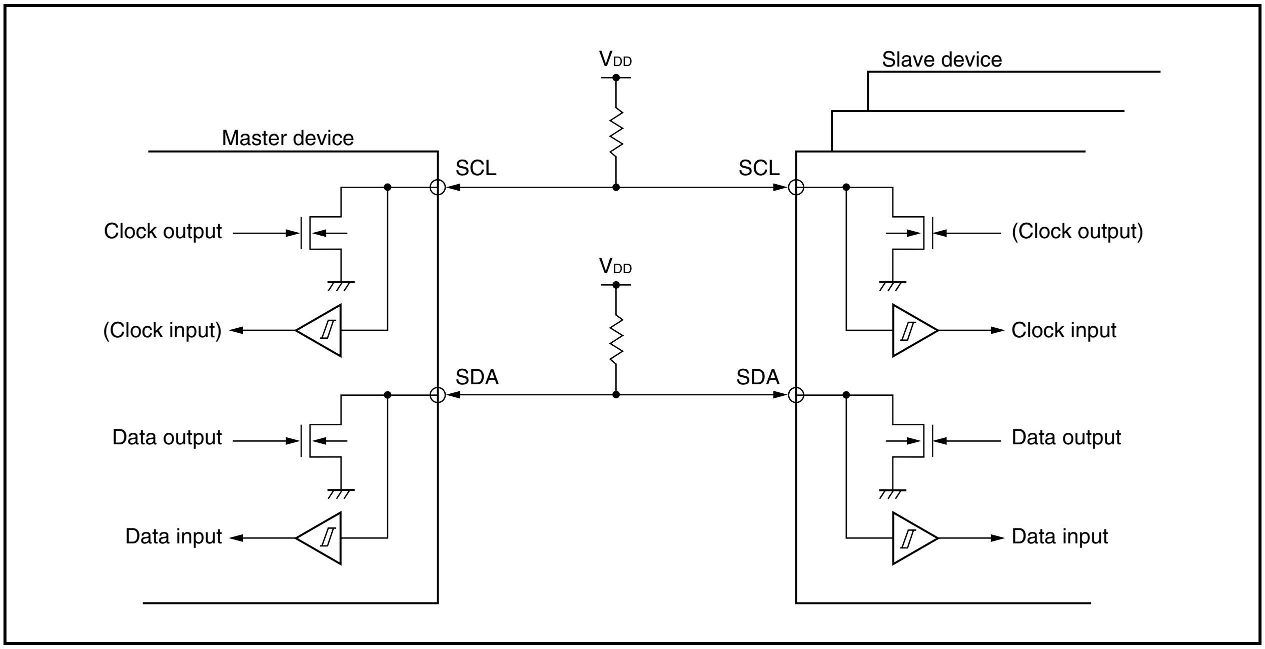
(1) Pin configuration

The serial clock pin (SCL) and serial data bus pin (SDA) are configured as follows.

SCL .... This pin is used for serial clock I/O.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
SDA .... This pin is used for serial data I/O.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

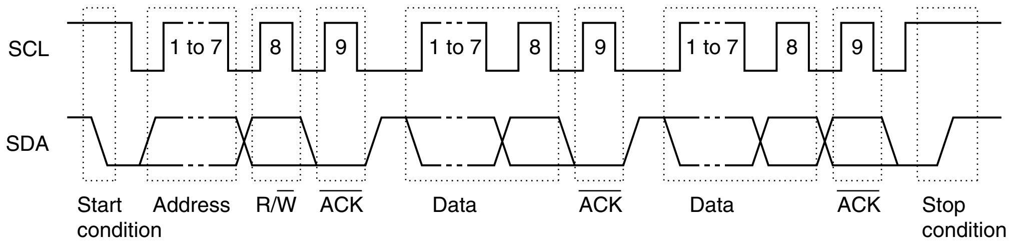
Figure 15-5. Pin Configuration Diagram



15.5 I^{2}C Bus Definitions and Control Methods

The following section describes the I^{2}C bus's serial data communication format and the signals used by the I^{2}C bus. The transfer timing for the "start condition", "data", and "stop condition" output via the I^{2}C bus's serial data bus is shown below.

Figure 15-6. I^{2}C Bus Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

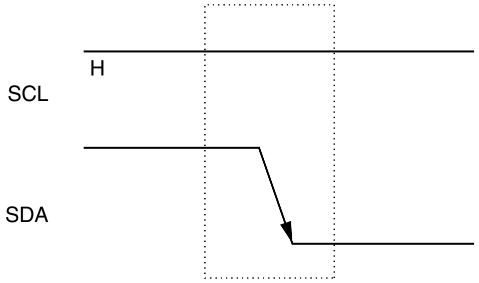
The acknowledge signal ($\overline{\text{ACK}}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL) is continuously output by the master device. However, in the slave device, SCL's low-level period can be extended and a wait can be inserted.

(1) Start condition

A start condition is met when the SCL pin is at high level and the SDA pin changes from high level to low level. The start conditions for the SCL pin and SDA pin are signals that the master device outputs to the slave device when starting a serial transfer. The slave device includes hardware for detecting start conditions.

Figure 15-7. Start Conditions



A start condition is output when bit 1 (STT) of the IIC control register (IICC) is set to 1 after a stop condition has been detected (SPD: Bit 0 = 1 in IIC status register (IICS)). When a start condition is detected, bit 1 (STD) of IICS is set to 1.

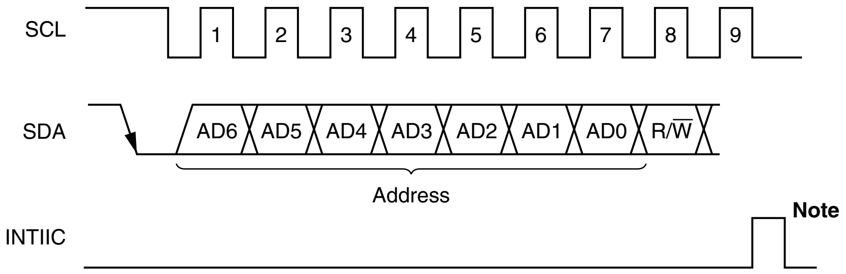
(2) Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit data matches the data values stored in slave address register (SVA). If the 7-bit data matches the SVA register values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 15-8. Address



Note INTIIC is not generated if data other than a local address or extension code is received during slave device operation.

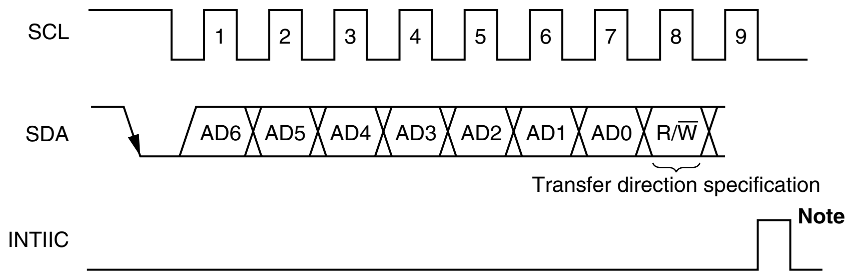
The slave address and the eighth bit, which specifies the transfer direction as described in **(3) Transfer direction specification** below, are written together to the IIC shift register (IIC) and are then output. Received addresses are written to IIC.

The slave address is assigned to the higher 7 bits of the IIC register.

(3) Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 15-9. Transfer Direction Specification



Note INTIIC is not generated if data other than a local address or extension code is received during slave device operation.

(4) Acknowledge signal ($\overline{\text{ACK}}$)

The acknowledge signal ($\overline{\text{ACK}}$) is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{\text{ACK}}$ signal for each 8 bits of data it receives. The transmitting device normally receives an $\overline{\text{ACK}}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{\text{ACK}}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{\text{ACK}}$ signal is returned after it transmits 8 bits of data. When an $\overline{\text{ACK}}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{\text{ACK}}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{\text{ACK}}$ signal may be caused by the following two factors.

<1> Reception was not correctly performed.

<2> The final data was received.

When the receiving device sets the SDA line to low level during the ninth clock, the $\overline{\text{ACK}}$ signal becomes active (normal receive response).

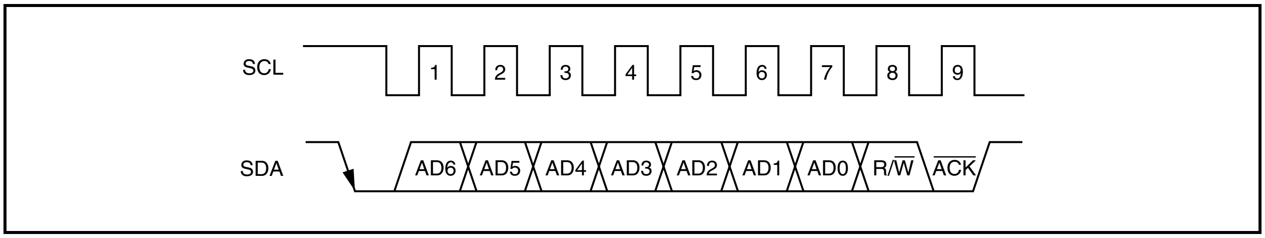
When bit 2 (ACKE) of the IIC control register (IICC) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled.

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRC) of the IIC status register (IICS) to be set. When the TRC bit's value is 0, it indicates receive mode. Therefore, ACKE should be set to 1.

When the slave device is receiving (when TRC = 0), if the slave device does not need to receive any more data after receiving several bytes, setting ACKE to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRC = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting ACKE to 0 will prevent the $\overline{\text{ACK}}$ signal from being returned. This prevents the MSB data from being output via the SDA line (i.e., stops transmission) during transmission from the slave device.

Figure 15-10. $\overline{\text{ACK}}$ Signal



When the local address is received, an $\overline{\text{ACK}}$ signal is automatically output in synchronization with the falling edge of the eighth clock of SCL regardless of the ACKE bit value. No $\overline{\text{ACK}}$ signal is output if the received address is not a local address.

The $\overline{\text{ACK}}$ signal output method during data reception is based on the wait timing setting, as described below.

When 8-clock wait is selected: The $\overline{\text{ACK}}$ signal is output at the falling edge of the eighth clock of SCL if ACKE is set to 1 before wait cancellation.

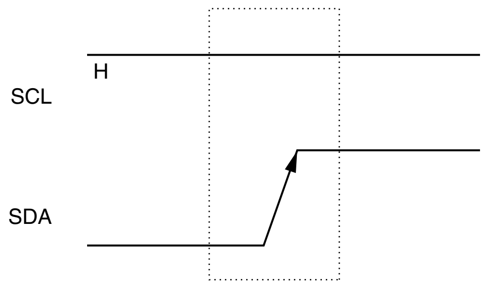
When 9-clock wait is selected: The $\overline{\text{ACK}}$ signal is automatically output at the falling edge of the eighth clock of SCL if ACKE has already been set to 1.

(5) Stop condition

When the SCL pin is at high level, changing the SDA pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device outputs to the slave device when serial transfer has been completed. The slave device includes hardware that detects stop conditions.

Figure 15-11. Stop Condition



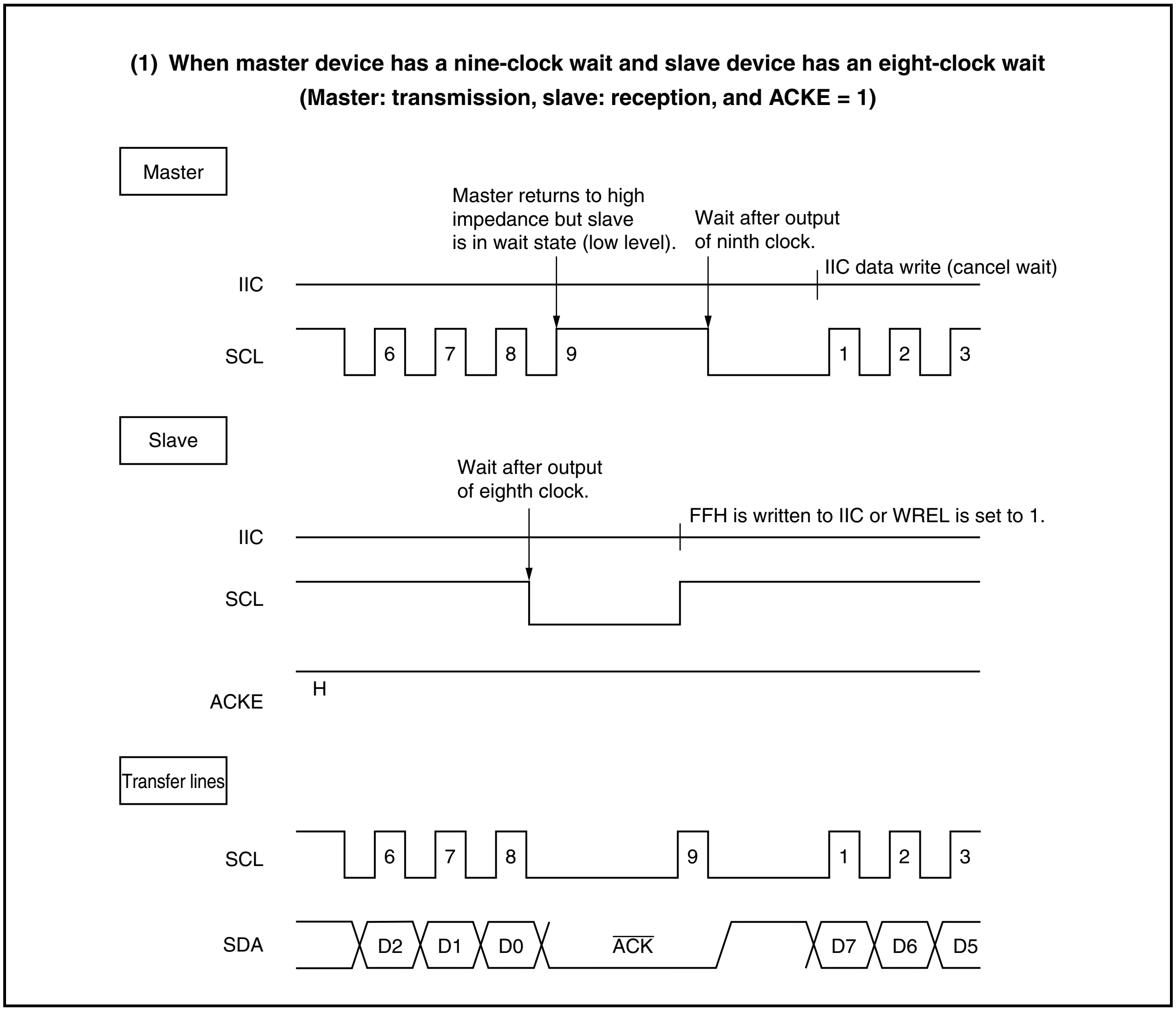
A stop condition is generated when bit 0 (SPT) of the IIC control register (IICC) is set to 1. When the stop condition is detected, bit 0 (SPD) of the IIC status register (IICS) is set to 1 and INTIIC is generated when bit 4 (SPIE) of IICC is set to 1.

(6) Wait signal ($\overline{\text{WAIT}}$)

The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL pin to low level notifies the communication partner of the wait status. When the wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 15-12. Wait Signal (1/2)

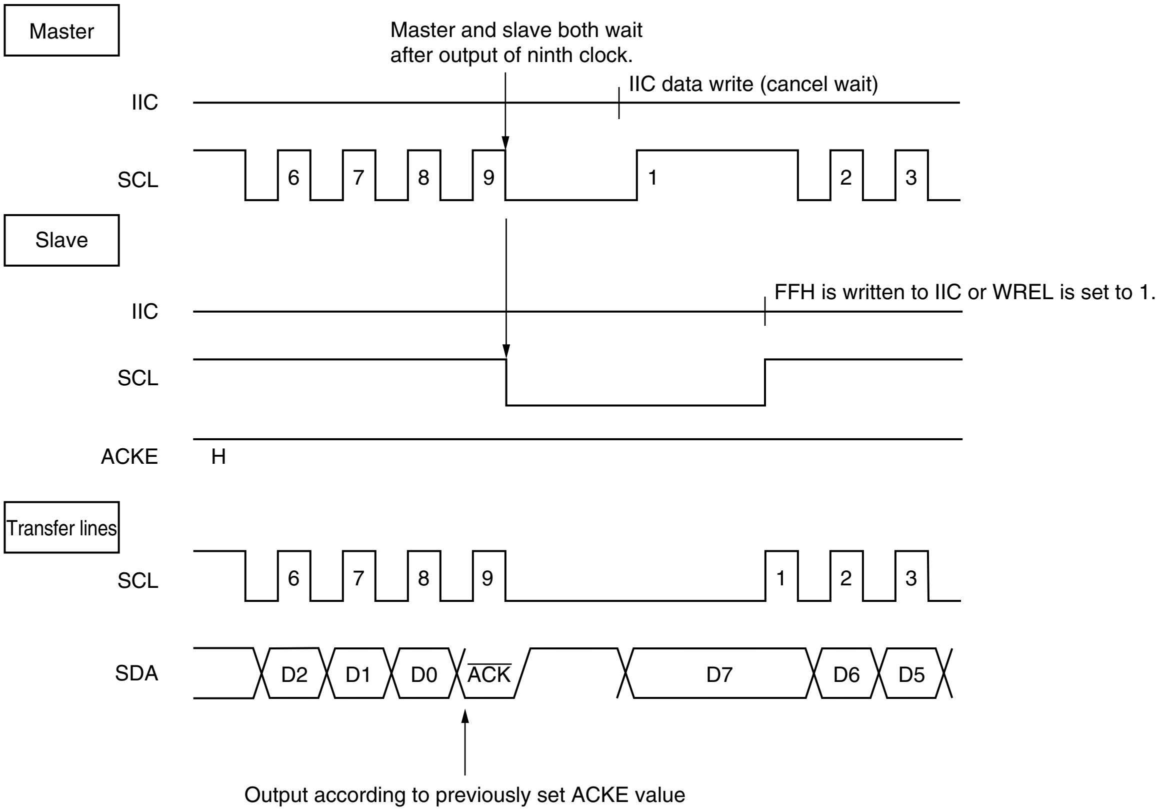


(1) When master device has a nine-clock wait and slave device has an eight-clock wait (Master: transmission, slave: reception, and ACKE = 1)

Master
Master returns to high impedance but slave is in wait state (low level).
Wait after output of ninth clock.
IIC data write (cancel wait)
IIC
SCL
6
7
8
9
1
2
3
Slave
Wait after output of eighth clock.
FFH is written to IIC or WREL is set to 1.
IIC
SCL
ACKE
H
Transfer lines
SCL
6
7
8
9
1
2
3
SDA
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5

Figure 15-12. Wait Signal (2/2)

(2) When master and slave devices both have a nine-clock wait (Master: transmission, slave: reception, and ACKE = 1)


Master
Master and slave both wait after output of ninth clock.
IIC
IIC data write (cancel wait)
SCL
6
7
8
9
1
2
3
Slave
IIC
FFH is written to IIC or WREL is set to 1.
SCL
ACKE
H
Transfer lines
SCL
6
7
8
9
1
2
3
SDA
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5
Output according to previously set ACKE value

Remark ACKE: Bit 2 of IIC control register (IICC)
WREL: Bit 5 of IIC control register (IICC)

A wait may be automatically generated depending on the setting of bit 3 (WTIM) of the IIC control register (IICC).

Normally, when bit 5 (WREL) of the IICC register is set to 1 or when FFH is written to the IIC shift register (IIC), the wait status is canceled and the transmitting side writes data to the IIC register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STT) of IICC to 1
- By setting bit 0 (SPT) of IICC to 1

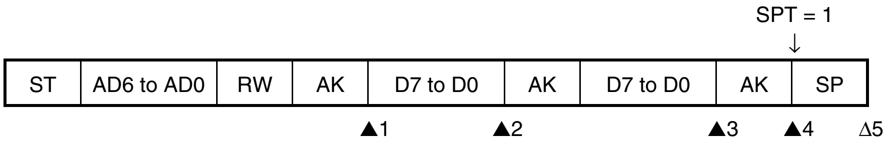
15.6 I²C Interrupt Request (INTIIC)

The following shows the value of the IIC status register (IICS) at the INTIIC interrupt request generation timing and at the INTIIC interrupt timing.

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When WTIM = 0



▲1: IICS = 10XXX110B

▲2: IICS = 10XXX000B

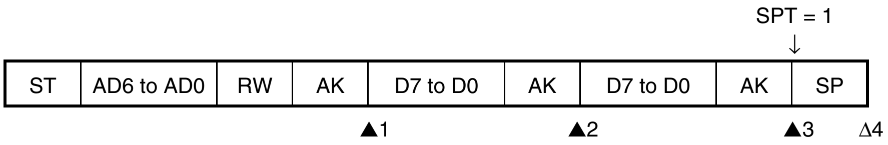
▲3: IICS = 10XXX000B (WTIM = 0)

▲4: IICS = 10XXXX00B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1



▲1: IICS = 10XXX110B

▲2: IICS = 10XXX100B

▲3: IICS = 10XXXX00B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM = 0

					STT = 1 ↓						SPT = 1 ↓	
ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2	▲3				▲4	▲5	▲6	Δ7

▲1: IICS = 10XXX110B

▲2: IICS = 10XXX000B (WTIM = 1)

▲3: IICS = 10XXXX00B (WTIM = 0)

▲4: IICS = 10XXX110B (WTIM = 0)

▲5: IICS = 10XXX000B (WTIM = 1)

▲6: IICS = 10XXXX00B

Δ 7: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

<2> When WTIM = 1

					STT = 1 ↓						SPT = 1 ↓	
ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3		▲4	Δ5

▲1: IICS = 10XXX110B

▲2: IICS = 10XXXX00B

▲3: IICS = 10XXX110B

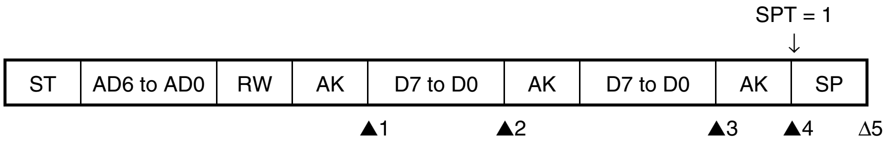
▲4: IICS = 10XXXX00B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM = 0



▲1: IICS = 1010X110B

▲2: IICS = 1010X000B

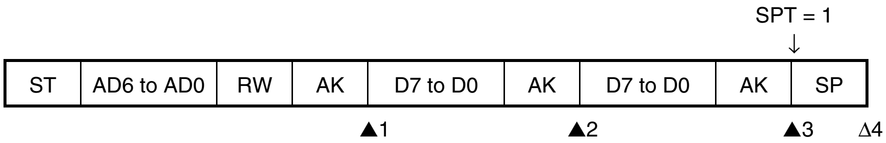
▲3: IICS = 1010X000B (WTIM = 1)

▲4: IICS = 1010XX00B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1



▲1: IICS = 1010X110B

▲2: IICS = 1010X100B

▲3: IICS = 1010XX00B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(2) Slave device operation (when receiving slave address data (matches with SVA))

(a) Start ~ Address ~ Data ~ Data ~ Stop

<1> When WTIM = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		Δ4

▲1: IICS = 0001X110B

▲2: IICS = 0001X000B

▲3: IICS = 0001X000B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS = 0001X110B

▲2: IICS = 0001X100B

▲3: IICS = 0001XX00B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM = 0 (after restart, matches with SVA)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2					▲3	▲4		Δ5

▲1: IICS = 0001X110B

▲2: IICS = 0001X000B

▲3: IICS = 0001X110B

▲4: IICS = 0001X000B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

<2> When WTIM = 1 (after restart, matches with SVA)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3		▲4	Δ5

▲1: IICS = 0001X110B

▲2: IICS = 0001XX00B

▲3: IICS = 0001X110B

▲4: IICS = 0001XX00B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2				▲3		▲4		Δ5

▲1: IICS = 0001X110B

▲2: IICS = 0001X000B

▲3: IICS = 0010X010B

▲4: IICS = 0010X000B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2			▲3	▲4		▲5	Δ6

▲1: IICS = 0001X110B

▲2: IICS = 0001XX00B

▲3: IICS = 0010X010B

▲4: IICS = 0010X110B

▲5: IICS = 0010XX00B

Δ 6: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM = 0 (after restart, does not match with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2					▲3			Δ4

▲1: IICS = 0001X110B

▲2: IICS = 0001X000B

▲3: IICS = 00000X10B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1 (after restart, does not match with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3			Δ4

▲1: IICS = 0001X110B

▲2: IICS = 0001XX00B

▲3: IICS = 00000X10B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(3) Slave device operation (when receiving extension code)

(a) Start ~ Code ~ Data ~ Data ~ Stop

<1> When WTIM = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS = 0010X010B

▲2: IICS = 0010X000B

▲3: IICS = 0010X000B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS = 0010X010B

▲2: IICS = 0010X110B

▲3: IICS = 0010X100B

▲4: IICS = 0010XX00B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM = 0 (after restart, matches with SVA)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
		▲1		▲2					▲3	▲4		Δ5

▲1: IICS = 0010X010B

▲2: IICS = 0010X000B

▲3: IICS = 0001X110B

▲4: IICS = 0001X000B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1 (after restart, matches with SVA)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
		▲1	▲2		▲3				▲4		▲5	Δ6

▲1: IICS = 0010X010B

▲2: IICS = 0010X110B

▲3: IICS = 0010XX00B

▲4: IICS = 0001X110B

▲5: IICS = 0001XX00B

Δ 6: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1		▲2				▲3		▲4		Δ5

▲1: IICS = 0010X010B

▲2: IICS = 0010X000B

▲3: IICS = 0010X010B

▲4: IICS = 0010X000B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

<2> When WTIM = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3			▲4	▲5		▲6	Δ7

▲1: IICS = 0010X010B

▲2: IICS = 0010X110B

▲3: IICS = 0010XX00B

▲4: IICS = 0010X010B

▲5: IICS = 0010X110B

▲6: IICS = 0010XX00B

Δ 7: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM = 0 (after restart, does not match with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2					▲3		Δ4

▲1: IICS = 0010X010B

▲2: IICS = 0010X000B

▲3: IICS = 00000X10B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1 (after restart, does not match with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2		▲3				▲4		Δ5

▲1: IICS = 0010X010B

▲2: IICS = 0010X110B

▲3: IICS = 0010XX00B

▲4: IICS = 00000X10B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
								Δ1

Δ 1: IICS = 00000001B

Remark Δ: Generated only when SPIE = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

<1> When WTIM = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		Δ4

▲1: IICS = 0101X110B (Example: When ALD is read during interrupt servicing)

▲2: IICS = 0001X000B

▲3: IICS = 0001X000B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS = 0101X110B (Example: When ALD is read during interrupt servicing)

▲2: IICS = 0001X100B

▲3: IICS = 0001XX00B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(b) When arbitration loss occurs during transmission of extension code

<1> When WTIM = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS = 0110X010B (Example: When ALD is read during interrupt servicing)

▲2: IICS = 0010X000B

▲3: IICS = 0010X000B

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

<2> When WTIM = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		▲4 Δ5

▲1: IICS = 0110X010B (Example: When ALD is read during interrupt servicing)

▲2: IICS = 0010X110B

▲3: IICS = 0010X100B

▲4: IICS = 0010XX00B

Δ 5: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1					Δ2

▲1: IICS = 01000110B (Example: When ALD is read during interrupt servicing)

Δ 2: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1

(b) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
		▲1						Δ2

▲1: IICSn = 0110X010B (Example: When ALD is read during interrupt servicing)

IICC's LREL is set to 1 via software

Δ 2: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care

(c) When arbitration loss occurs during data transfer

<1> When WTIM = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
				▲1	▲2				Δ3

▲1: IICS = 10001110B

▲2: IICS = 01000000B (Example: When ALD is read during interrupt servicing)

Δ 3: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1

<2> When WTIM = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
				▲1		▲2			Δ3

▲1: IICS = 10001110B

▲2: IICS = 01000100B (Example: When ALD is read during interrupt servicing)

Δ 3: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1

(d) When loss occurs due to restart condition during data transfer

<1> Not extension code (Example: mismatches with SVA)

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1					▲2			Δ3

▲1: IICS = 1000X110B

▲2: IICS = 01000110B (Example: When ALD is read during interrupt servicing)

Δ 3: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care
Dn = D6 to D0

<2> Extension code

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1				▲2				Δ3

▲1: IICS = 1000X110B

▲2: IICS = 0110X010B (Example: When ALD is read during interrupt servicing)

IICC's LREL is set to 1 via software

Δ 3: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don't care
Dn = D6 to D0

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	RW	AK	D7 to Dn	SP
			▲1		Δ2

▲1: IICS = 1000X110B

Δ 2: IICS = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care
Dn = D6 to D0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

When WTIM = 1

					STT = 1 ↓					
ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3			Δ4

▲1: IICS = 1000X110B

▲2: IICS = 1000XX00B

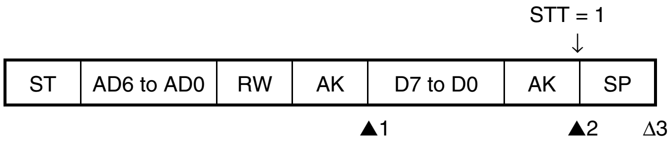
▲3: IICS = 01000100B (Example: When ALD is read during interrupt servicing)

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

When WTIM = 1



▲1: IICS = 1000X110B

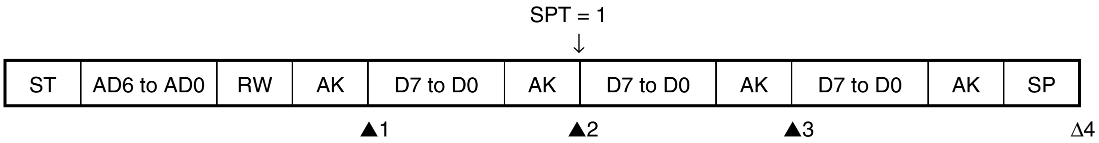
▲2: IICS = 1000XX00B

Δ 3: IICS = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

When WTIM = 1



▲1: IICS = 1000X110B

▲2: IICS = 1000XX00B

▲3: IICS = 01000000B (Example: When ALD is read during interrupt servicing)

Δ 4: IICS = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: Don’t care

15.7 Interrupt Request (INTIIC) Generation Timing and Wait Control

The setting of bit 3 (WTIM) of the IIC control register (IICC) determines the timing by which INTIIC is generated and the corresponding wait control, as shown below.

Table 15-3. INTIIC Signal Generation Timing and Wait Control

WTIM	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9[Notes 1, 2]	8[Note 2]	8[Note 2]	9	8	8
1	9[Notes 1, 2]	9[Note 2]	9[Note 2]	9	9	9

Notes 1. The slave device's INTIIC signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register (SVA).
At this point, $\overline{\text{ACK}}$ is output regardless of the value set to bit 2 (ACKE) of the IICC register. For a slave device that has received an extension code, the INTIIC signal occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of slave address register (SVA), neither the INTIIC signal nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: The interrupt and wait timing are determined regardless of the WTIM bit.
- Master device operation: The interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: The interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: The interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WREL) of the IIC control register (IICC) to 1
- By writing to the IIC shift register (IIC)
- By start condition setting (bit 1 (STT) of IIC control register (IICC) = 1)
- By stop condition setting (bit 0 (SPT) of IIC control register (IICC) = 1)

When an 8-clock wait has been selected (WTIM = 0), the output level of $\overline{\text{ACK}}$ must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC signal is generated when a stop condition is detected.

15.8 Address Match Detection Method

When in I^{2}C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An interrupt request (INTIIC) occurs when a local address has been set to the slave address register (SVA) and when the address set to SVA matches the slave address sent by the master device, or when an extension code has been received.

15.9 Error Detection

In I^{2}C bus mode, the status of the serial data bus (SDA) during data transmission is captured by the IIC shift register (IIC) of the transmitting device, so the IIC data prior to transmission can be compared with the transmitted IIC data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

15.10 Extension Code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXC) is set for extension code reception and an interrupt request (INTIIC) is issued at the falling edge of the eighth clock.
The local address stored in the slave address register (SVA) is not affected.

(2) If 11110xx0 is set to SVA by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC signal occurs at the falling edge of the eighth clock.

Higher four bits of data match: EXC = 1^Note
Seven bits of data match: COI = 1^Note

Note EXC: Bit 5 of IIC status register (IICS)
COI: Bit 4 of IIC status register (IICS)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
For example, when operation as a slave is not desired after the extension code is received, set bit 6 (LREL) of the IIC control register (IICC) to 1 and the CPU will enter the next communication wait state.

Table 15-4. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	×	CBUS address
0000 010	×	Address that is reserved for different bus format
1111 0xx	×	10-bit slave address specification

15.11 Arbitration

When several master devices simultaneously output a start condition (when STT is set to 1 before STD is set to 1[Note]), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD) in the IIC status register (IICS) is set via the timing by which the arbitration loss occurred, and the SCL and SDA lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD = 1 setting that has been made by software.

For details of interrupt request timing, see **15.6 I^{2}C Interrupt Request (INTIIC)**.

Note STD: Bit 1 of IIC status register (IICS)
STT: Bit 1 of IIC control register (IICC)

Figure 15-13. Arbitration Timing Example

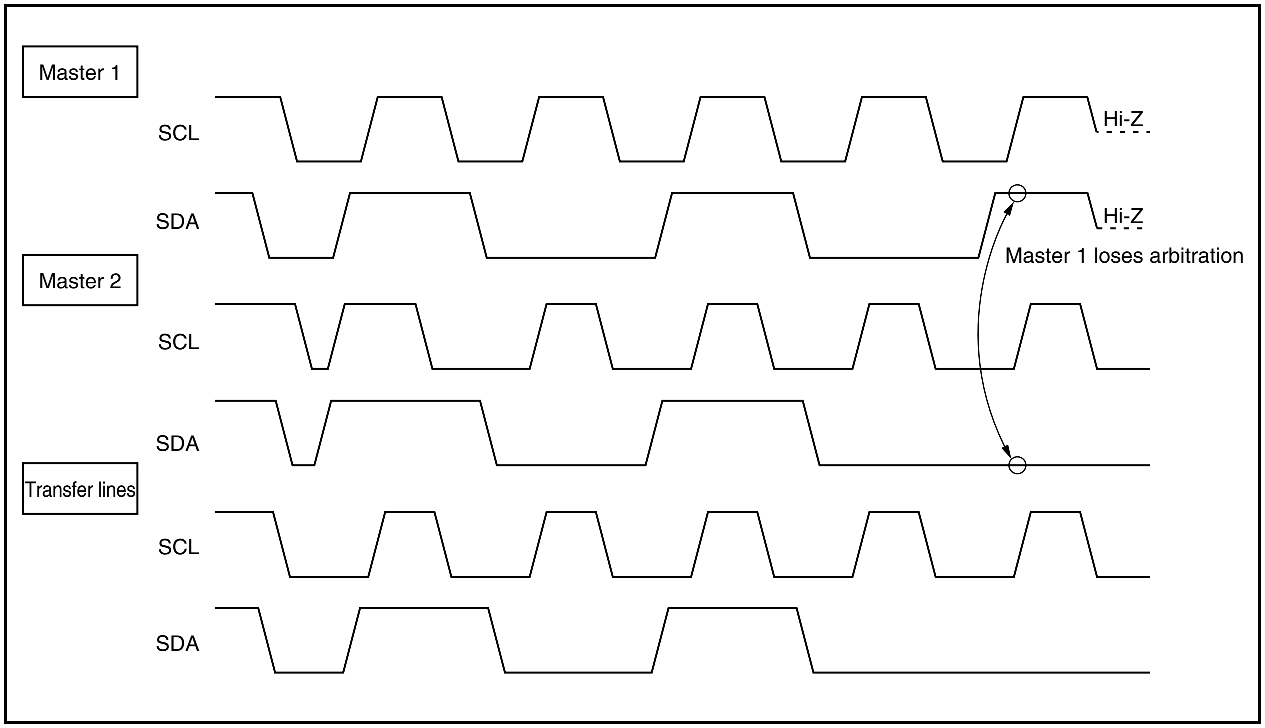


Table 15-5. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ signal transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIE = 1)[Note 2]
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIE = 1)[Note 2]
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL is at low level while attempting to output a restart condition	

Notes 1. When WTIM (bit 3 of IIC control register (IICC)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a possibility that arbitration will occur, set SPIE = 1 for master device operation.

Remark SPIE: Bit 5 of IIC control register (IICC)

15.12 Wakeup Function

The I^{2}C bus slave function is a function that generates an interrupt request (INTIIC) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has output a start condition) to a slave device.

However, when a stop condition is detected, bit 5 (SPIE) of the IIC control register (IICC) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

15.13 Communication Reservation

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when bit 6 (LREL) of the IIC control register (IICC) was set to “1”).

If bit 1 (STT) of IICC is set while the bus is not being used, a start condition is automatically generated and the wait status is set after the bus is released (after a stop condition is detected).
When the bus release is detected (when a stop condition is detected), writing to the IIC shift register (IIC) causes the master’s address transfer to start. At this point, bit 4 (SPIE) of IICC should be set.
When the STT bit has been set, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been released ....a start condition is generated
If the bus has not been released (standby mode) ....communication reservation

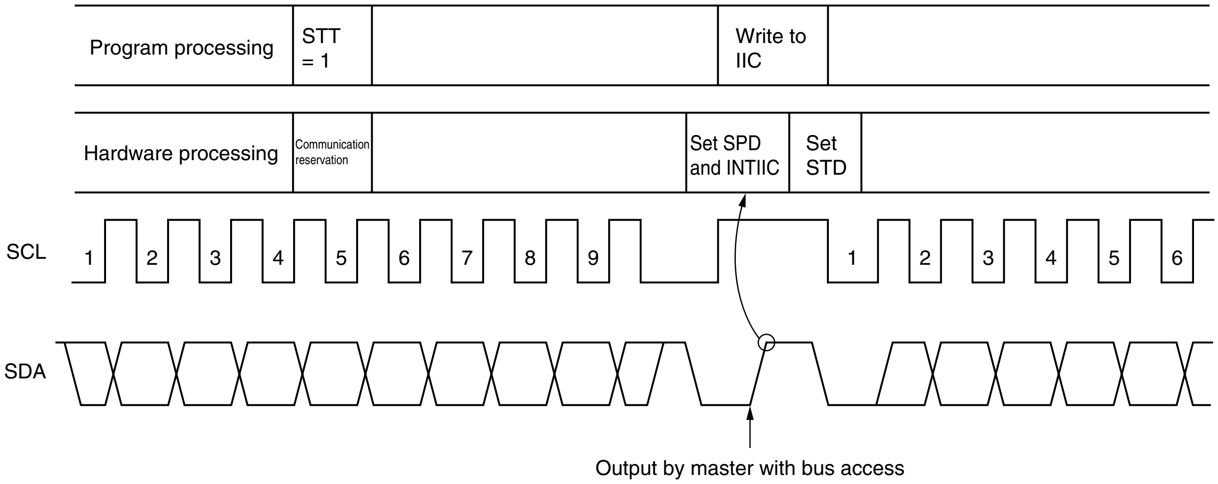
To detect which operation mode has been determined for the STT bit, set the STT bit, wait for the wait period, then check the MSTS (bit 7 of IIC status register (IICS)).
Wait periods, which should be set via software, are listed in Table 15-6. These wait periods can be set via the settings for bits 3, 1, and 0 (SMC, CL1, and CL0) in the IIC clock selection register (IICCL).

Table 15-6. Wait Periods

SMC	CL1	CL0	Wait Period
0	0	0	26 clocks
0	0	1	46 clocks
0	1	0	92 clocks
0	1	1	37 clocks
1	0	0	16 clocks
1	0	1	
1	1	0	32 clocks
1	1	1	13 clocks

The communication reservation timing is shown below.

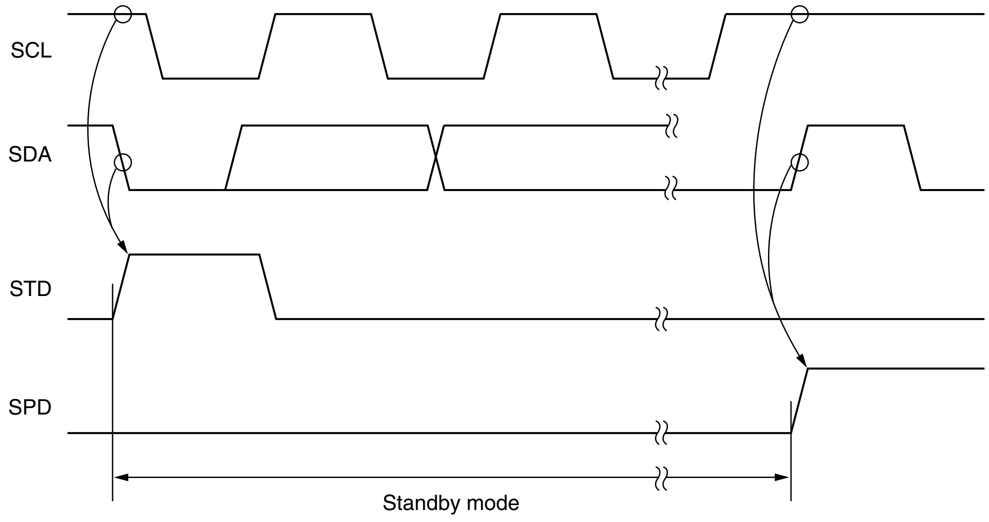
Figure 15-14. Communication Reservation Timing



IIC: IIC shift register
STT: Bit 1 of IIC control register (IICC)
STD: Bit 1 of IIC status register (IICS)
SPD: Bit 0 of IIC status register (IICS)

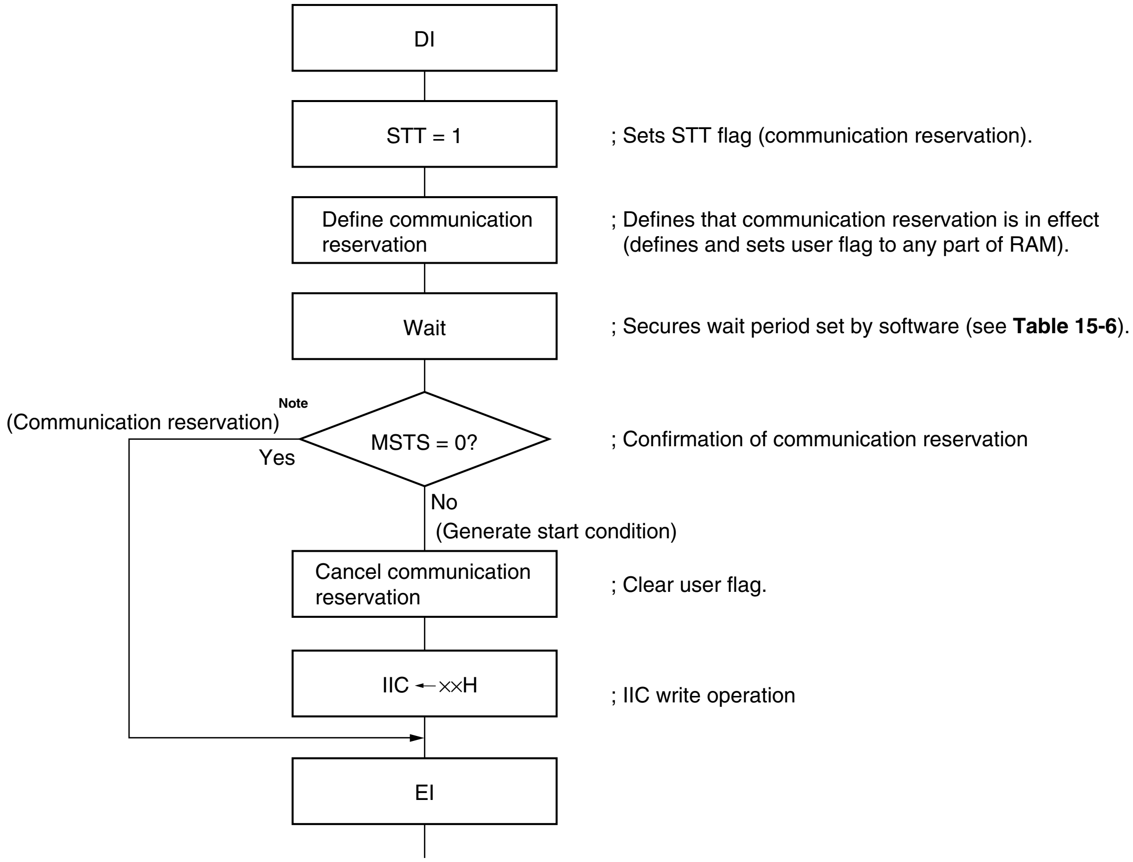
Communication reservations are accepted of the following timing. After bit 1 (STD) of the IIC status register (IICS) is set to 1, a communication reservation can be made by setting bit 1 (STT) of the IIC control register (IICC) to 1 before a stop condition is detected.

Figure 15-15. Timing for Accepting Communication Reservations



The communication reservation flow chart is illustrated below.

Figure 15-16. Communication Reservation Flow Chart


DI
STT = 1
; Sets STT flag (communication reservation).
Define communication reservation
; Defines that communication reservation is in effect (defines and sets user flag to any part of RAM).
Wait
; Secures wait period set by software (see **Table 15-6**).
(Communication reservation)Note
MSTS = 0?
Yes
; Confirmation of communication reservation
No
(Generate start condition)
Cancel communication reservation
; Clear user flag.
IIC ← ××H
; IIC write operation
EI

Note The communication reservation operation executes a write to the IIC shift register (IIC) when a stop condition interrupt request occurs.

15.14 Cautions

After a reset, when changing from a mode in which no stop condition has been detected (the bus has not been released) to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

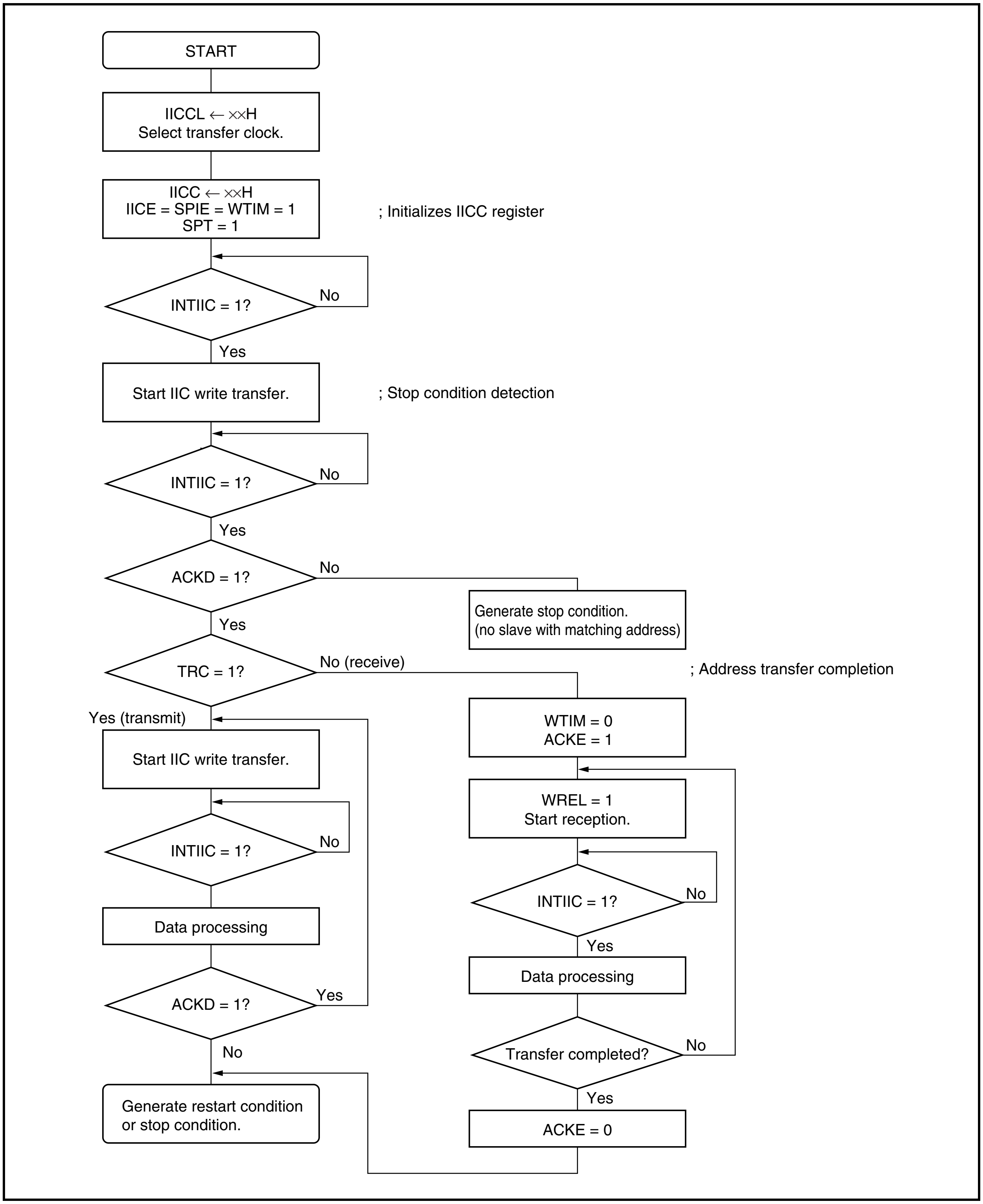
<1> Set the IIC clock selection register (IICCL).
<2> Set bit 7 (IICE) of the IIC control register (IICC).
<3> Set bit 0 of IICC.

15.15 Communication Operations

(1) Master operations

The following is a flow chart of the master operations.

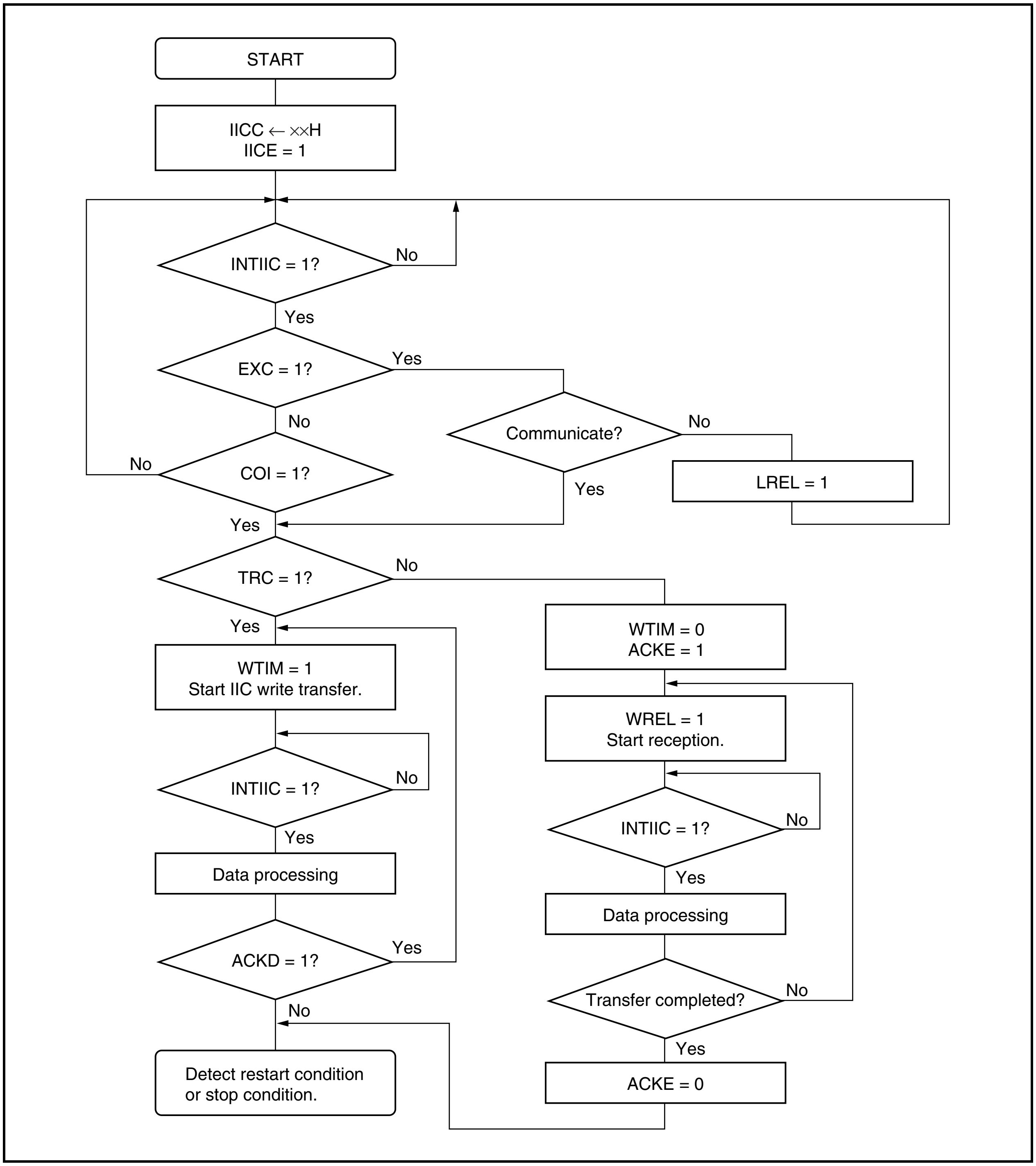
Figure 15-17. Master Operation Flow Chart



(2) Slave operation

An example of slave operation is shown below.

Figure 15-18. Slave Operation Flow Chart



15.16 Timing of Data Communication

When using I^{2}C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC bit (bit 3 of the IIC status register (IICS)), which specifies the data transfer direction and then starts serial communication with the slave device.

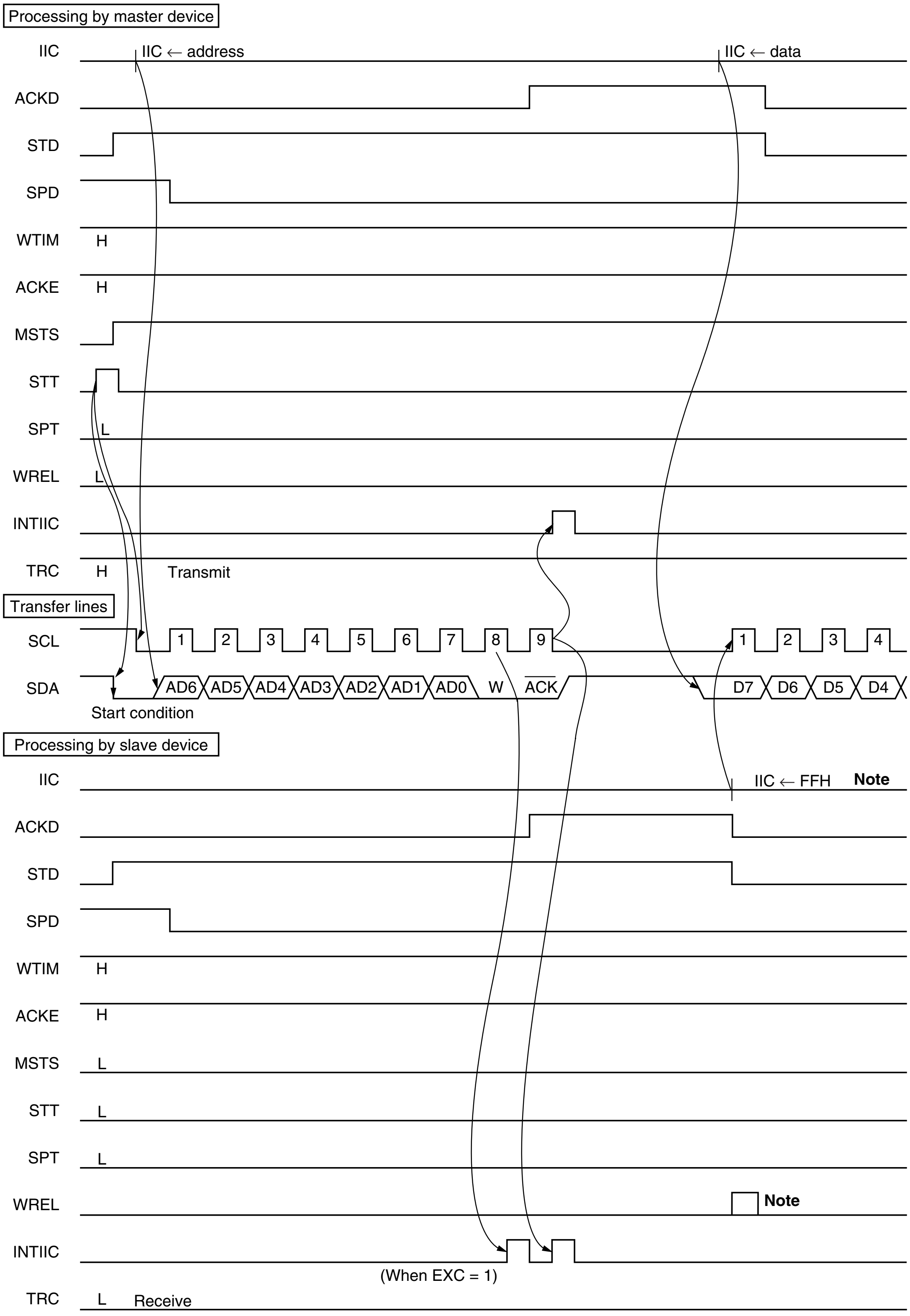
The shift operation of the IIC bus shift register (IIC) is synchronized with the falling edge of the serial clock (SCL). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA pin.

Data input via the SDA pin is captured by IIC at the rising edge of SCL.

The data communication timing is shown below.

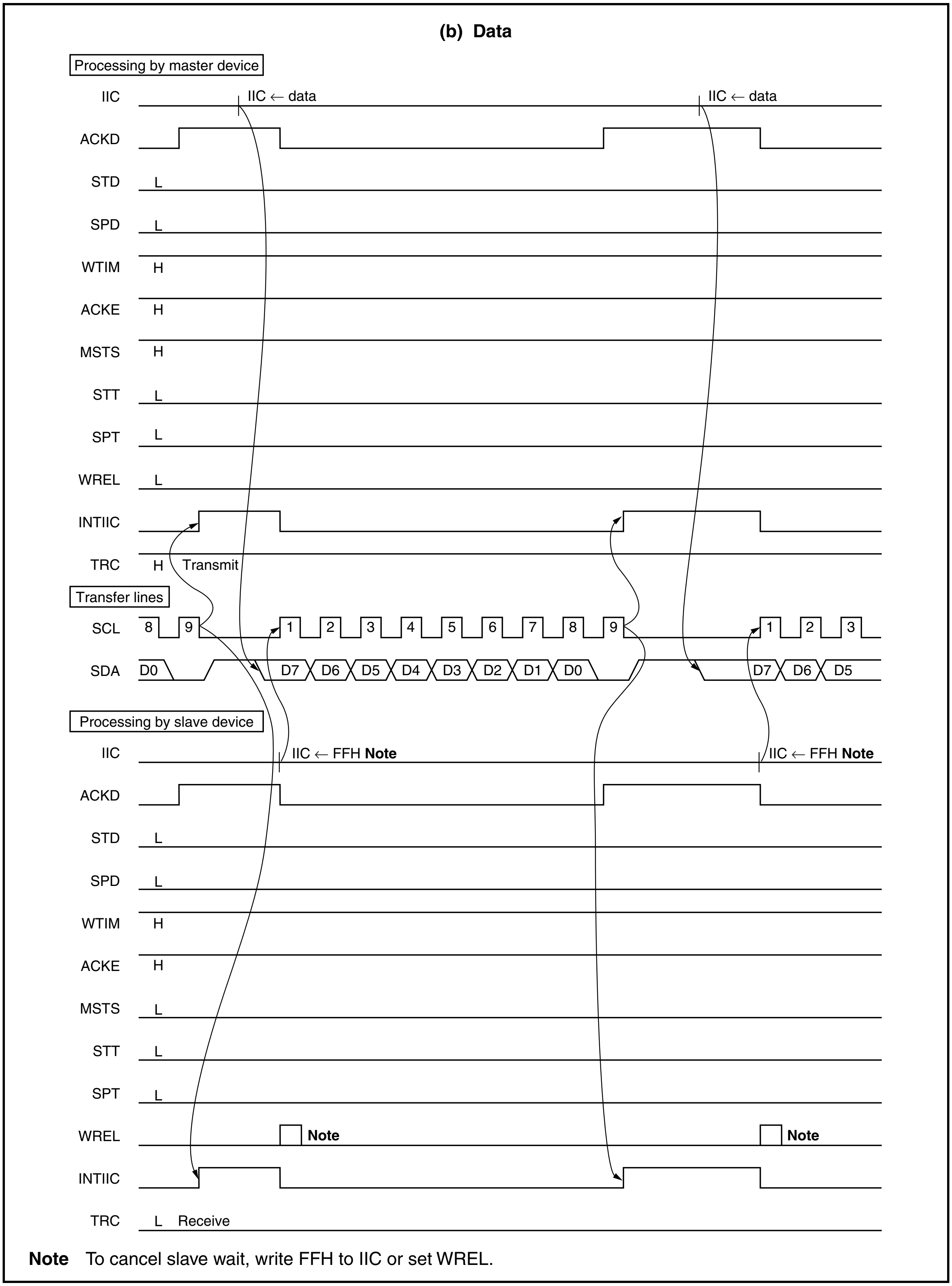
Figure 15-19. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address



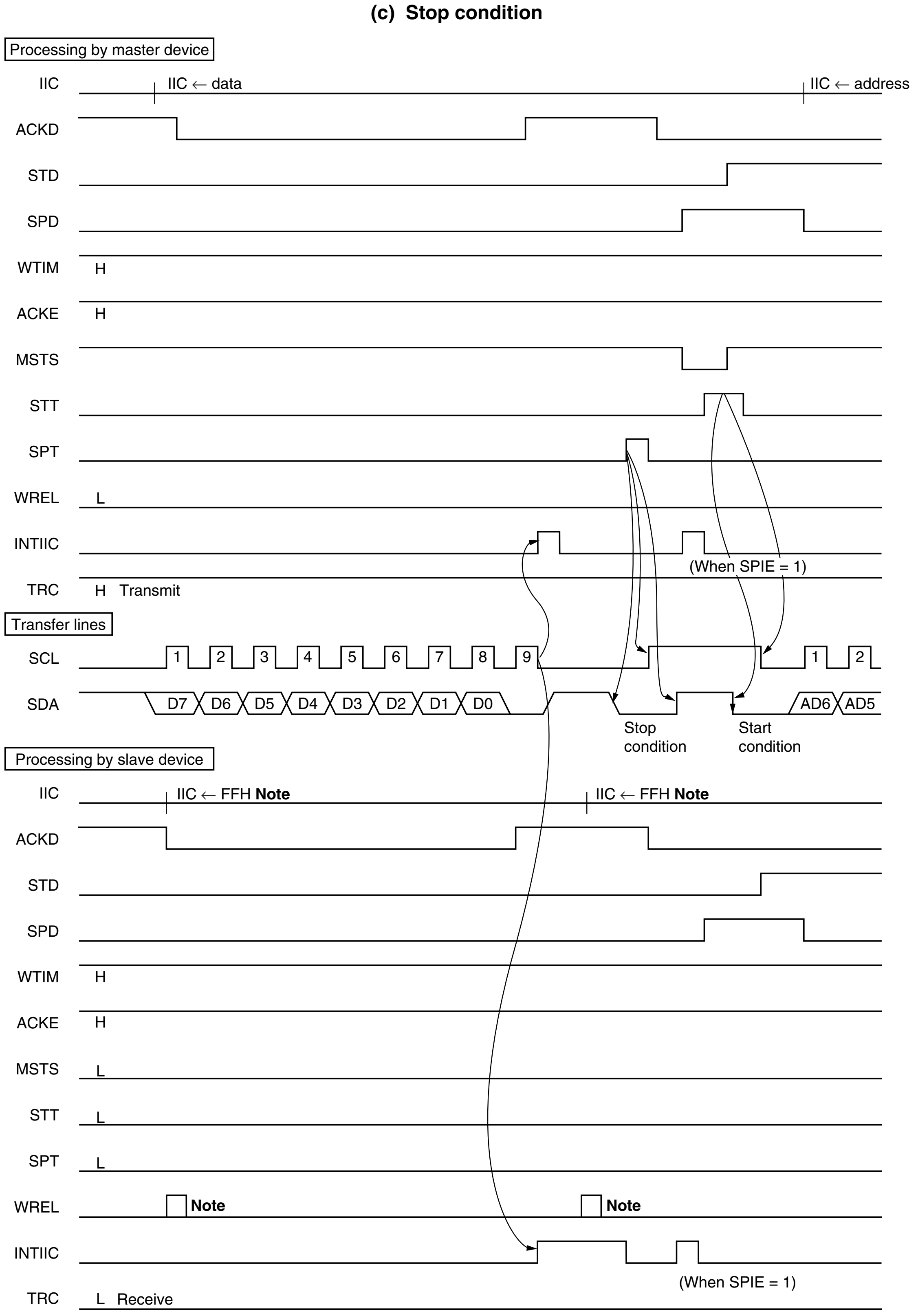
Note To cancel slave wait, write FFH to IIC or set WREL.

Figure 15-19. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)



Note To cancel slave wait, write FFH to IIC or set WREL.

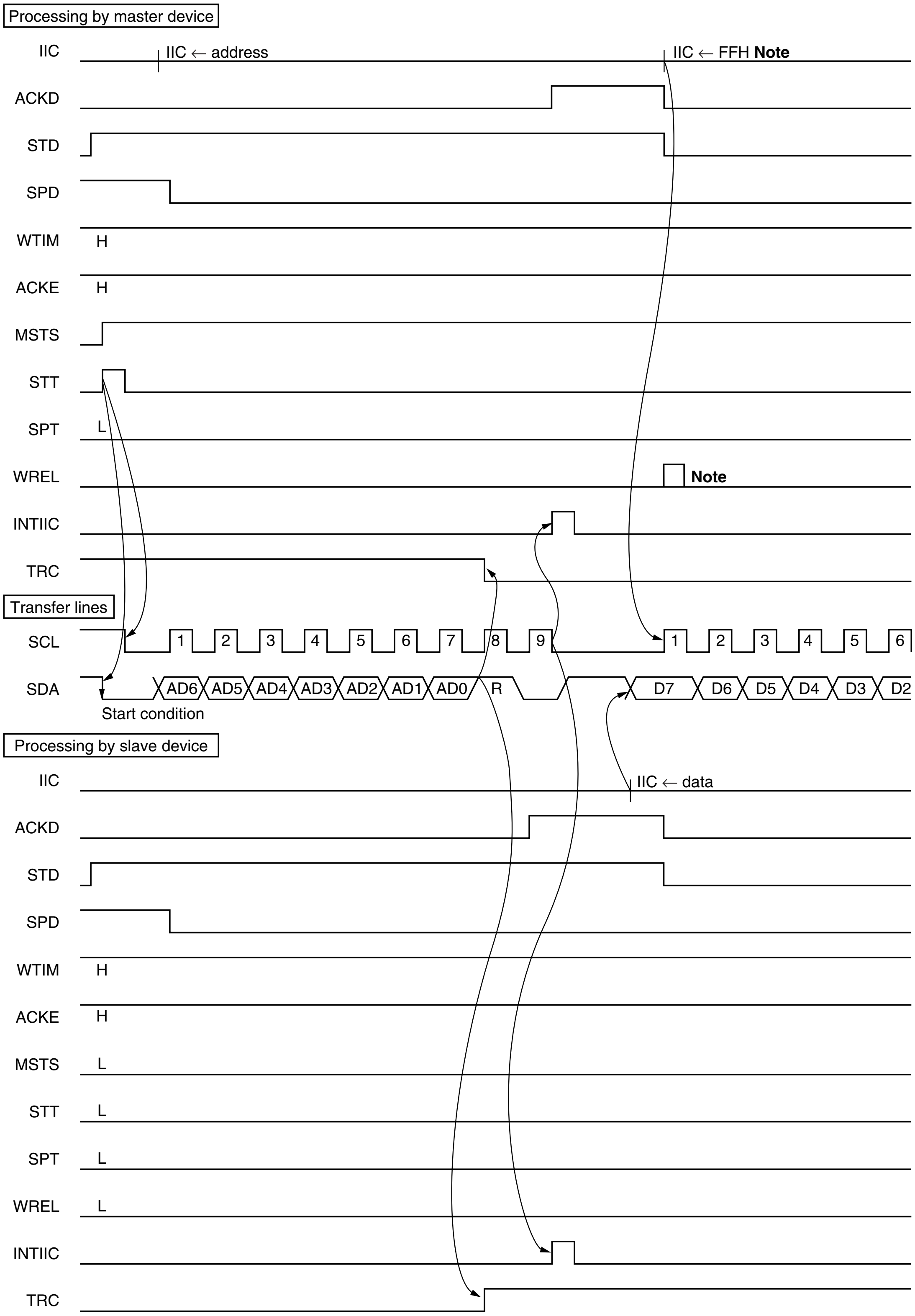
Figure 15-19. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)



Note To cancel slave wait, write FFH to IIC or set WREL.

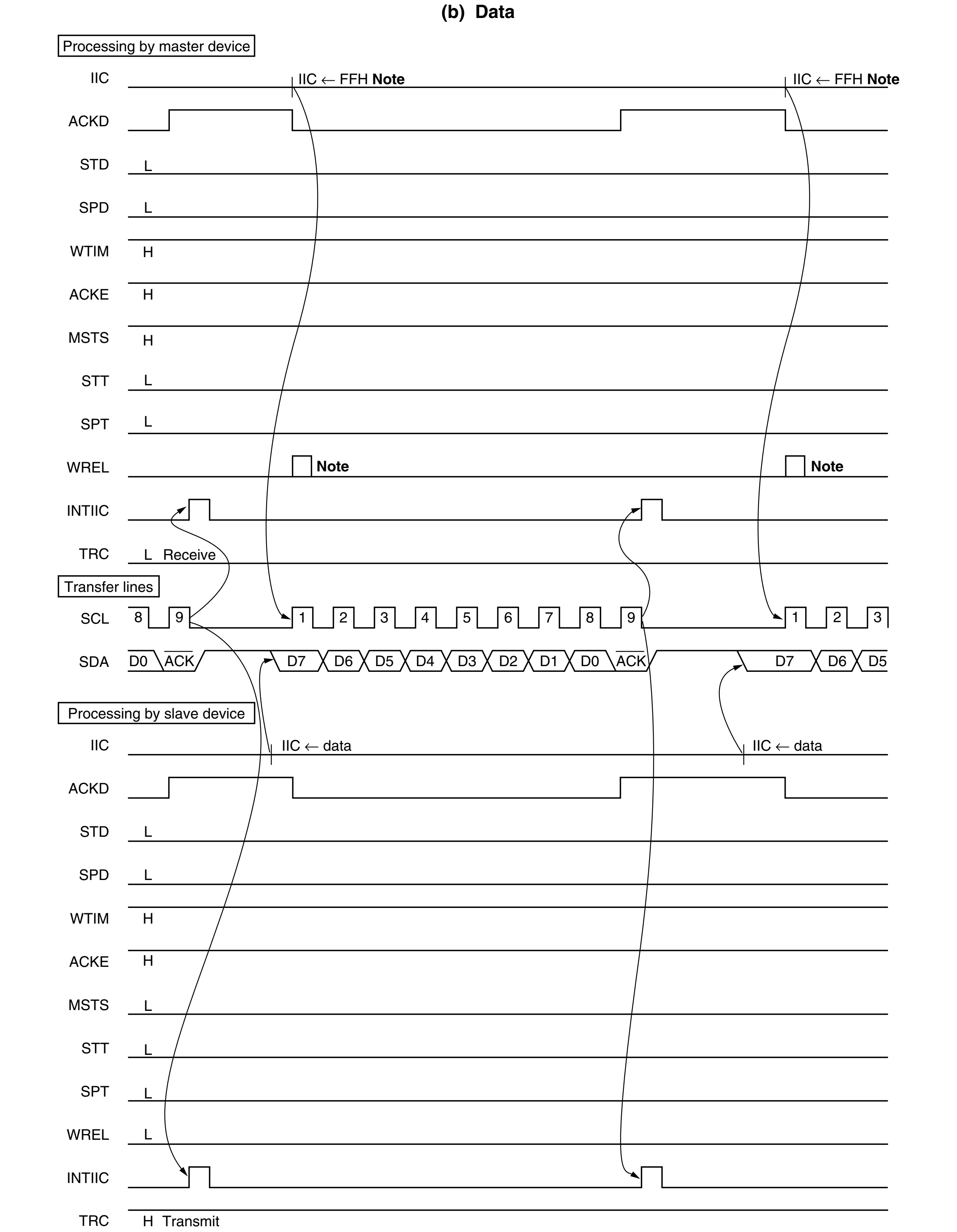
Figure 15-20. Example of Slave to Master Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address



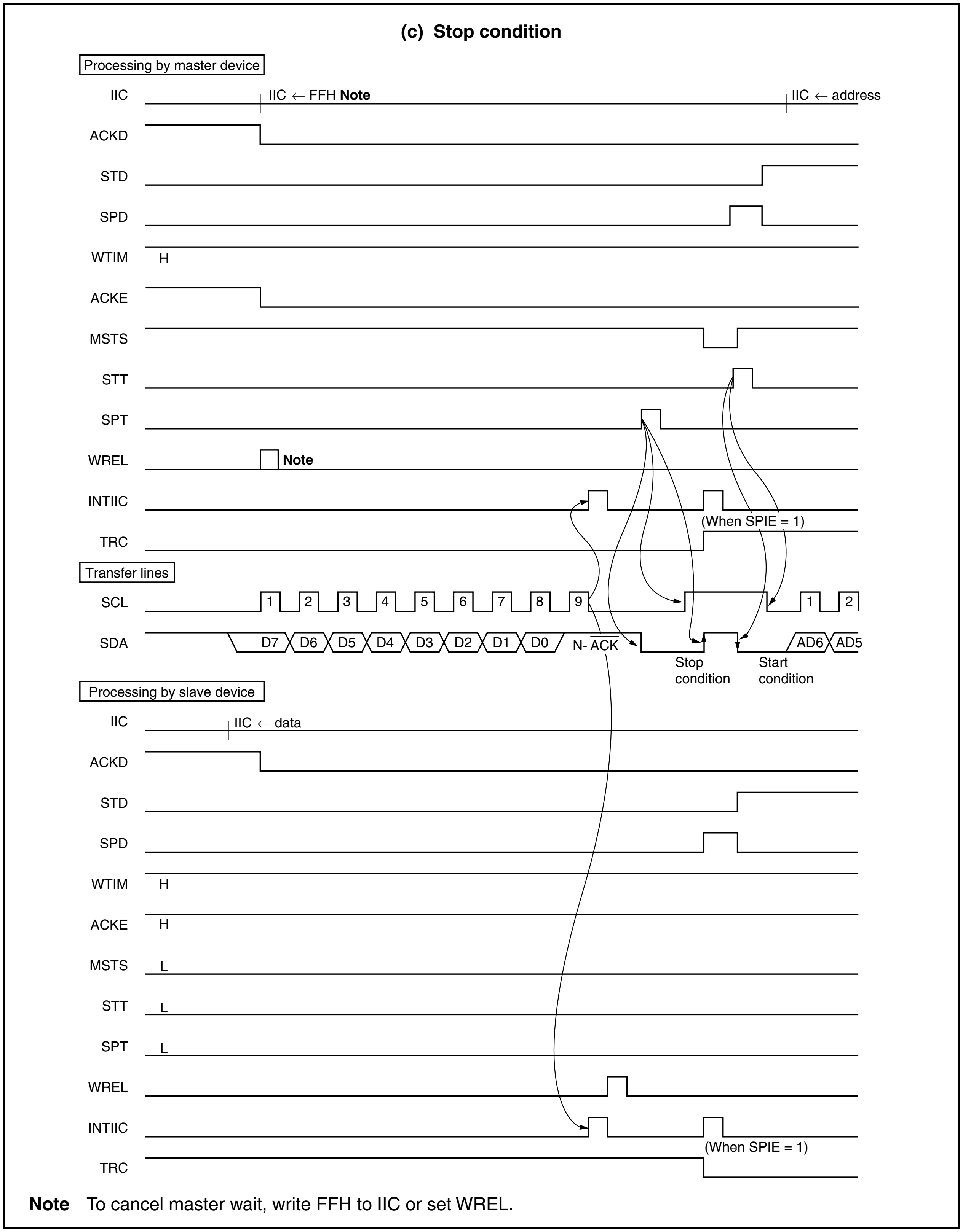
Note To cancel master wait, write FFH to IIC or set WREL.

Figure 15-20. Example of Slave to Master Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)



Note To cancel master wait, write FFH to IIC or set WREL.

Figure 15-20. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)



Note To cancel master wait, write FFH to IIC or set WREL.

CHAPTER 16 DMA FUNCTIONS (DMA CONTROLLER)

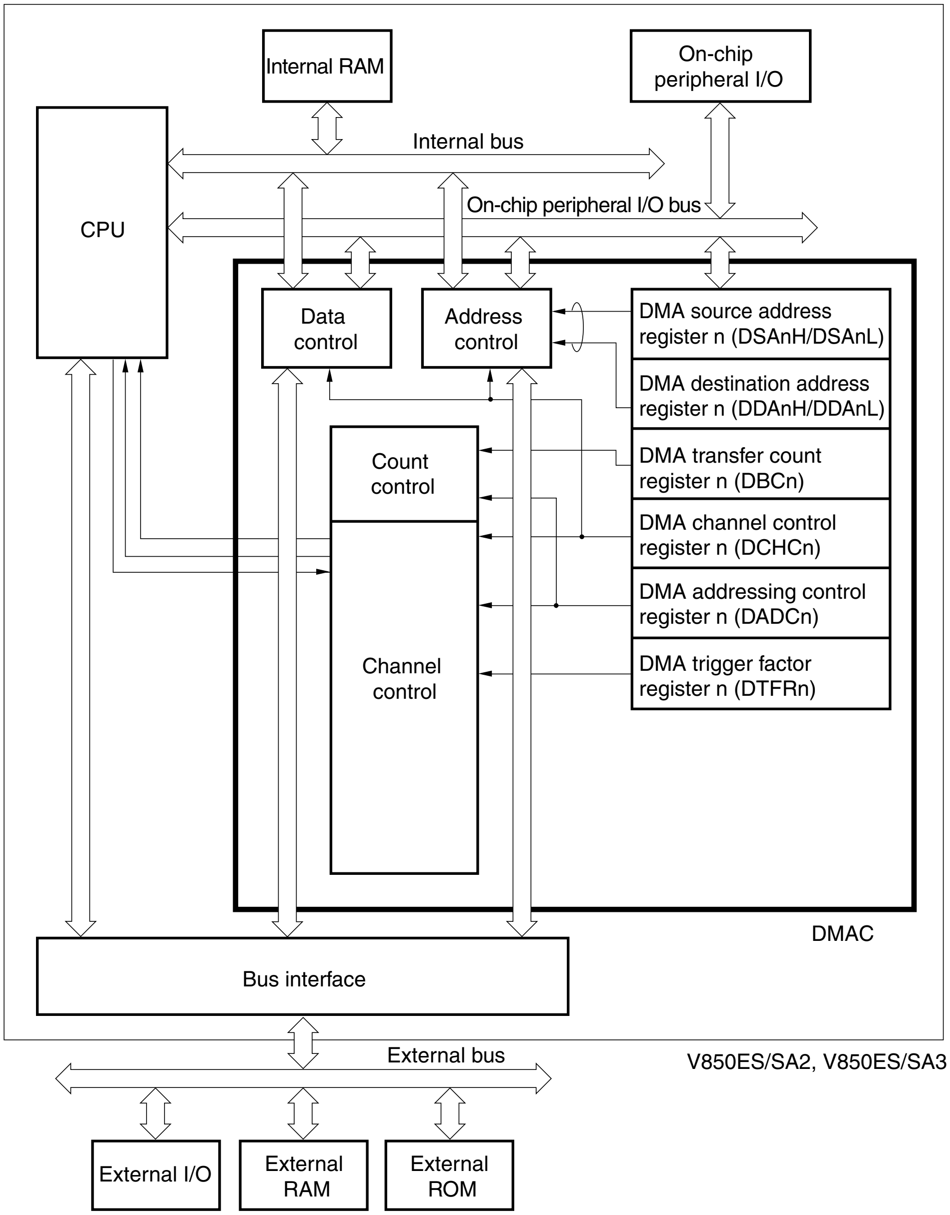
The V850ES/SA2 and V850ES/SA3 include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between memory and I/O, between memories, or between I/Os based on DMA requests issued by the on-chip peripheral I/O (serial interface, real-time pulse unit, and A/D converter), interrupts from external input pins, or software triggers (memory refers to internal RAM or external memory).

16.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Transfer type: Two-cycle transfer
- Transfer mode: Single transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer/counter, A/D converter) or interrupts from external input pin
  - Requests by software trigger
- Transfer objects
  - Internal RAM ↔ internal peripheral I/O
  - Peripheral I/O ↔ internal peripheral I/O
  - Internal RAM ↔ external memory
  - External memory ↔ internal peripheral I/O
  - External memory ↔ external memory

16.2 Configuration


Internal RAM
On-chip peripheral I/O
Internal bus
CPU
On-chip peripheral I/O bus
Data control
Address control
DMA source address register n (DSAnH/DSAnL)
DMA destination address register n (DDAnH/DDAnL)
Count control
DMA transfer count register n (DBCn)
DMA channel control register n (DCHCn)
DMA addressing control register n (DADCn)
Channel control
DMA trigger factor register n (DTFRn)
DMAC
Bus interface
External bus
V850ES/SA2, V850ES/SA3
External I/O
External RAM
External ROM

Remark n = 0 to 3

16.3 Control Registers

16.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)

These registers are used to set the DMA source addresses (28 bits each) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DSAnH and DSAnL.

(1) DMA source address registers 0H to 3H (DSA0H to DSA3H)

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DSA0H FFFFF082H, DSA1H FFFFF08AH,
DSA2H FFFFF092H, DSA3H FFFFF09AH

	15	14	13	12	11	10	9	8
DSAnH	IR	0	0	0	0	0	SA25	SA24

	7	6	5	4	3	2	1	0
	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16

(n = 0 to 3)

IR	DMA source address specification
0	External memory, on-chip peripheral I/O
1	Internal RAM

SA25 to SA16	Sets the DMA source addresses (A25 to A16). During DMA transfer, it stores the next DMA transfer source address.

(2) DMA source address registers 0L to 3L (DSA0L to DSA3L)

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DSA0L FFFFF080H, DSA1L FFFFF088H,
DSA2L FFFFF090H, DSA3L FFFFF098H

	15	14	13	12	11	10	9	8
DSAnL	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8

	7	6	5	4	3	2	1	0
	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

(n = 0 to 3)

SA15 to SA0	Sets the DMA source addresses (A15 to A0). During DMA transfer, it stores the next DMA transfer source address.

16.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)

These registers are used to set the DMA destination address (28 bits each) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

(1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH,
DDA2H FFFFF096H, DDA3H FFFFF09EH

	15	14	13	12	11	10	9	8
DDAnH	IR	0	0	0	0	0	DA25	DA24
	7	6	5	4	3	2	1	0
	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16

(n = 0 to 3)

IR	DMA destination address specification
0	External memory, on-chip peripheral I/O
1	Internal RAM

DA25 to DA16	Sets the DMA destination addresses (A25 to A16). During DMA transfer, it stores the next DMA transfer destination address.

(2) DMA destination address registers 0L to 3L (DDA0L to DDA3L)

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DDA0L FFFFF084H, DDA1L FFFFF08CH,
DDA2L FFFFF094H, DDA3L FFFFF09CH

	15	14	13	12	11	10	9	8
DDAnL	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
	7	6	5	4	3	2	1	0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

(n = 0 to 3)

DA15 to DA0	Sets the DMA destination addresses (A15 to A0). During DMA transfer, it stores the next DMA transfer destination address.

16.3.3 DMA byte count registers 0 to 3 (DBC0 to DBC3)

These 16-bit registers are used to set the byte transfer count for DMA channels n (n = 0 to 3). They store the remaining transfer count during DMA transfer.

These registers can be read or written in 16-bit units.

Remark If the DBCn register is read during DMA transfer after a terminal count has occurred without the register being overwritten, the value set immediately before the DMA transfer will be read out (0000H will not be read, even if DMA transfer has ended).

After reset: Undefined R/W Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H,
DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

	15	14	13	12	11	10	9	8
DBCn	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8

	7	6	5	4	3	2	1	0
	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

(n = 0 to 3)

BC15 to BC0	Byte transfer count setting or remaining byte transfer count during DMA transfer
0000H	Byte transfer count 1 or remaining byte transfer count
0001H	Byte transfer count 2 or remaining byte transfer count
:	:
FFFFH	Byte transfer count 65,536 (2^{16}) or remaining byte transfer count

16.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)

These 16-bit registers are used to control the DMA transfer mode for DMA channel n (n = 0 to 3). These registers cannot be accessed during DMA operation.

They can be read or written in 16-bit units.

After reset: 0000H R/W Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H,
DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

	15	14	13	12	11	10	9	8
DADCn	0	DS0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	SAD1	SAD0	DAD1	DAD0	0	0	0	0

(n = 0 to 3)

DS0	Setting of transfer data size for DMA transfer
0	8 bits
1	16 bits

SAD1	SAD0	Setting of count direction of the source address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DAD1	DAD0	Setting of count direction of the destination address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

16.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)

These 8-bit registers are used to control the DMA transfer operating mode for DMA channel n (n = 0 to 3).

These registers can be read or written in 8-bit or 1-bit units. (However, bit 7 is read-only and bits 1 and 2 are write-only. If bit 1 or 2 is read, the read value is always 0.)

After reset: 00H R/W Address: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H,
DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

	<7>	6	5	4	3	<2>	<1>	<0>
DCHCn (n = 0 to 3)	TCn[Note 1]	0	0	0	0	INITn[Note 2]	STGn[Note 2]	Enn

TCn	Status flag indicates whether DMA transfer through DMA channel n has ended or not
0	DMA transfer had not ended.
1	DMA transfer had ended.
It is set to 1 when DMA transfer ends and cleared (to 0) when it is read.	

INITn	When changing the DDAnH, DDAnL, DSAnL, DSAnH, or DBCn register before the number of transfers set by DBCn has finished, set this bit to 1 to initialize DMA.
Set the INIT bit to 1 when the Enn bit is 0.	

STGn	If this bit is set to 1 in the DMA transfer enable state (TCn bit = 0, Enn bit = 1), DMA transfer is started.

Enn	Setting of whether DMA transfer through DMA channel n is to be enabled or disabled
0	DMA transfer disabled
1	DMA transfer enabled
This bit is cleared to 0 when DMA transfer ends.	

Notes 1. The TCn bit is read-only.

2. The INITn and STGn bits are write-only.

Caution **Before generating a DMA transfer request by software, make sure that the TCn bit is set to 1 and then clear the TCn bit to 0.**

Remark If the completion of DMA transfer and the bit manipulation instruction for the DCHCn register conflict, the Enn bit may not be cleared.

16.3.6 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

These 8-bit registers are used to control the DMA transfer start trigger through interrupt requests from on-chip peripheral I/O.

The interrupt requests set with these registers serve as DMA transfer start factors.

These registers can be read or written in 8-bit or 1-bit units. However, only bit 7 (DFn) can be read/written in 1-bit units.

After reset: 00H R/W Address: DTFR0 FFFFF810H, DTFR1 FFFFF812H,
DTFR2 FFFFF814H, DTFR3 FFFFF816H

	<7>	6	5	4	3	2	1	0
DTFRn (n = 0 to 3)	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

DFn[Note]	Setting of interrupt source that serves as the DMA start factor
0	No DMA transfer request
1	DMA transfer request

Note The DFn bit is a write-only bit. Write 0 to this bit to clear a DMA transfer request if the interrupt that is specified as the cause of starting DMA transfer while DMA transfer is disabled.

Cautions 1. Be sure to stop DMA operation before making changes to DTFRn register settings.

2. An interrupt request input in a standby mode (IDLE or software STOP mode) cannot be used as a DMA transfer start factor.

3. For details of IFCn5 to IFCn0 bits, refer to Table 16-1 DMA Start Factor.

Remark n = 0 to 3

Table 16-1. DMA Start Factor

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
0	0	0	0	0	0	DMA request by interrupt disabled
0	0	0	0	0	1	INTWDTM
0	0	0	0	1	0	INTP0
0	0	0	0	1	1	INTP1
0	0	0	1	0	0	INTP2
0	0	0	1	0	1	INTP3
0	0	0	1	1	0	INTP4
0	0	0	1	1	1	INTP5
0	0	1	0	0	0	INTP6
0	0	1	0	0	1	INTRTC
0	0	1	0	1	0	INTCC00
0	0	1	0	1	1	INTCC01
0	0	1	1	0	0	INTOVF0
0	0	1	1	0	1	INTCC10
0	0	1	1	1	0	INTCC11
0	0	1	1	1	1	INTOVF1
0	1	0	0	0	0	INTTM2
0	1	0	0	0	1	INTTM3
0	1	0	0	1	0	INTTM4
0	1	0	0	1	1	INTTM5
0	1	0	1	0	0	INTCSI0
0	1	0	1	0	1	INTIIC
0	1	0	1	1	0	INTCSI1
0	1	0	1	1	1	INTSRE0
0	1	1	0	0	0	INTSR0
0	1	1	0	0	1	INTST0
0	1	1	0	1	0	INTCSI2
0	1	1	0	1	1	INTSRE1
0	1	1	1	0	0	INTSR1
0	1	1	1	0	1	INTST1
0	1	1	1	1	0	INTCSI3
0	1	1	1	1	1	INTCSI4
1	0	0	0	0	0	INTAD
1	0	0	0	0	1	INTOVF
1	0	0	0	1	0	INTBRG
1	0	0	1	1	1	Setting prohibited after this

Remark n = 0 to 3

16.4 DMA Bus States

16.4.1 Types of bus states

The DMAC bus states consist of the following 10 states.

(1) TI state

The TI state is an idle state, during which no access request is issued.

The DMA request signals are sampled at the rising edge of the CLKOUT signal.

(2) T0 state

DMA transfer ready state (state in which a DMA transfer request has been issued and the bus mastership is acquired for the first DMA transfer).

(3) T1R state

The bus enters the T1R state at the beginning of a read operation in the two-cycle transfer mode.

Address driving starts. After entering the T1R state, the bus invariably enters the T2R state.

(4) T1RI state

The T1RI state is a state in which the bus waits for the acknowledge signal corresponding to an external memory read request.

After entering the last T1RI state, the bus invariably enters the T2R state.

(5) T2R state

The T2R state corresponds to the last state of a read operation in the two-cycle transfer mode, or to a wait state.

In the last T2R state, read data is sampled. After entering the last T2R state, the bus invariably enters the T1W state.

(6) T2RI state

State in which the bus is ready for DMA transfer to on-chip peripheral I/O or internal RAM (state in which the bus mastership is acquired for DMA transfer to on-chip peripheral I/O or internal RAM).

After entering the last T2RI state, the bus invariably enters the T1W state.

(7) T1W state

The bus enters the T1W state at the beginning of a write operation in the two-cycle transfer mode.

Address driving starts. After entering the T1W state, the bus invariably enters the T2W state.

(8) T1WI state

State in which the bus waits for the acknowledge signal corresponding to an external memory write request.

After entering the last T1WI state, the bus invariably enters the T2W state.

(9) T2W state

The T2W state corresponds to the last state of a write operation in the two-cycle transfer mode, or to a wait state.

In the last T2W state, the write strobe signal is made inactive.

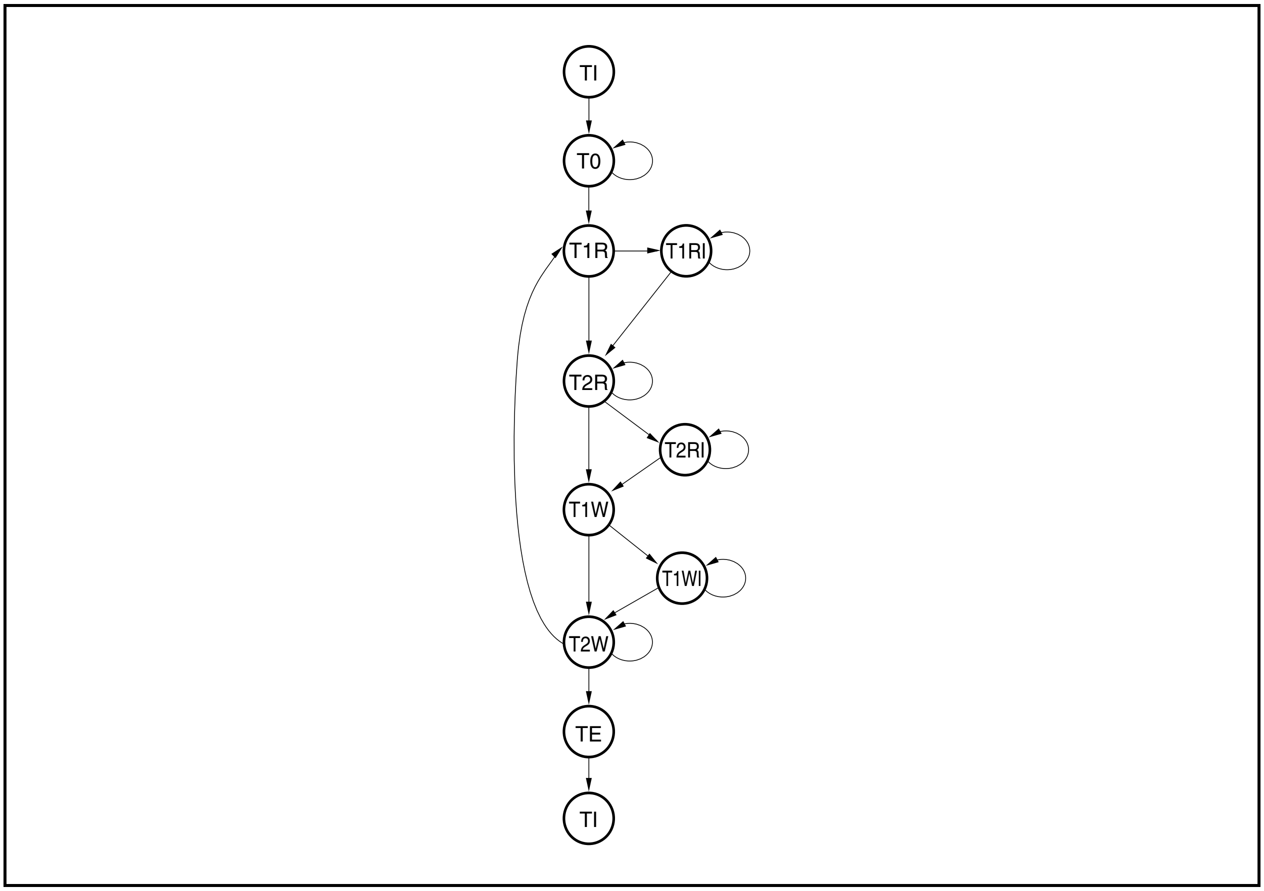
(10) TE state

The TE state corresponds to DMA transfer completion. The DMAC generates the internal DMA transfer completion signal and various internal signals are initialized (n = 0 to 3). After entering the TE state, the bus invariably enters the TI state.

16.4.2 DMAC bus cycle state transition

Except for the block transfer mode, each time the processing for a DMA transfer is completed, the bus mastership is released.

Figure 16-1. DMAC Bus Cycle (Two-Cycle Transfer) State Transition



16.5 Transfer Mode

16.5.1 Single transfer mode

In single transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again once. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

16.6 Transfer Types

16.6.1 Two-cycle transfer

In two-cycle transfer, data transfer is performed in two cycles, a read cycle (source to DMAC) and a write cycle (DMAC to destination).

In the first cycle, the source address is output and reading is performed from the source to the DMAC. In the second cycle, the destination address is output and writing is performed from the DMAC to the destination.

16.7 Transfer Object

16.7.1 Transfer type and transfer object

Table 13-2 shows the relationship between transfer type and transfer object (√: Transfer enabled, ×: Transfer disabled).

Table 16-2. Relationship Between Transfer Type and Transfer Object

		Destination				
		Internal ROM	On-Chip Peripheral I/O	External I/O	Internal RAM	External Memory
Source	On-chip peripheral I/O	×	√	√	√	√
	External I/O	×	√	√	√	√
	Internal RAM	×	√	√	×	√
	External memory	×	√	√	√	√
	Internal ROM	×	×	×	×	×

Caution The operation is not guaranteed for combinations of transfer destination and source marked with "×" in Table 16-2.

Remark During two-cycle 16-bit transfer, if the data bus width of the transfer source and that of the transfer destination are different, the operation becomes as follows.

In the case of transfer from a 16-bit bus to an 8-bit bus
A 16-bit read cycle is generated and then an 8-bit write cycle is generated twice.

In the case of transfer from an 8-bit bus to a 16-bit bus
An 8-bit read cycle is generated twice and then a 16-bit write cycle is generated.

16.7.2 External bus cycles during DMA transfer (two-cycle transfer)

The external bus cycles during DMA transfer (two-cycle transfer) are shown below.

Table 16-3. External Bus Cycles During DMA Transfer (Two-Cycle Transfer)

Transfer Object	External Bus Cycle	
On-chip peripheral I/O, internal RAM	None[Note]	–
External I/O	Yes	SRAM cycle
External memory	Yes	Memory access cycle set by the BCT register

Note Other external cycles such as a CPU-based bus cycle can be started.

16.8 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

These priorities are valid in the TI state only. In the block transfer mode, the channel used for transfer is never switched.

16.9 DMA Transfer Start Factors

There are two types of DMA transfer start factors, as shown below.

(1) Request from software

If the STGn, Enn, and TCn bits of the DCHCn register are set as follows, DMA transfer starts (n = 0 to 3).

- STGn bit = 1
- Enn bit = 1
- TCn bit = 0

(2) Request from on-chip peripheral I/O

If, when the Enn and TCn bits of the DCHCn register are set as shown below, an interrupt request is issued from the on-chip peripheral I/O that is set in the DTFRn register, DMA transfer starts (n = 0 to 3).

- Enn bit = 1
- TCn bit = 0

16.10 DMA Transfer End

16.10.1 DMA transfer end interrupt

When DMA transfer ends and the TCn bit of the DCHCn register is set to 1, a DMA transfer end interrupt (INTDMAn) is issued to the interrupt controller (INTC) (n = 0 to 3).

16.10.2 Terminal count output upon DMA transfer end

The terminal count signal becomes active for one clock during the last DMA transfer cycle.

⋆ 16.11 Precautions

(1) The transfer operation is not guaranteed if the source or the destination address exceeds the area of DMA objects (external memory, internal RAM, or peripheral I/O) during DMA transfer.

(2) DMA transfer of 16-bit bus width misaligned data is not supported.

(3) The time required to respond to a DMA request, and the minimum number of clocks required for DMA transfer are shown below.

Single transfer: DMA response time (<1>) + Transfer source memory access (<2>) + 1[Note 1] + Transfer destination memory access (<2>)

DMA Cycle		Minimum Number of Execution Clocks
<1> DMA request response time		4 clocks (MIN.) + Noise elimination time[Note 2]
<2> Memory access	External memory access	Depends on connected memory.
	Internal RAM access	2 clocks[Note 3]
	Peripheral I/O register access	3 clocks + Number of wait cycles specified by VSWC register[Note 4]

Notes 1. One clock is always inserted between a read cycle and a write cycle in DMA transfer.

2. If an external interrupt (INTPn) is specified as the trigger to start DMA transfer, noise elimination time is added (n = 0 to 7).

3. Two clocks are required for a DMA cycle.

4. More wait cycles are necessary for accessing a special internal peripheral I/O register (for details, refer to **3.4.8 (2)**).

(4) The CPU can access external memory, on-chip peripheral I/O, and internal RAM not undergoing DMA transfer. While data transfer among external memories or to and from I/O is being performed, the CPU can access internal RAM.
While data transfer is being executed between internal RAMs, the CPU can access external memory and peripheral I/O.

(5) Set the VSWC register to 00H or 01H. For details of the VSWC register, refer to **CHAPTER 5 BUS CONTROL FUNCTION**.

(6) Set the DSAn, DDAn, DBCn, and DADCn registers at any of the following timing other than during DMA operation.

- After reset and before the first DMA transfer starts
- After the channel has been initialized and before the first DMA transfer starts
- After DMA transfer is complete (the TCn bit of the DCHCn register is 1) and before the next DMA transfer start request is generated

(7) If the DSAn or DDAn register is read during DMA transfer, the value being updated may be read.

(8) DMA transfer in progress can be stopped by clearing the Enn bit of the DCHCn register to 0 during DMA transfer.

Follow either of the procedures below to clear the Enn bit of the DCHCn register to 0.

(a) To discard all transfers in progress and start DMA transfer from the beginning

<1> Stop generation of the DMA transfer start factor (stop the DMA operation).

<2> Make sure that the DFn bit of the DTFRn register is cleared to 0 (clear the bit if set).

<3> After making sure that the DFn bit of the DTFRn register is cleared to 0, clear the Enn bit of the DCHCn register to 0.

<4> Set the INITn bit of the DCHCn register (this operation initializes the transfer status).

(b) To interrupt and resume the transfer in progress

<1> Stop generation of the DMA transfer start factor (stop the DMA operation).

<2> Make sure that the DFn bit of the DTFRn register is cleared to 0 (if set, wait until the pending DMA transfer requests have been completed).

<3> After making sure that the DFn bit of the DTFRn register is cleared to 0, clear the Enn bit of the DCHCn register to 0 (this operation interrupts DMA transfer).

<4> Set the Enn bit of the DCHCn register to 1 to resume DMA transfer.

<5> Start the operation of the DMA transfer start factor that was stopped.

(9) The INIT bit is used to initialize DMA; however, if the setting of the INIT bit and the DMA transfer of another channel conflict, initialization processing is not performed. Set the INIT bit following either of the procedures below.

(a) To temporarily stop the transfers of all DMA channels

<1> Disable interrupts (DI).

<2> Read the Enn bits of the DCHCn registers of the DMA channels in use other than the channel to be initialized and transfer the read value to a general-purpose register.

<3> Clear the Enn bits of the DMA channels in use (including the channel to be initialized) to 0. For the last DMA channel, execute the clear instruction for the Enn bit twice[Note].

For example, execute the following instructions when channels 0, 1, and 2 are used.

- Clear E00 of the DCHC0 register to 0
- Clear E11 of the DCHC1 register to 0
- Clear E22 of the DCHC2 register to 0
- Clear E22 of the DCHC2 register to 0

<4> Set the INITn bit of the channel to be initialized to 1.

<5> Read the TCn bit of each of the channels not to be forcibly stopped. If the TCn and Enn bits of each channel not to be initialized, and the read TCn bit and the value read in <2> are 1 (the ANDed values are 1), clear the saved Enn bit to 0.

<6> Write the values of the Enn bits manipulated in step <5> above to the DCHCn register.

<7> Enable interrupts (EI).

Note If the transfer target (transfer source or transfer destination) is the internal RAM, execute the instruction three times.

Caution 1. Always perform step <5> to prevent the Enn bit of the channels successfully completed in <2> and <3> from being illegally set.

Caution 2. Do not configure programs that expect the TCn bit of the DCHCn register to be set to 1 by other than the above processing (the TCn bit of the DCHCn register is cleared to 0 after a read, so the bit is cleared when the instruction in <5> has been executed).

(b) To repetitively set the INITn bit until DMA is successfully initialized

<1> Clear the Enn bit of the DCHCn register of the channel to be initialized to 0.

<2> Clear the Enn bit of the DCHCn register of the channel to be initialized to 0.
If the transfer target of the channel to be initialized (transfer source or transfer destination) is the internal RAM, clear the Enn bit of the DCHCn register to 0 once again.

<3> Copy the number of initial transfers of the channel to be initialized to a general-purpose register.

<4> Set the INITn bit of the channel to be initialized to 1.

<5> Read the value of the DMA transfer count register (DBCn) of the channel to be initialized and compare it with the value copied in step <3>. If these values do not match, repeat steps <4> and <5>.

(10) When DMA transfer is complete, update is performed in the order of clearing the Enn bit of the DCHCn register to 0 and then setting the TCn bit to 1. Therefore, the status of Enn bit = TCn bit = 0 may be read depending on the timing of reading the DCHCn register.

(11) When the TCn bit of the DCHCn register is confirmed to be set to 1 after reading the bit, read the TCn bit three more times.

(12) The shortest interval for requesting DMA transfer to the same channel varies depending on the bus wait setting during the read cycle/write cycle, activation status of other channels, and external bus hold request.
Input a sufficiently long interval for a transfer request to the same channel so that the bus cycle can be completed in the system.

(13) Do not apply start factors (hardware trigger, software trigger) to the same channel at the same time. If two start factors are generated for one channel, only one or the other factor is valid. Therefore, the entire system may not operate normally.

(14) Before starting DMA transfer by manipulating the STGn bit (setting the STGn bit of the DCHCn register to 1), perform a second or subsequent transfer after making sure that the previous DMA transfer has been completed (check the DBCn register or the TCn bit of the DCHCn register).

(15) When performing DMA transfer whose target is the internal RAM while an instruction from the internal RAM is in progress, do not include bit manipulation instructions and misalign access in the program that executes instructions from the internal RAM.

16.11.1 Interrupt factors

DMA transfer is interrupted if a bus hold is issued.

If the factor (bus hold) interrupting DMA transfer disappears, DMA transfer promptly restarts.

CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850ES/SA2 and V850ES/SA3 are provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 38 to 40 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850ES/SA2 and V850ES/SA3 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

17.1 Features

○ Interrupts

- External interrupts: 8 sources (including NMI)
- Internal interrupts:

μPD703200, 703201, 70F3201:	30 sources
μPD703200Y, 703201Y, 70F3201Y:	31 sources
μPD703204, 70F3204:	31 sources
μPD703204Y, 70F3204Y:	32 sources

- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

○ Exceptions

- Software exceptions: 32 sources
- Exception trap: 2 sources (illegal opcode exception, debug trap)

Interrupt/exception sources are listed in Table 17-1.

Table 17-1. Interrupt Source List (1/2)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input	Pin	0000H	00000000H	Undefined	–
				WDT overflow[Note 1] (WDTRES)	WDT				
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
		–	INTWDT	WDT overflow	WDT	0020H	00000020H	nextPC	–
Software exception	Exception	–	TRAP0n[Note 2]	TRAP instruction	–	004nH[Note 2]	00000040H	nextPC	–
		–	TRAP1n[Note 2]	TRAP instruction	–	005nH[Note 2]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTWDTM	Interval timer overflow	WDT	0080H	00000080H	nextPC	WDTIC
		1	INTP0	INTP0 pin valid edge input	Pin	0090H	00000090H	nextPC	PIC0
		2	INTP1	INTP1 pin valid edge input	Pin	00A0H	000000A0H	nextPC	PIC1
		3	INTP2	INTP2 pin valid edge input	Pin	00B0H	000000B0H	nextPC	PIC2
		4	INTP3	INTP3 pin valid edge input	Pin	00C0H	000000C0H	nextPC	PIC3
		5	INTP4	INTP4 pin valid edge input	Pin	00D0H	000000D0H	nextPC	PIC4
		6	INTP5	INTP5 pin valid edge input	Pin	00E0H	000000E0H	nextPC	PIC5
		7	INTP6	INTP6 pin valid edge input	Pin	00F0H	000000F0H	nextPC	PIC6
		8	INTRTC	RTC interrupt	RTC	0100H	00000100H	nextPC	RTCIC
		9	INTCC00	CC00 capture trigger input/ TM0-CC00 match	TM0	0110H	00000110H	nextPC	CCIC00
		10	INTCC01	CC01 capture trigger input/ TM0-CC01 match	TM0	0120H	00000120H	nextPC	CCIC01
		11	INTOVF0	TM0 overflow	TM0	0130H	00000130H	nextPC	OVFIC0
		12	INTCC10	CC10 capture trigger input/ TM1-CC10 match	TM1	0140H	00000140H	nextPC	CCIC10
		13	INTCC11	CC11 capture trigger input/ TM1-CC11 match	TM1	0150H	00000150H	nextPC	CCIC11
		14	INTOVF1	TM1 overflow	TM1	0160H	00000160H	nextPC	OVFIC1
		15	INTTM2	TM2-CR2 match/ TM2 overflow	TM2	0170H	00000170H	nextPC	TMIC2
		16	INTTM3	TM3-CR3 match/ TM3 overflow	TM3	0180H	00000180H	nextPC	TMIC3
		17	INTTM4	TM4-CR4 match/ TM4 overflow	TM4	0190H	00000190H	nextPC	TMIC4
		18	INTTM5	TM5-CR5 match/ TM5 overflow	TM5	01A0H	000001AH	nextPC	TMIC5
		19	INTCSI0	CSI0 transfer completion	CSI0	01B0H	000001B0H	nextPC	CSIIC0
		20	INTIIC[Note 3]	I^2C transfer completion	I^2C	01C0H	000001C0H	nextPC	IICIC0
		21	INTCSI1	CSI1 transfer completion	CSI1	01D0H	000001D0H	nextPC	CSIIC1
		22	INTSRE0	UART0 reception error	UART0	01E0H	000001E0H	nextPC	SREIC0
		23	INTSR0	UART0 reception completion	UART0	01F0H	000001F0H	nextPC	SRIC0
		24	INTST0	UART0 transmission completion	UART0	0200H	00000200H	nextPC	STIC0
		25	INTCSI2	CSI2 transfer completion	CSI2	0210H	00000210H	nextPC	CSIIC2

Notes 1. Execute a system reset after interrupt servicing is finished. It is impossible to restore by executing the RETI instruction.

2. n = 0 to FH

3. μPD703200Y, 703201Y, 70F3201Y, 703204Y, 70F3204Y only

Table 17-1. Interrupt Source List (2/2)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	26	INTSRE1	UART1 reception error	UART1	0220H	00000220H	nextPC	SREIC1
		27	INTSR1	UART1 reception completion	UART1	0230H	00000230H	nextPC	SRIC1
		28	INTST1	UART1 transmission completion	UART1	0240H	00000240H	nextPC	STIC1
		29	INTCSI3	CSI3 transfer completion	CSI3	0250H	00000250H	nextPC	CSIIC3
		30	INTCSI4Note	CSI4 transfer completion	CSI4	0260H	00000260H	nextPC	CSIIC4
		31	INTAD	A/D conversion completion	ADC	0270H	00000270H	nextPC	ADIC
		32	INTDMA0	DMA0 transfer completion	DMA	0280H	00000280H	nextPC	DMAIC0
		33	INTDMA1	DMA1 transfer completion	DMA	0290H	00000290H	nextPC	DMAIC1
		34	INTDMA2	DMA2 transfer completion	DMA	02A0H	000002A0H	nextPC	DMAIC2
		35	INTDMA3	DMA3 transfer completion	DMA	02B0H	000002B0H	nextPC	DMAIC3
		36	INTROV	RTC overflow	RTC	02C0H	000002C0H	nextPC	ROVIC
		37	INTBRG	BRG match	BRG	02D0H	000002D0H	nextPC	BRGIC

Note V850ES/SA3 only

Remarks 1. Default Priority: The priority order when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC or FEPC when interrupt processing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value that starts the processing following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

17.2 Non-Maskable Interrupts

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. An NMI is not subject to priority control and takes precedence over all the other interrupts.

This product has the following two non-maskable interrupts.

- NMI pin input (NMI)
- Non-maskable interrupt request generated by overflow of watchdog timer (INTWDT)

The valid edge of the NMI pin can be selected from four types: “rising edge”, “falling edge”, “both edges”, and “no edge detection”.

The non-maskable interrupt generated by overflow of the watchdog timer (INTWDT) functions when the WDTM4 and WDTM3 bits of the watchdog timer mode register (WDTM) are set to “10”.

If two or more non-maskable interrupts occur at the same time, the interrupt with the higher priority is serviced, as follows (the interrupt with the lower priority is ignored).

INTWDT > NMI

If a new NMI or INTWDT request is issued while a NMI is being serviced, it is serviced as follows.

(1) If new NMI request is issued while NMI is being serviced

The new NMI request is held pending, regardless of the value of the NP bit of the program status word (PSW) in the CPU. The pending NMI interrupt is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

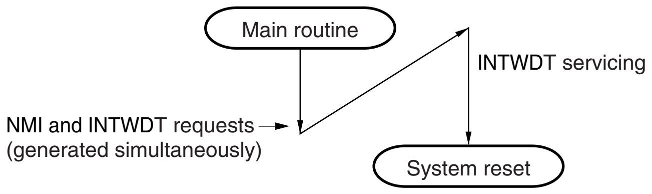
(2) If INTWDT request is issued while NMI is being serviced

The INTWDT request is held pending if the NP bit of the PSW remains set (1) while the NMI is being serviced. The pending INTWDT request is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

If the NP bit of PSW is cleared (0) while the NMI is being serviced, the newly generated INTWDT request is executed (the NMI servicing is stopped).

Caution If a non-maskable interrupt request is generated, the values of the PC and PSW are saved to the NMI status save registers (FEPC and FEPSW). At this time, execution can be returned by the RETI instruction only from an NMI. Execution cannot be returned while INTWDT is being serviced. Therefore, reset the system after the interrupt has been serviced.

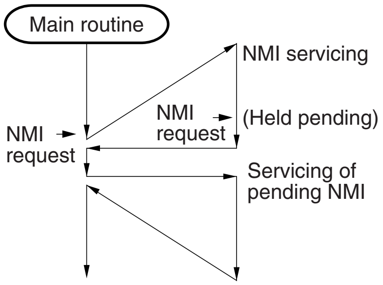
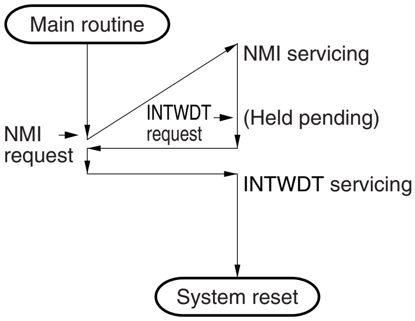
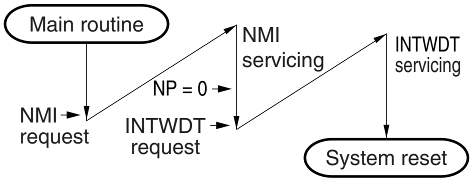
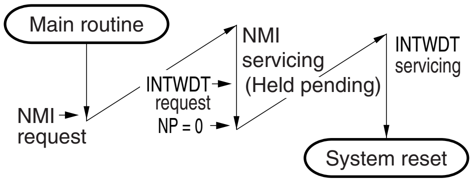
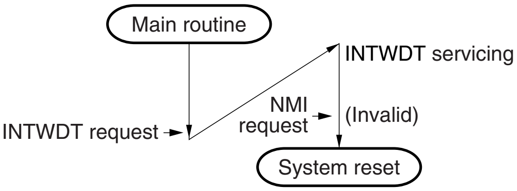
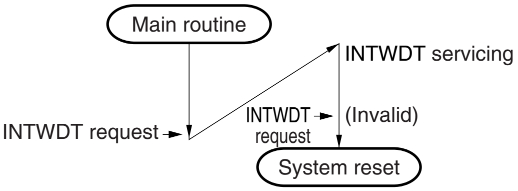
Figure 17-1. Non-Maskable Interrupt Request Acknowledgment Operation (1/2)



(a) NMI and INTWDT requests generated at the same time

Figure 17-1. Non-Maskable Interrupt Request Acknowledgment Operation (2/2)

(b) Non-maskable interrupt request generated during non-maskable interrupt servicing

Non-maskable interrupt being serviced	Non-maskable interrupt request generated during non-maskable interrupt servicing	
	NMI	INTWDT
NMI	• NMI request generated during NMI servicing  Main routine NMI servicing NMI request (Held pending) NMI request Servicing of pending NMI	• INTWDT request generated during NMI servicing (NP = 1 retained before INTWDT request)  Main routine NMI servicing INTWDT request (Held pending) NMI request INTWDT servicing System reset • INTWDT request generated during NMI servicing (NP = 0 set before INTWDT request)  Main routine NMI servicing INTWDT servicing NP = 0 NMI request INTWDT request System reset • INTWDT request generated during NMI servicing (NP = 0 set after INTWDT request)  Main routine NMI servicing INTWDT servicing INTWDT request (Held pending) NP = 0 NMI request System reset
INTWDT	• NMI request generated during INTWDT servicing  Main routine INTWDT servicing NMI request (Invalid) INTWDT request System reset	• INTWDT request generated during INTWDT servicing  Main routine INTWDT servicing INTWDT request (Invalid) INTWDT request System reset

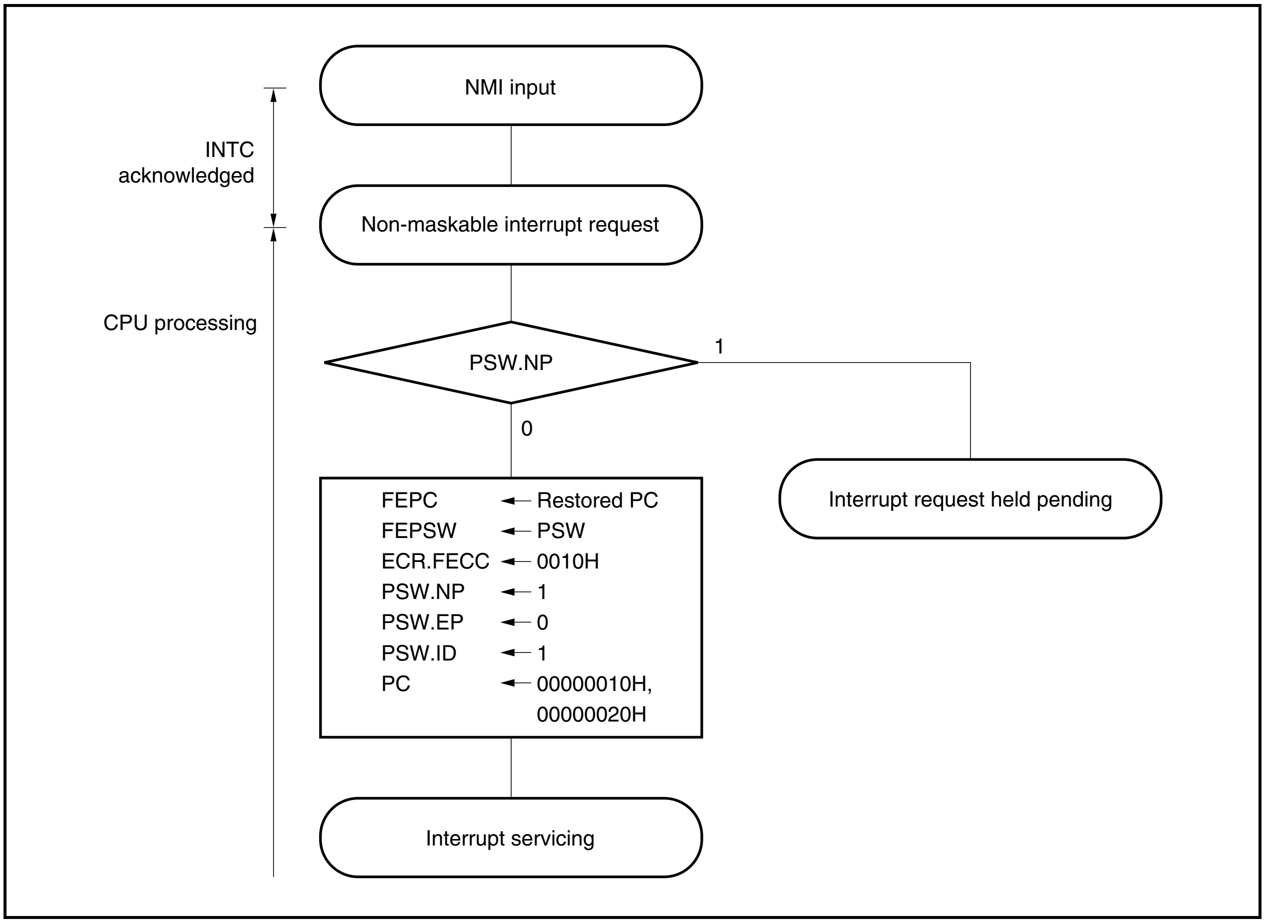
17.2.1 Operation

If a non-maskable interrupt is generated by NMI input, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to FEPC.
<2> Saves the current PSW to FEPSW.
<3> Writes exception code 0010H to the higher halfword (FECC) of ECR.
<4> Sets the NP and ID bits of the PSW and clears the EP bit.
<5> Sets the handler address (00000010H, 00000020H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The servicing configuration of a non-maskable interrupt is shown below.

Figure 17-2. Servicing Configuration of Non-Maskable Interrupt



17.2.2 Restore

(1) From NMI

Execution is restored from the NMI by the RETI instruction.

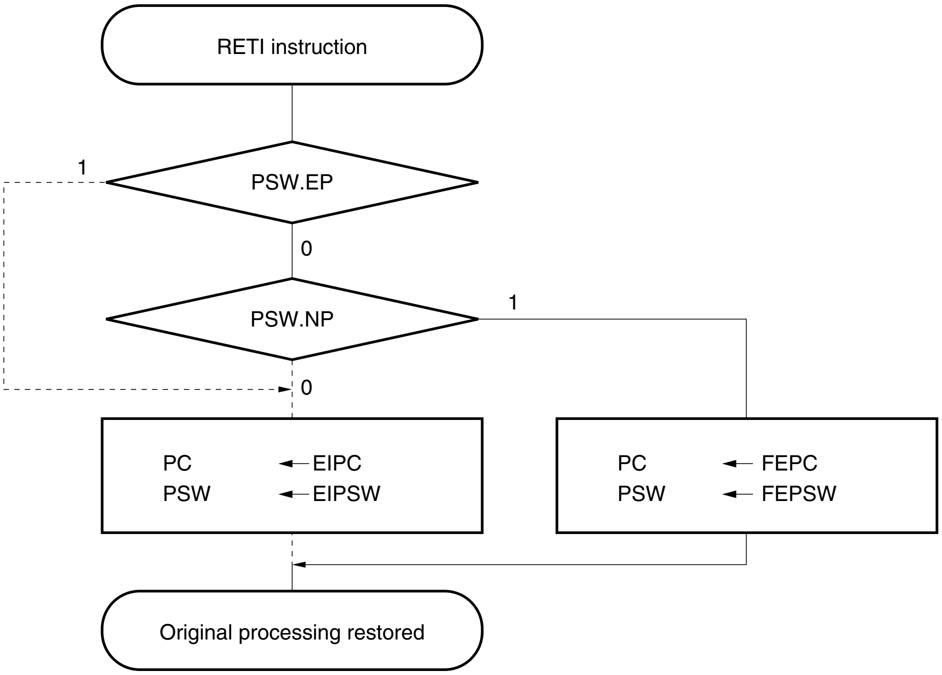
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from FEPC and FEPSW, respectively, because the EP bit of the PSW is 0 and the NP bit of the PSW is 1.

<2> Transfers control back to the address of the restored PC and PSW.

The following illustrates how the RETI instruction is processed.

Figure 17-3. RETI Instruction Processing



Caution When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

(2) From INTWDT

Execution cannot be returned from INTWDT by the RETI instruction. Execute a system reset after the interrupt has been serviced.

17.2.3 NP flag

The NP flag is a status flag that indicates that non-maskable interrupt servicing is under execution.

This flag is set when a non-maskable interrupt request has been acknowledged, and masks non-maskable interrupt requests to prohibit multiple interrupts from being acknowledged.

After reset: 00000020H

	31 8	7	6	5	4	3	2	1	0
PSW	0	**NP**	EP	ID	SAT	CY	OV	S	Z

NP	NMI interrupt servicing status
0	No NMI interrupt servicing
1	NMI interrupt currently being serviced

17.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850ES/SA2 and V850ES/SA3 have 38 to 40 maskable interrupt sources.

If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request has been acknowledged, the acknowledgment of other maskable interrupt requests is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To enable multiple interrupts, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

If the WDTM4 bit of the watchdog timer mode register (WDTM) is cleared to 0, the watchdog timer overflow interrupt functions as a maskable interrupt (INTWDTM).

17.3.1 Operation

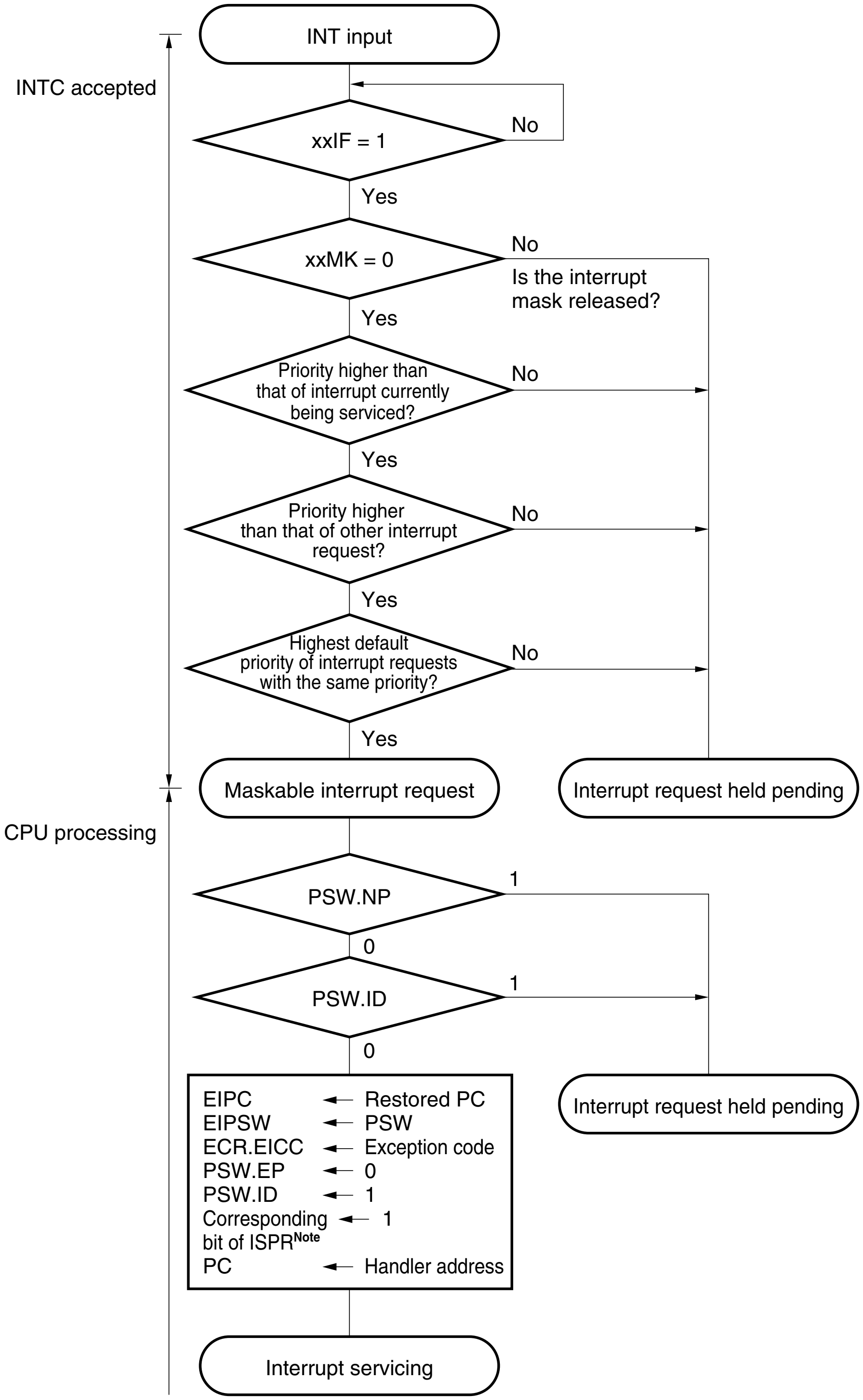
If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the ID bit of the PSW and clears the EP bit.
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request masked by INTC and the maskable interrupt request generated while another interrupt is being serviced (while PSW.NP = 1 or PSW.ID = 1) are held pending inside INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request if either the maskable interrupt is unmasked or PSW.NP and PSW.ID are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 17-4. Maskable Interrupt Servicing


INT input
INTC accepted
xxIF = 1
No
Yes
xxMK = 0
No
Is the interrupt mask released?
Yes
Priority higher than that of interrupt currently being serviced?
No
Yes
Priority higher than that of other interrupt request?
No
Yes
Highest default priority of interrupt requests with the same priority?
No
Yes
Maskable interrupt request
Interrupt request held pending
CPU processing
PSW.NP
1
0
PSW.ID
1
0
EIPC ← Restored PC
EIPSW ← PSW
ECR.EICC ← Exception code
PSW.EP ← 0
PSW.ID ← 1
Corresponding bit of ISPRNote ← 1
PC ← Handler address
Interrupt request held pending
Interrupt servicing

Note For the ISPR register, see **17.3.6 In-service priority register (ISPR)**.

The INT input masked by the interrupt controllers and the INT input that occurs while another interrupt is being serviced (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the interrupt controller. In such case, if the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 as set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt servicing.

17.3.2 Restore

Recovery from maskable interrupt servicing is carried out by the RETI instruction.

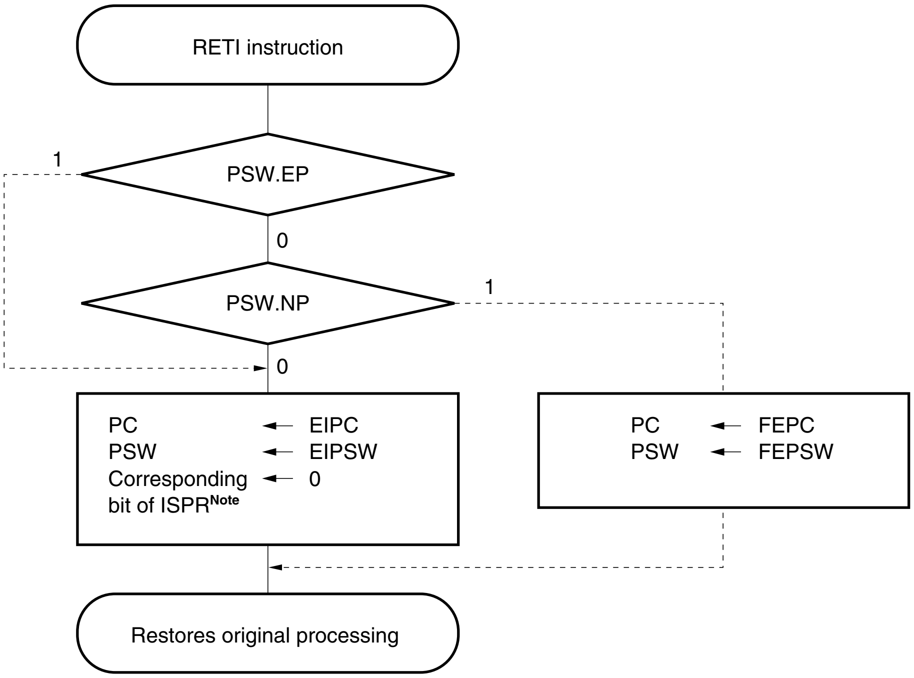
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.

<2> Transfers control to the address of the restored PC and PSW.

The following illustrates the processing of the RETI instruction.

Figure 17-5. RETI Instruction Processing



Note For the ISPR register, see **17.3.6 In-service priority register (ISPR)**.

Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

17.3.3 Priorities of maskable interrupts

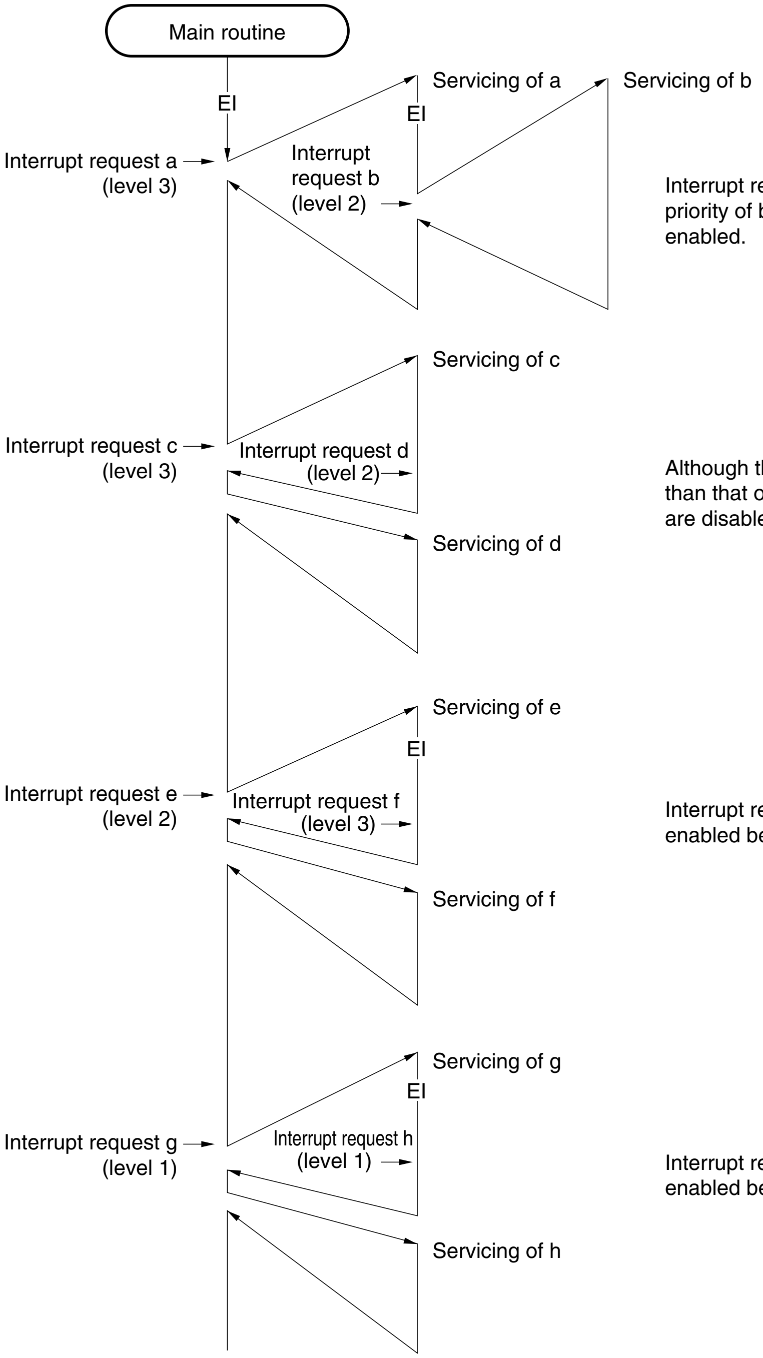
The V850ES/SA2 and V850ES/SA3 provide multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to **Table 17-1 Interrupt Source List**. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of PSW is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

Remark xx: Identification name of each peripheral unit (AD, BRG, CC, CSI, DMA, IIC, OVF, P, ROV, RTC, SRE, ST, TM, WDT)
n: Peripheral unit number (None or 0 to 3).

Figure 17-6. Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (1/2)



Interrupt request b is acknowledged because the priority of b is higher than that of a and interrupts are enabled.

Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.

Interrupt request f is held pending even if interrupts are enabled because its priority is lower than that of e.

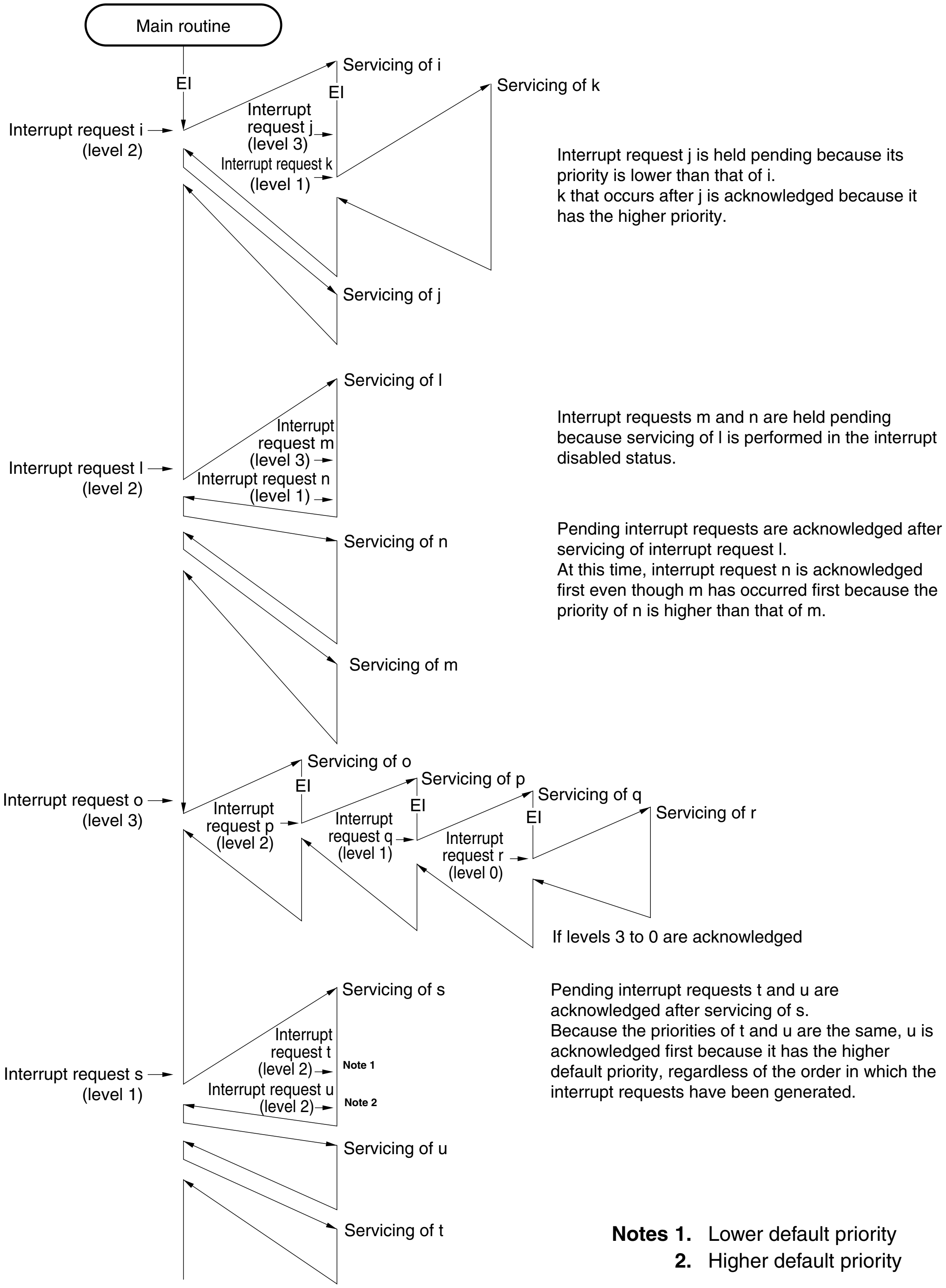
Interrupt request h is held pending even if interrupts are enabled because its priority is the same as that of g.

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 17-6. Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (2/2)


Main routine
EI
Servicing of i
EI
Servicing of k
Interrupt request i (level 2)
Interrupt request j (level 3)
Interrupt request k (level 1)
Servicing of j
Servicing of l
Interrupt request m (level 3)
Interrupt request l (level 2)
Interrupt request n (level 1)
Servicing of n
Servicing of m
Servicing of o
EI
Servicing of p
EI
Servicing of q
EI
Servicing of r
Interrupt request o (level 3)
Interrupt request p (level 2)
Interrupt request q (level 1)
Interrupt request r (level 0)
Servicing of s
Interrupt request t (level 2)
Note 1
Interrupt request s (level 1)
Interrupt request u (level 2)
Note 2
Servicing of u
Servicing of t

Interrupt request j is held pending because its priority is lower than that of i.
k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because servicing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after servicing of interrupt request l.
At this time, interrupt request n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

If levels 3 to 0 are acknowledged

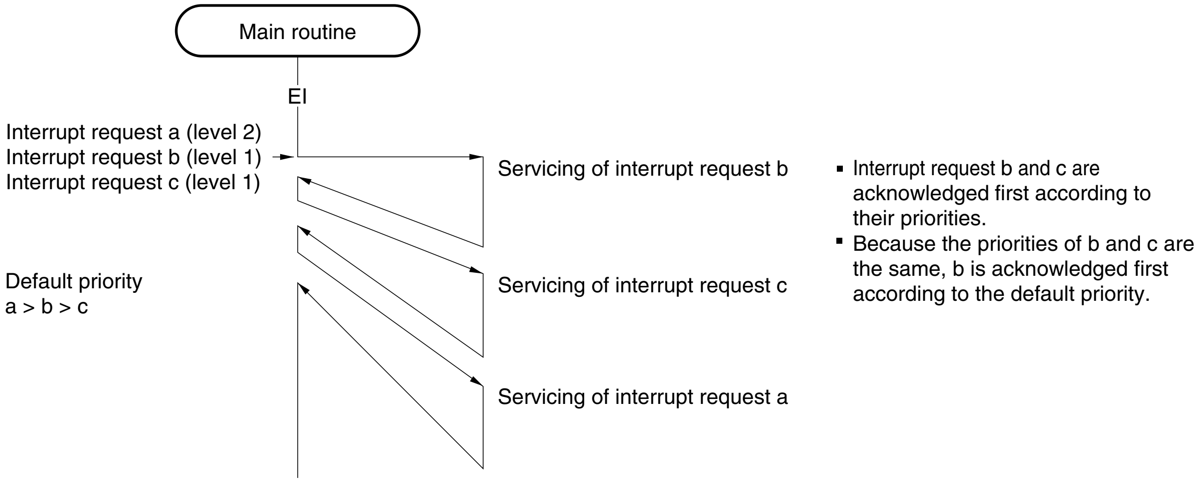
Pending interrupt requests t and u are acknowledged after servicing of s.
Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.

Notes 1. Lower default priority

2. Higher default priority

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 17-7. Example of Servicing Interrupt Requests Simultaneously Generated


Main routine
EI
Interrupt request a (level 2)
Interrupt request b (level 1)
Interrupt request c (level 1)
Default priority
a > b > c
Servicing of interrupt request b
Servicing of interrupt request c
Servicing of interrupt request a
▪ Interrupt request b and c are acknowledged first according to their priorities.
▪ Because the priorities of b and c are the same, b is acknowledged first according to the default priority.

Caution **To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.**

Remarks 1. a to c in the figure are the temporary names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

17.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each interrupt request (maskable interrupt) and sets the control conditions for each maskable interrupt request.

This register can be read or written in 8-bit or 1-bit units.

This register is set to 47H after reset.

★ **Caution Read the xxIFn bit of the xxICn register with interrupts disabled (DI). If the xxIFn bit is read with interrupts enabled (EI), a normal value may not be read when the timing of interrupt acknowledgment and reading of the bit conflict.**

After reset: 47H   R/W   Address: FFFFF110H to FFFFF15AH

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag[Note]
0	Interrupt request not issued
1	Interrupt request issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxIFn is reset automatically by the hardware if an interrupt request is acknowledged.

Remark xx: Identification name of each peripheral unit (OV, P00 to P03, P10 to P13, CM, DMA, CSI, SE, SR, ST, AD)

n: Peripheral unit number (None or 0 to 3).

The addresses and bits of the interrupt control registers are as follows.

Table 17-2. Interrupt Control Register (xxICn)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	WDTIC	WDTIF	WDTMK	0	0	0	WDTPR2	WDTPR1	WDTPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	RTCIC	RTCIF	RTCMK	0	0	0	RTCR2	RTCR1	RTCR0
FFFFF122H	CCIC00	CCIF00	CCMK00	0	0	0	CCR002	CCR001	CCR000
FFFFF124H	CCIC01	CCIF01	CCMK01	0	0	0	CCR012	CCR011	CCR010
FFFFF126H	OVFIC0	OVFIF0	OVFMK0	0	0	0	OVFR02	OVFR01	OVFR00
FFFFF128H	CCIC10	CCIF10	CCMK10	0	0	0	CCR102	CCR101	CCR100
FFFFF12AH	CCIC11	CCIF11	CCMK11	0	0	0	CCR112	CCR111	CCR110
FFFFF12CH	OVFIC1	OVFIF1	OVFMK1	0	0	0	OVFR12	OVFR11	OVFR10
FFFFF12EH	TMIC2	TMF2	TMMK2	0	0	0	TMPR22	TMPR21	TMPR20
FFFFF130H	TMIC3	TMF3	TMMK3	0	0	0	TMPR32	TMPR31	TMPR30
FFFFF132H	TMIC4	TMF4	TMMK4	0	0	0	TMPR42	TMPR41	TMPR40
FFFFF134H	TMIC5	TMF5	TMMK5	0	0	0	TMPR52	TMPR51	TMPR50
FFFFF136H	CSIIC0	CSIIF0	CSIMK0	0	0	0	CSIPR02	CSIPR01	CSIPR00
FFFFF138H	IICIC[Note 1]	IICIF	IICMK	0	0	0	IICPR2	IICPR1	IICPR0
FFFFF13AH	CSIIC1	CSIIF1	CSIMK1	0	0	0	CSIPR12	CSIPR11	CSIPR10
FFFFF13CH	SREIC0	SREIF0	SREMK0	0	0	0	SREPR02	SREPR01	SREPR00
FFFFF13EH	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF140H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF142H	CSIIC2	CSIIF2	CSIMK2	0	0	0	CSIPR22	CSIPR21	CSIPR20
FFFFF144H	SREIC1	SREIF1	SREMK1	0	0	0	SREPR12	SREPR11	SREPR10
FFFFF146H	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF148H	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF14AH	CSIIC3	CSIIF3	CSIMK3	0	0	0	CSIPR32	CSIPR31	CSIPR30
FFFFF14CH	CSIIC4[Note 2]	CSIIF4	CSIMK4	0	0	0	CSIPR42	CSIPR41	CSIPR40
FFFFF14EH	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF150H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF152H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF154H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF156H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF158H	ROVIC	ROVIF	ROVMK	0	0	0	ROVPR2	ROVPR1	ROVPR0
FFFFF15AH	BRGIC	BRGIF	BRGMK	0	0	0	BRGPR2	BRGPR1	BRGPR0

Notes 1. μPD703200Y, 703201Y, 703204Y, 70F3201Y, 70F30204Y only

2. V850ES/SA3 only

17.3.5 Interrupt mask registers 0 to 2 (IMR0 to IMR2)

These registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR2 registers is equivalent to the xxMKn bit of the xxICn register.

The IMRm register can be read or written in 16-bit units (m = 0 to 2).

If the higher 8 bits of the IMRm register are used as an IMRmH register and the lower 8 bits as an IMRmL register, these registers can be read or written in 8-bit or 1-bit units (m = 0 to 2).

Bits 15 to 6 of the IMR2 register (bits 7 to 0 of the IMR2H register and bits 7 and 6 of the IMR2L register) are fixed to 1. If these bits are not 1, the operation cannot be guaranteed.

These registers are set to FFFFH after reset.

Caution The device file defines the xxMKn bit of the xxICn register as a reserved word. If a bit is manipulated using the name of xxMKn, the contents of the xxICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

After reset: FFFFH R/W Address: FFFFF104H

	15	14	13	12	11	10	9	8
IMR2	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
	1	1	BRGMK	OVFMK	DMAMK3	DMAMK2	DMAMK1	DMAMK0

After reset: FFFFH R/W Address: FFFFF102H

	15	14	13	12	11	10	9	8
IMR1	ADMK	CSIMK4[Note]	CSIMK3	STMK1	SRMK1	SREMK1	CSIMK2	STMK0
	7	6	5	4	3	2	1	0
	SRMK0	SREMK0	CSIMK1	IICMK	CSIMK0	TMMK5	TMMK4	TMMK3

After reset: FFFFH R/W Address: FFFFF100H

	15	14	13	12	11	10	9	8
IMR0	TMMK2	OVFMK1	CCMK11	CCMK10	OVFMK0	CCMK01	CCMK00	RTCMK
	7	6	5	4	3	2	1	0
	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDTMK

xxMKn	Interrupt mask flag setting
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note This bit is valid only for the V850ES/SA3. In the V850EA/SA2, be sure to set this bit to 1.

Remark xx: Identification name of each peripheral unit (AD, BRG, CC, CSI, DMA, IIC, OVF, D, ROV, RTC, SRE, ST, TM, WDT).
n: Peripheral unit number (None, or 0 to 3)

17.3.6 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt request is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request having the highest priority is automatically cleared to 0 by hardware. However, it is not cleared to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

★ **Caution When an interrupt is acknowledged while the ISPR register is being read with interrupts enabled (EI), the value of the ISPR register after the bit has been set to 1 as a result of interrupt acknowledgment may be read. To correctly read the values in the ISPR register before interrupt acknowledgment, read the register with interrupts disabled (DI).**

After reset: 00H R Address: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently acknowledged
0	Interrupt request with priority n not acknowledged
1	Interrupt request with priority n acknowledged

Remark n = 0 to 7 (priority level)

17.3.7 ID flag

This flag controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests. An interrupt disable flag (ID) is incorporated, which is assigned to the PSW.

After reset: 00000020H

	31 to 8	7	6	5	4	3	2	1	0
PSW	0	NP	EP	ID	SAT	CY	OV	S	Z

ID	Maskable interrupt servicing specification[Note]
0	Maskable interrupt request acknowledgment enabled
1	Maskable interrupt request acknowledgment disabled (pending)

Note Interrupt disable flag (ID) function

This bit is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupt requests and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware.

The interrupt request generated during the acknowledgment disabled period (ID = 1) is acknowledged when the xxIFn bit of xxICn is set to 1, and the ID flag is reset to 0.

17.3.8 Watchdog timer mode register (WDTM)

This register is a special register and can be written only in a specific sequence. To generate a maskable interrupt (INTWDT), clear the WDTM4 bit of this register to 0.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset (for details, refer to **CHAPTER 10 WATCHDOG TIMER FUNCTIONS**).

After reset: 00H R/W Address: FFFFF6C2H

	<7>	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Watchdog timer operation mode selection[Note 1]
0	Stops counting.
1	Clears counter and starts counting.

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	0	Interval timer mode (Maskable interrupt INTWDTM occurs if overflow occurs.)
0	1	
1	0	Watchdog timer mode 1 (Non-maskable interrupt INTWDT occurs if overflow occurs.)
1	1	Watchdog timer mode 2 (Reset operation WDTRES is started if overflow occurs.)

17.4 Noise Elimination at External Interrupt Request Input Pins

The external interrupt request input pins (NMI, INTP0 to INTP6) incorporate noise eliminators that eliminate noise via analog delay. Therefore, unless the levels input to the NMI and INTP0 to INTP6 pins are retained for a certain length of time, the input cannot be detected as an edge. An edge is detected a certain time later.

An external interrupt request input pin can also be used to cancel software STOP mode.

17.4.1 Edge detection function of external interrupt request input pins

The valid edge of an external interrupt request input pin can be selected from "rising edge", "falling edge", "both edges", and "no edge detection".

Use rising edge specification register 0 (INTR0) and falling edge specification register 0 (INTF0) to specify the valid edges of non-maskable interrupts (NMI) and maskable interrupts (INTP0 to INTP4). These registers can be read or written in 8-bit or 1-bit units.

The NMI and INTP0 to INTP4 pins are set to "no edge detection" after reset; therefore, an interrupt request is not acknowledged unless the valid edge is enabled using the INTF0 and INTR0 registers (the ports function as normal ports).

Use rising edge specification register 9 (INTR9) and falling edge specification register 9 (INTF9) to specify the valid edges of maskable interrupts (INTP5 and INTP6).

The INTP5 and INTP6 pins are set to "no edge detection" after reset; therefore, an interrupt request is not acknowledged unless the valid edge is enabled using the INTF9 and INTR9 registers (the ports function as normal ports).

When using an external interrupt request input pin as an I/O port, set the valid edge of the NMI or INTP0 to INTP6 pins to "no edge detection".

(1) External interrupt rising edge specification register 0 (INTR0)

This is an 8-bit register that specifies detection of the rising edge of the NMI and INTP0 to INTP4 pins.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF0n and INTR0n to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC20H

	7	6	5	4	3	2	1	0
INTR0	0	0	INTR05	INTR04	INTR03	INTR02	INTR01	INTR00
			INTP4	INTP3	INTP2	INTP1	INTP0	NMI

Remark For how to specify a valid edge, refer to **Table 17-3**.

(2) External interrupt falling edge specification register 0 (INTF0)

This is an 8-bit register that specifies detection of the falling edge of the NMI and INTP0 to INTP4 pins.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF0n and INTR0n to 0, and then set the port mode.

After reset: 00H R/W Address: FFFFFC00H

	7	6	5	4	3	2	1	0
INTF0	0	0	INTF05	INTF04	INTF03	INTF02	INTF01	INTF00
			INTP4	INTP3	INTP2	INTP1	INTP0	NMI

Remark For how to specify a valid edge, refer to **Table 17-3**.

Table 17-3. Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification (n = 0 to 5)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 0: Control of NMI pin
n = 1 to 5: Control of INTP0 to INTP4 pins

(3) External interrupt rising edge specification register 9 (INTR9)

This is an 8-bit register that specifies detection of the rising edge of the INTP5 and INTP6 pins.

This register can be read or written in 16-bit units. When the lower 8 bits of the INTR9 register are used as INTR9L register, however, it can be read or written in 8-bit or 1-bit units.

This register is cleared to 0000H after reset.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF9n and INTR9n to 0, and then set the port mode.

After reset: 0000H R/W Address: INTR9: FFFFFC32H, INTR9L: FFFFFC32H

	15	14	13	12	11	10	9	8
INTR9	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(INTR9L)	0	0	0	0	INTR93	INTR92	0	0
					INTP6	INTP5		

Remark For how to specify a valid edge, refer to **Table 17-4**.

(4) External interrupt falling edge specification register 9 (INTF9)

This is an 8-bit register that specifies detection of the falling edge of the INTP5 and INTP6 pins.

This register can be read or written in 16-bit units. When the lower 8 bits of the INTF9 register are used as INTF9L register, however, it can be read or written in 8-bit or 1-bit units.

This register is cleared to 0000H after reset.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear INTF9n and INTR9n to 0, and then set the port mode.

After reset: 0000H R/W Address: INTF9: FFFFFC12H, INTF9L: FFFFFC12H

	15	14	13	12	11	10	9	8
INTF9	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(INTF9L)	0	0	0	0	INTF93	INTF92	0	0
					INTP6	INTP5		

Remark For how to specify a valid edge, refer to **Table 17-4**.

Table 17-4. Valid Edge Specification

INTF9n	INTR9n	Valid Edge Specification (n = 2, 3)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Remark n = 2, 3: Control of INTP5 and INTP6 pins

17.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

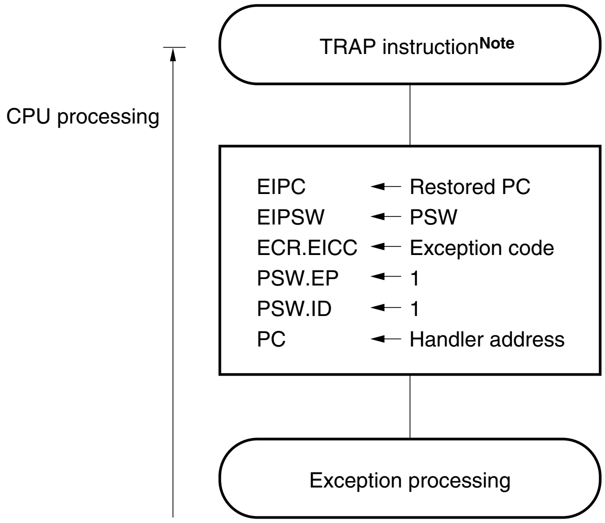
17.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the EP and ID bits of the PSW.
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

The following illustrates the processing of a software exception.

Figure 17-8. Software Exception Processing



Note TRAP instruction format: TRAP vector (the vector is a value from 0 to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 0 to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

17.5.2 Restore

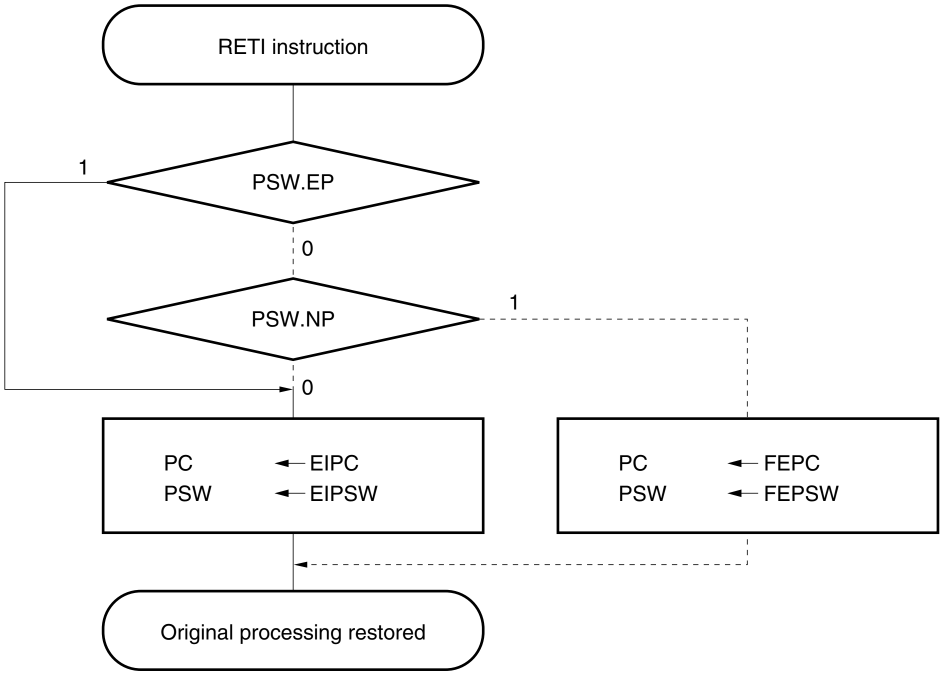
Recovery from software exception processing is carried out by the RETI instruction.

By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 1.
<2> Transfers control to the address of the restored PC and PSW.

The following illustrates the processing of the RETI instruction.

Figure 17-9. RETI Instruction Processing



Caution **When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.**

Remark The solid line shows the CPU processing flow.

17.5.3 EP flag

The EP flag is bit 6 of the PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

After reset: 00000020H

	31 8	7	6	5	4	3	2	1	0
PSW	0	NP	**EP**	ID	SAT	CY	OV	S	Z

EP	Exception processing status
0	Exception processing not in progress.
1	Exception processing in progress.

17.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/SA2 and V850ES/SA3, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

17.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.

15 11	10 5	4 0	31 27	26 23	22	16
X X X X X	1 1 1 1 1 1	X X X X X	X X X X X	0 1 1 1 to 1 1 1 1	X X X X X X	0

×: Arbitrary

Caution Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used.

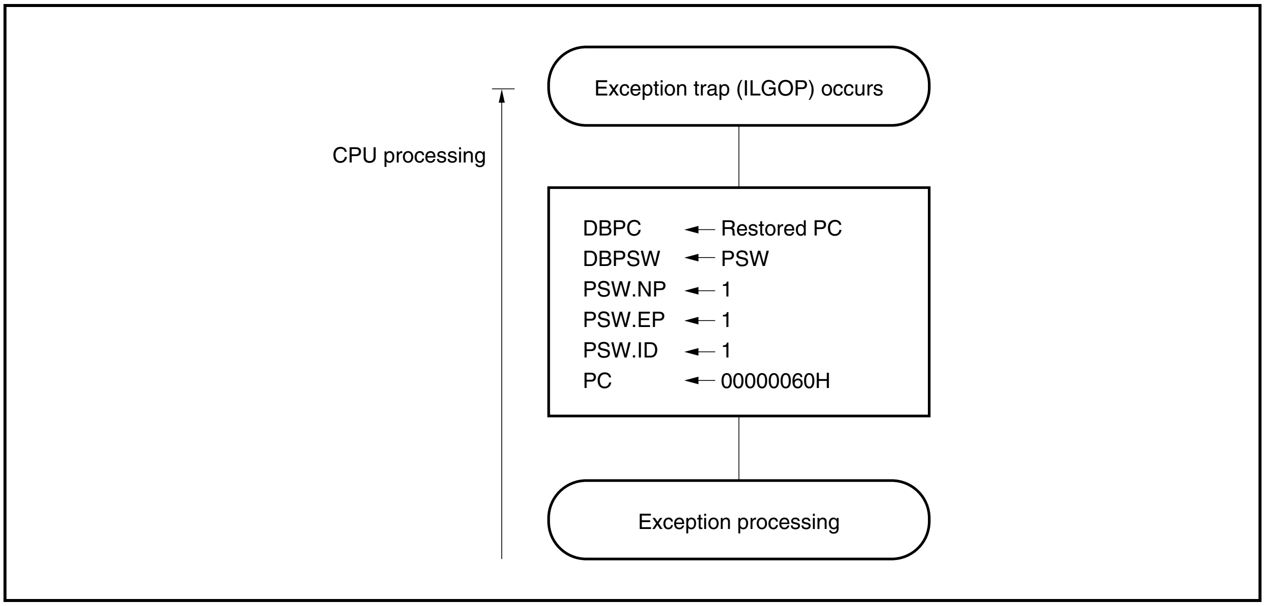
(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the NP, EP, and ID bits of the PSW.
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

The following illustrates the processing of the exception trap.

Figure 17-10. Exception Trap Processing



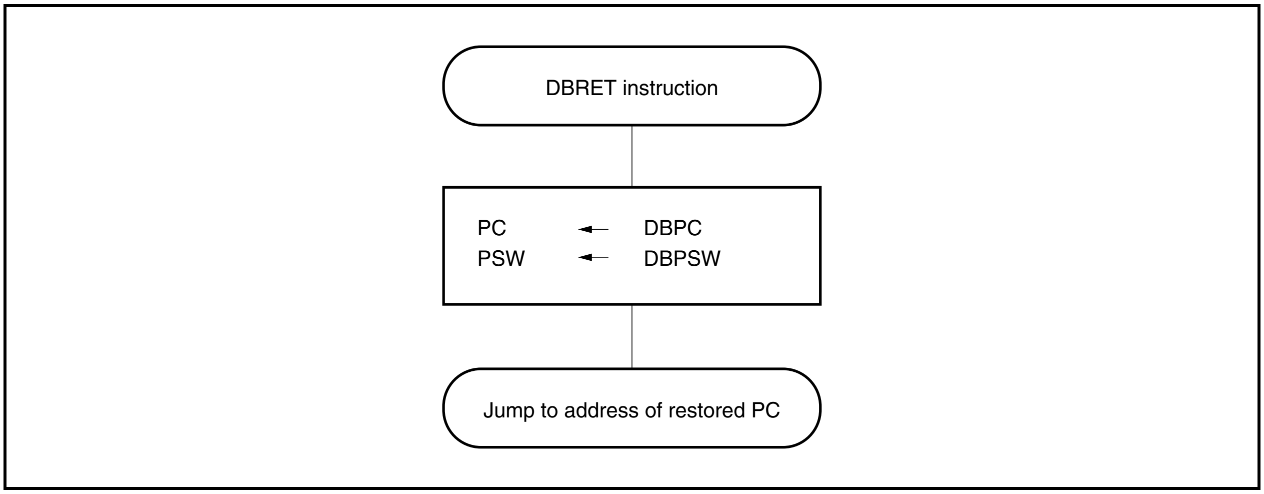
(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

The following illustrates the restore processing from an exception trap.

Figure 17-11. Restore Processing from Exception Trap



★

17.6.2 Debug trap

A debug trap is an exception that occurs upon execution of the DBTRAP instruction and that can be acknowledged at all times.

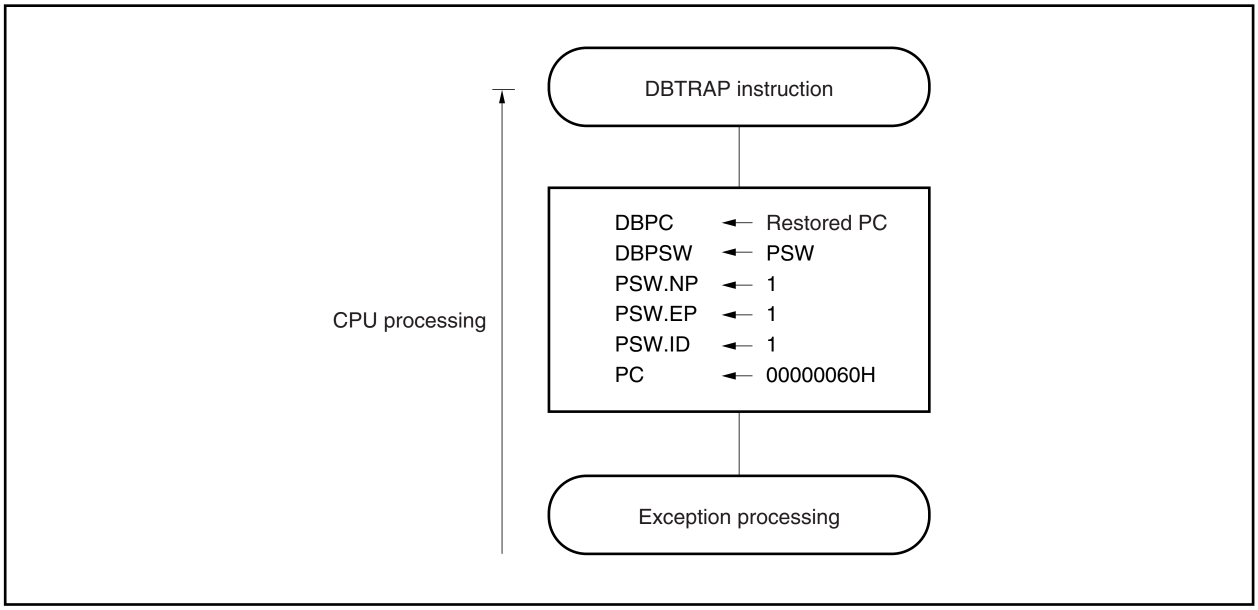
When a debug trap occurs, the CPU performs the following processing.

(1) Operation

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the NP, EP, and ID bits of the PSW.
<4> Sets the handler address (00000060H) for the debug trap routine to the PC and transfers control.

The following illustrates the debug trap processing flow.

Figure 17-12. Debug Trap Processing



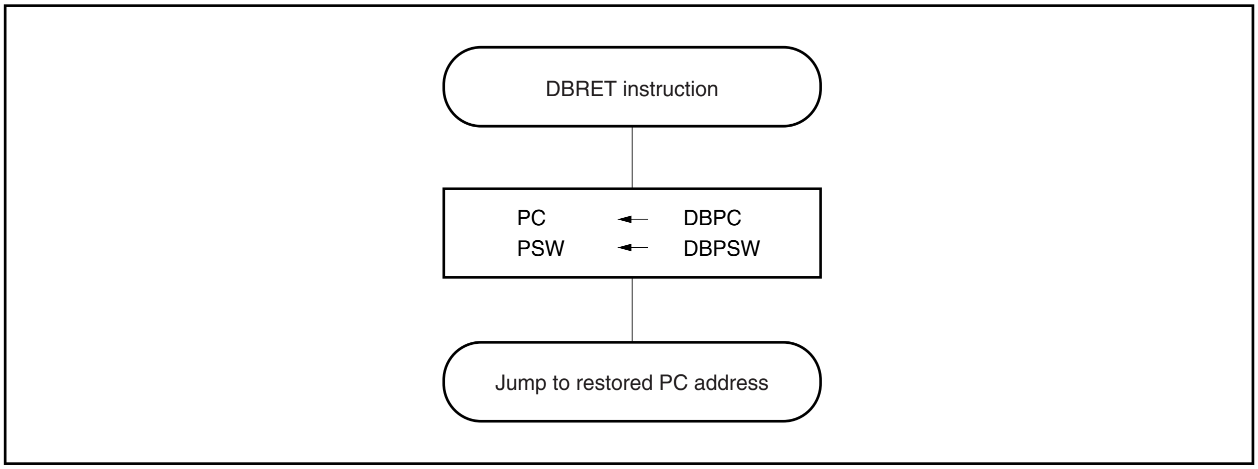
(2) Restore

Execution is restored from debug trap processing by the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the loaded address of the restored PC and PSW.

The following illustrates the processing flow for restore from debug trap processing.

Figure 17-13. Processing Flow for Restore from Debug Trap

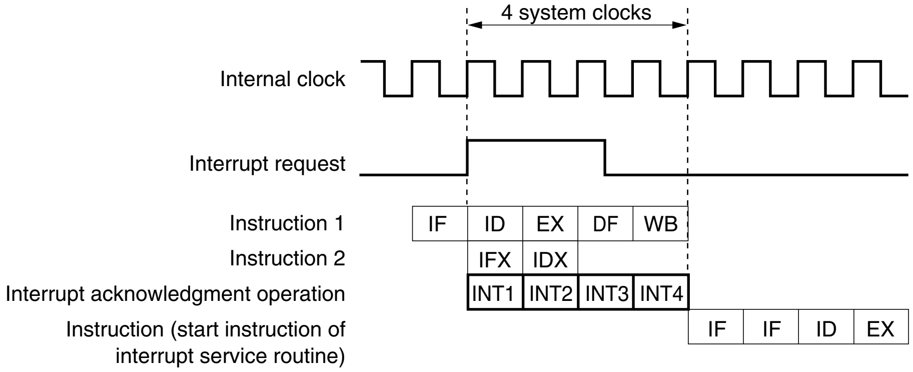


17.7 Interrupt Acknowledge Time of CPU

Except the following cases, the interrupt acknowledge time of the CPU is 4 clocks minimum. To input interrupt requests successively, input the next interrupt at least 4 clocks after the preceding interrupt.

- In software/hardware STOP mode
- When the external bus is accessed
- When interrupt request non-sampling instructions are successively executed (Refer to **17.8 Periods in Which Interrupts Are Not Acknowledged by CPU**.)
- When the interrupt control register is accessed
- When a peripheral I/O register is accessed

Figure 17-14. Pipeline Operation at Interrupt Request Acknowledgment (Outline)



Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt acknowledge time (internal system clock)			Condition
	Internal interrupt	External interrupt	
Minimum	4	4 + Analog delay time	The following cases are exceptions. • In IDLE/software STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession Access to interrupt control register • Access to peripheral I/O register
Maximum	6	6 + Analog delay time	

17.8 Periods in Which Interrupts Are Not Acknowledged by CPU

An interrupt is acknowledged by the CPU while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the command register (PRCMD)
- The load, store, or bit manipulation instructions for the following interrupt-related registers.

  Interrupt control register (xxICn), interrupt mask registers 0 to 2 (IMR0 to IMR2), in-service priority register (ISPR)

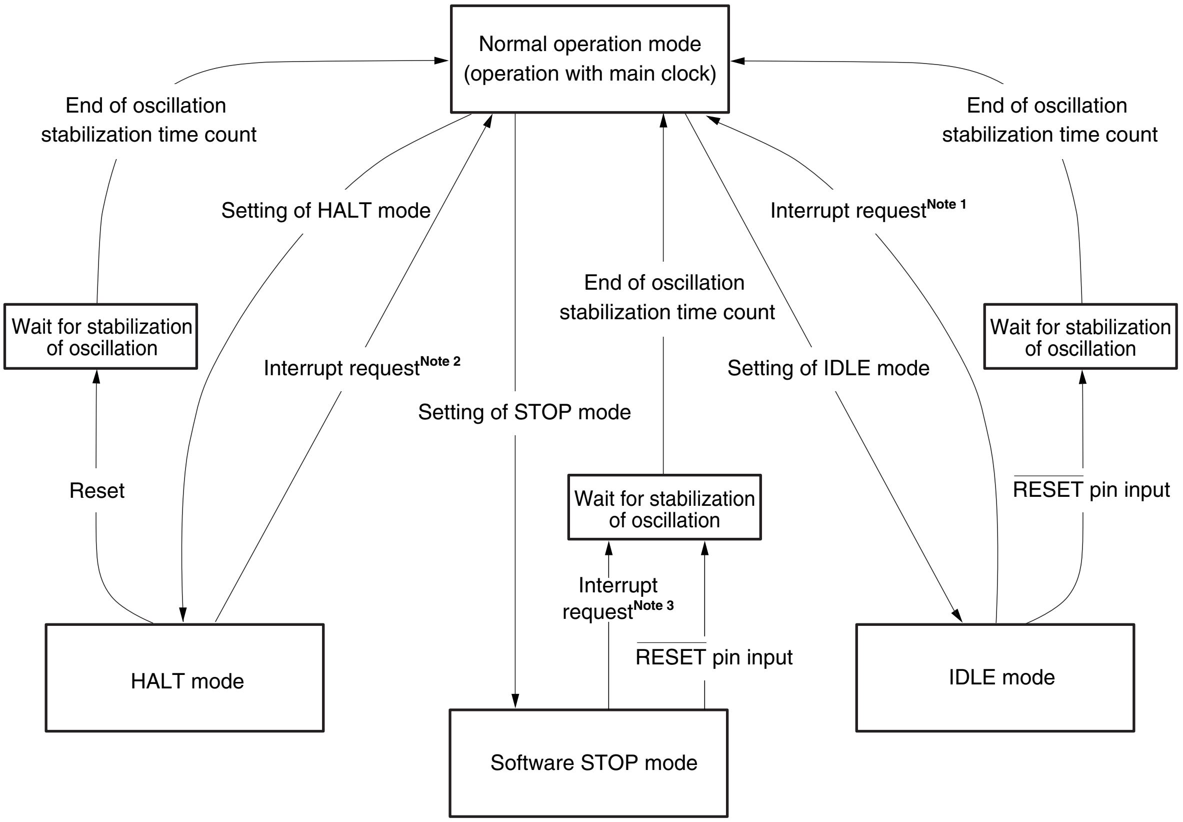
18.1 Overview

The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 18-1.

Table 18-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU
IDLE mode	Mode to stop all the internal operations of the chip except the oscillator
Software STOP mode	Mode to stop all the internal operations of the chip except the subclock oscillator
Subclock operation mode	Mode to use the subclock as the internal system clock
Sub-IDLE mode	Mode to stop all the internal operations of the chip, except the oscillator, in the subclock operation mode

★ **Figure 18-1. Status Transition**


Normal operation mode
(operation with main clock)
End of oscillation stabilization time count
End of oscillation stabilization time count
Setting of HALT mode
Interrupt request[Note 1]
End of oscillation stabilization time count
Wait for stabilization of oscillation
Interrupt request[Note 2]
Setting of STOP mode
Setting of IDLE mode
Wait for stabilization of oscillation
Reset
Wait for stabilization of oscillation
RESET pin input
Interrupt request[Note 3]
RESET pin input
HALT mode
IDLE mode
Software STOP mode

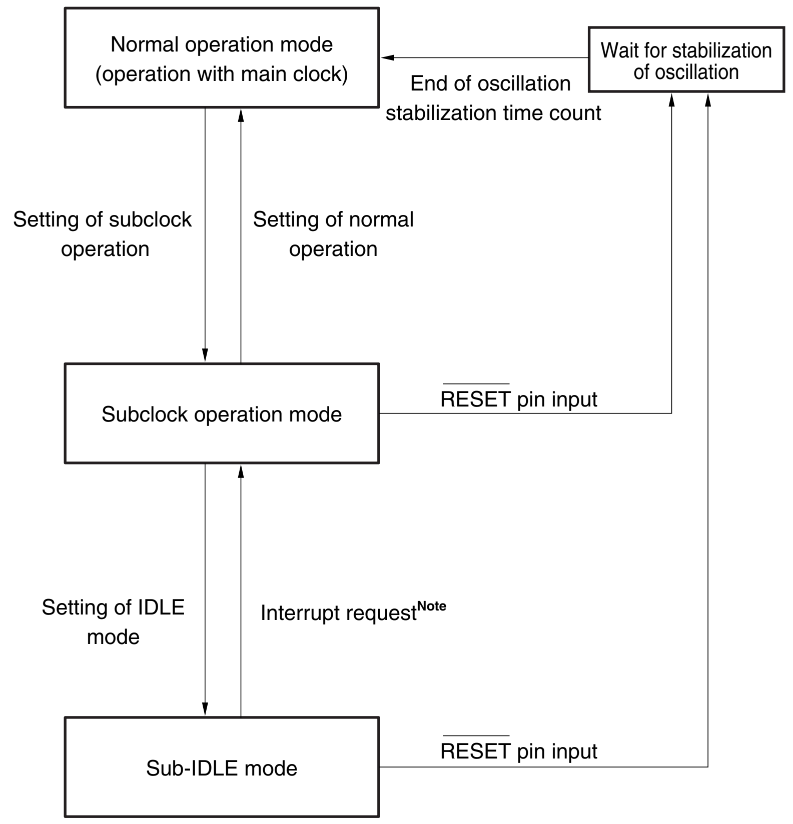
Notes 1. Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP6 pin input), or unmasked internal interrupt request from peripheral functions operable in IDLE mode.

2. Non-maskable interrupt request, unmasked maskable interrupt request, or reset input by WDT overflow.

3. Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP6 pin input), or unmasked internal interrupt request from peripheral functions operable in software STOP mode.

★

Figure 18-2. Status Transition (During Subclock Operation)


Normal operation mode
(operation with main clock)
Wait for stabilization of oscillation
End of oscillation stabilization time count
Setting of subclock operation
Setting of normal operation
Subclock operation mode
$\overline{\text{RESET}}$ pin input
Setting of IDLE mode
Interrupt request[Note]
Sub-IDLE mode
$\overline{\text{RESET}}$ pin input

Note Non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP6 pin input), or unmasked internal interrupt request from peripheral functions operable in sub-IDLE mode.

18.2 HALT Mode

18.2.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

In the HALT mode, the clock oscillator continues operating. Only clock supply to the CPU is stopped; clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 18-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Caution Insert five or more NOP instructions after the HALT instruction.

18.2.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request, $\overline{\text{RESET}}$ pin input, and reset by the watchdog timer.

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request or unmasked maskable interrupt request

The HALT mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the HALT mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the HALT mode is released and that interrupt request is acknowledged.

Table 18-2. Operation After Releasing HALT Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing HALT mode by $\overline{\text{RESET}}$ pin input and reset by watchdog timer

The same operation as the normal reset operation is performed.

Table 18-3. Operation Status in HALT Mode

Setting of HALT Mode / Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
CPU		Stops operation	
DMA		Operable	
Interrupt controller		Operable	
ROM correction		Stops operation	
16-bit timer/event counters (TM0, TM1)		Operable	
8-bit timer/event counters (TM2 to TM5)		Operable	
Real-timer counter		Operable when divided f_X/BRG output is selected as count clock	Operable
Watchdog timer		Operable	
Serial interface	CSI0 to CSI4	Operable	
	I^2C Note 1	Operable	
	UART0, UART1	Operable	
A/D converter		Operable	
D/A converter		Normal mode: Stops operation (output is retained) Note 2 Real-time output mode: Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before HALT mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.	

Notes 1. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

2. If the HALT mode is set immediately after the D/A conversion has started (during conversion), the operation continues until the D/A conversion is completed, when the D/A conversion is completed, the output is retained.

18.3 IDLE Mode

18.3.1 Setting and operation status

The IDLE mode is set by clearing the PSM bit of the power save mode register (PSMR) to 0 and setting the STP bit of the power save control register (PSC) to 1 in the normal operation mode.

In the IDLE mode, the clock oscillator continues operation but clock supply to the CPU and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Table 18-5 shows the operation status in the IDLE mode.

The IDLE mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The main clock oscillator does not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE mode has been released, in the same manner as when the HALT mode is released.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE mode.

18.3.2 Releasing IDLE mode

The IDLE mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP6 pin input), unmasked internal interrupt request from the peripheral functions operable in the IDLE mode, or $\overline{\text{RESET}}$ input.

After the IDLE mode has been released, the normal operation mode is restored.

(1) Releasing IDLE mode by non-maskable interrupt request or unmasked maskable interrupt request

The IDLE mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the IDLE mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the IDLE mode is released and that interrupt request is acknowledged.

Table 18-4. Operation After Releasing IDLE Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 18-5. Operation Status in IDLE Mode

Setting of IDLE Mode Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation	
ROM correction		Stops operation	
16-bit timer/event counters (TM0, TM1)		Stops operation	
8-bit timer/event counters (TM2 to TM5)		Stops operation	
Real-time counter		Operable when divided f_X/BRG output is selected as count clock	Operable
Watchdog timer		Stops operation	
Serial interface	CSI0 to CSI4	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0 to 4)	
	I^{2}C[Note 1]	Stops operation	
	UART0, UART1	Stops operation	
A/D converter		Operable when f_{BRG} is selected as operation clock	
D/A converter		Stops operation (output is retained)[Note 2]	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before IDLE mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE mode was set.	

Notes 1. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

2. If the IDLE mode is set immediately after the D/A conversion has started (during conversion), the operation continues until the D/A conversion is completed, when the D/A conversion is completed, the output is retained.

18.4 Software STOP Mode

18.4.1 Setting and operation status

The software STOP mode is set when the PSM bit of the PSMR register is set to 1 and the STP bit of the PSC register is set to 1 in the normal operation mode.

In the software STOP mode, the subclock oscillator continues operating but the main clock oscillator stops. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the software STOP mode was set are retained. The on-chip peripheral functions that operate with the clock oscillated by the subclock oscillator or an external clock continue operating.

Table 18-7 shows the operation status in the software STOP mode.

Because the software STOP mode stops operation of the main clock oscillator, it reduces the power consumption to a level lower than the IDLE mode. If the subclock oscillator and external clock are not used, the power consumption can be minimized with only leakage current flowing.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the software STOP mode.

18.4.2 Releasing software STOP mode

The software STOP mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP6 pin input), unmasked internal interrupt request from the peripheral functions operable in the software STOP mode, or $\overline{\text{RESET}}$ pin input.

After the software STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

(1) Releasing software STOP mode by non-maskable interrupt request or unmasked maskable interrupt request

The software STOP mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the software STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the software STOP mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the software STOP mode is released and that interrupt request is acknowledged.

Table 18-6. Operation After Releasing Software STOP Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing software STOP mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 18-7. Operation Status in Software STOP Mode

Setting of Software STOP Mode Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Stops operation	
Subclock oscillator		–	Oscillation enabled
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation	
ROM correction		Stops operation	
16-bit timer/event counters (TM0, TM1)		Stops operation	
8-bit timer/event counters (TM2 to TM5)		Stops operation	
Real-time counter		Stops operation	Operable when f_{XT} is selected as count clock
Watchdog timer		Stops operation	
Serial interface	CSI0 to CSI4	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0 to 4)	
	I^{2}C[Note]	Stops operation	
	UART0, UART1	Stops operation	
A/D converter		Stops operation	
D/A converter		Stops operation	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before software STOP mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the software STOP mode was set.	

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

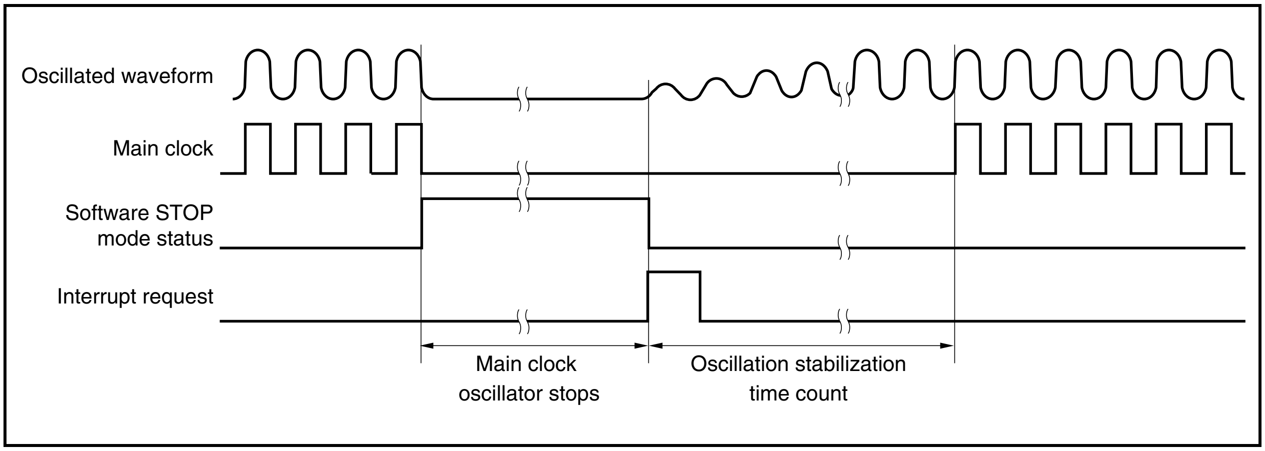
18.5 Securing Oscillation Stabilization Time

When the software STOP mode is released, only the oscillation stabilization time set by the OSTS register elapses. If the software STOP mode has been released by $\overline{\text{RESET}}$ pin input, however, the reset value of the OSTS register, $2^{19}/f_X$ elapses.

The timer for counting the oscillation stabilization time is shared with watchdog timer, so the oscillation stabilization time equal to the overflow time of the watchdog timer elapses.

The following illustrates the operation performed when the software STOP mode is released by an interrupt request.

Figure 18-3. Oscillation Stabilization Time


Oscillated waveform
Main clock
Software STOP mode status
Interrupt request
Main clock oscillator stops
Oscillation stabilization time count

Caution For details of the OSTS register, refer to 10.3 (1) Oscillation stabilization time selection register (OSTS).

18.6 Subclock Operation Mode

18.6.1 Setting and operation status

The subclock operation mode is set when the CK3 bit of the processor clock control register (PCC) is set to 1 in the normal operation mode.

When the subclock operation mode is set, the internal system clock is changed from the main clock to the subclock.

When the MCK bit of the PCC register is set to 1, the operation of the main clock oscillator is stopped. As a result, the system operates only with the subclock. However, watchdog timer stops counting when subclock operation is started (CLS bit of PCC register = 1). (Watchdog timer retains the value before the subclock operation mode was set.)

In the subclock operation mode, the power consumption can be reduced to a level lower than in the normal operation mode because the subclock is used as the internal system clock. In addition, the power consumption can be further reduced to the level of the software STOP mode by stopping the operation of the main system clock oscillator.

Table 18-8 shows the operation status in subclock operation mode.

Caution When manipulating the CK3 bit, do not change the set values of the CK2 to CK0 bits of the PCC register (using a bit manipulation instruction to manipulate the bit is recommended). For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

18.6.2 Releasing subclock operation mode

The subclock operation mode is released by $\overline{\text{RESET}}$ pin input when the CK3 bit is cleared to 0. If the main clock is stopped (MCK bit = 1), set the MCK bit to 1, secure the oscillation stabilization time of the main clock by software, and clear the CK3 bit to 0.

The normal operation mode is restored when the subclock operation mode is released.

Caution When manipulating the CK3 bit, do not change the set values of the CK2 to CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended).
For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

Table 18-8. Operation Status in Subclock Operation Mode

Setting of Subclock Operation Mode		Operation Status	
Item		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillation enabled	
CPU		Operable	
DMA		Operable	
Interrupt controller		Operable	
ROM correction		Operable	
16-bit timer/event counters (TM0, TM1)		Operable	Operable
8-bit timer/event counters (TM2 to TM5)		Operable	Operable
Real-time counter		Operable	Operable when f_{XT} is selected as count clock
Watchdog timer		Stops operation	
Serial interface	CSI0 to CSI4	Operable	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0 to 4)
	I^{2}C[Note]	Operable	Stops operation
	UART0, UART1	Operable	Stops operation
A/D converter		Operable	Stops operation
D/A converter		Operable	Operable (only when normal mode is selected)
External bus interface		Operable	
Port function		Settable	
Internal data		Settable	

Note μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

18.7 Sub-IDLE Mode

18.7.1 Setting and operation status

The sub-IDLE mode is set when the PSM bit of the power save mode register (PSMR) is cleared to 0 and the STP bit of the power save control register (PSC) is set to 1 in the subclock operation mode.

In this mode, the clock oscillator continues operation but clock supply to the CPU and the other on-chip peripheral functions is stopped.

As a result, program execution is stopped and the contents of the internal RAM before the sub-IDLE mode was set are retained. The CPU and the other on-chip peripheral functions are stopped. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Because the sub-IDLE mode stops operation of the CPU and other on-chip peripheral functions, it can reduce the power consumption more than the subclock operation mode. If the sub-IDLE mode is set after the main clock has been stopped, the power consumption can be reduced to a level as low as that in the software STOP mode.

Table 18-10 shows the operation status in the sub-IDLE mode.

18.7.2 Releasing sub-IDLE mode

The sub-IDLE mode is released by a non-maskable interrupt request (NMI pin input), unmasked external interrupt request (INTP0 to INTP6 pin input), unmasked internal interrupt request from the peripheral functions operable in the sub-IDLE mode, or $\overline{\text{RESET}}$ pin input.

When the sub-IDLE mode is released by an interrupt request, the subclock operation mode is set. If it is released by $\overline{\text{RESET}}$ pin input, the normal operation mode is restored.

(1) Releasing sub-IDLE mode by non-maskable interrupt request or unmasked maskable interrupt request

The sub-IDLE mode is released by a non-maskable interrupt request or an unmasked maskable interrupt request, regardless of the priority of the interrupt request. If the sub-IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request with a priority lower than that of the interrupt request currently being serviced is issued, only the sub-IDLE mode is released, and that interrupt request is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request), the sub-IDLE mode is released and that interrupt request is acknowledged.

Table 18-9. Operation After Releasing Sub-IDLE Mode by Interrupt Request

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request	Execution branches to the handler address	
Maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing sub-IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 18-10. Operation Status in Sub-IDLE Mode

Setting of Sub-IDLE Mode Item		Operation Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillation enabled	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation	
ROM correction		Stops operation	
16-bit timer/event counters (TM0, TM1)		Stops operation	
8-bit timer/event counters (TM2 to TM5)		Stops operation	
Real-timer counter		Operable	Operable when f_{XT} is selected as count clock
Watchdog timer		Stops operation	
Serial interface	CSI0 to CSI4	Operable when $\overline{\text{SCKn}}$ input clock is selected as operation clock (n = 0 to 4)	
	I^{2}C^Note 1^	Stops operation	
	UART0, UART1	Stops operation	
A/D converter		Stops operation	
D/A converter		Stops operation (output is retained)Note 2	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before sub-IDLE mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the sub-IDLE mode was set.	

Notes 1. μPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y only

2. If the sub-IDLE mode is set immediately after the D/A conversion has started (during conversion), the operation continues until the D/A conversion is completed, when the D/A conversion is completed, the output is retained.

18.8 Control Registers

(1) Power save control register (PSC)

This is an 8-bit register that controls the standby function. The STP bit of this register is used to specify the normal mode or IDLE/software STOP mode. The PSC register is a special register (refer to **3.4.8 Special registers**). Data can be written to this register only in a specific sequence so that its contents are not rewritten by mistake due to a program hang-up.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	4	3	2	<1>	0
PSC	0	0	0	0	0	0	STP	0

STP	Sets IDLE/software STOP mode
0	Normal mode
1	IDLE/software STOP mode

(2) Power save mode register (PSMR)

This is an 8-bit register that controls the operation status and clock operation in the power save mode.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM

PSM	Specifies operation in software standby mode (valid when bit 1 (STP) of the PSC register is set to 1)
0	IDLE mode
1	Software STOP mode

Cautions 1. Be sure to clear bits 1 to 7 of the PSMR register to 0.

2. The PSM bit is valid only when the STP bit of the PSC register is set to 1.

(3) Oscillation stabilization time selection register (OSTS)

This is an 8-bit register that controls the operation status and clock operation in the power save mode.

Refer to **10.3 (1) Oscillation stabilization time selection register (OSTS)**.

CHAPTER 19 RESET FUNCTION

19.1 Overview

The following reset functions are available.

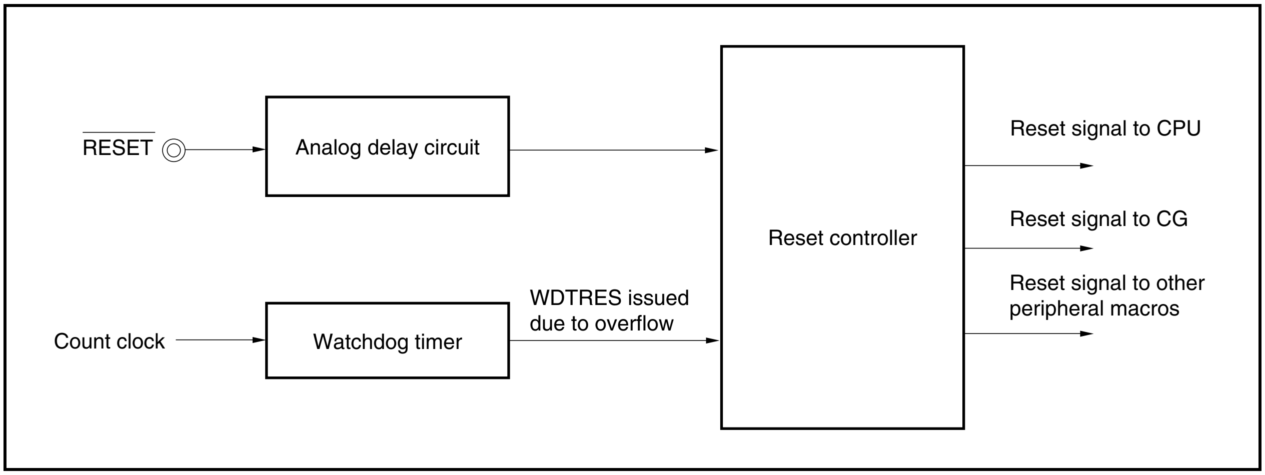
- Reset function by $\overline{\text{RESET}}$ pin input
- Reset function by WDT overflow (WDTRES)

If the $\overline{\text{RESET}}$ pin goes high, the reset status is released, and the CPU starts executing the program. Initialize the contents of each register in the program as necessary.

The $\overline{\text{RESET}}$ pin has a noise eliminator that operates by analog delay to prevent malfunction caused by noise.

19.2 Configuration

Figure 19-1. Reset Block Diagram



19.3 Operation

The system is reset, initializing each hardware unit, when a low level is input to the $\overline{\text{RESET}}$ pin by WDT overflow (WDTRES).

While a low level is being input to the $\overline{\text{RESET}}$ pin, the main clock oscillator stops. Therefore, the overall power consumption of the system can be reduced.

If the $\overline{\text{RESET}}$ pin goes high or if WDTRES is received, the reset status is released.

If the reset status is released by $\overline{\text{RESET}}$ pin input, the oscillation stabilization time elapses (reset value of OSTS register: 2^{19}/fx) and then the CPU starts program execution.

If the reset status is released by WDTRES, the oscillation stabilization time is not inserted because the main system clock oscillator does not stop.

Note Reset by WDT overflow (WDTRES) is valid only when the WDTM4 and WDTM3 bits of the watchdog timer mode register (WDTM) are set to “11” (refer to **10.3 (3)**).

Table 19-1. Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation stops (f_X = 0 level).	Oscillation starts
Subclock oscillator (f_{XT})	Oscillation can continue without effect from reset[Note 1].	
Peripheral clock (f_{XX} to f_{XX}/1024), internal system clock (f_{XX}), CPU clock (f_{CPU})	Operation stops	Operation starts. However, operation stops during oscillation stabilization time count.
WDT clock (f_{XW})	Operation stops	Operation starts
Internal RAM	Undefined if power-on reset occurs or writing data to RAM and reset conflict (data loss). Otherwise, retains values immediately before reset input.	
I/O lines (ports)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status	
Real-time counter	Operation can be started[Note 2]	
Other on-chip peripheral functions	Operation stops	Operation can be started

Notes 1. The on-chip feedback resistor is “connected” by default (refer to **6.3 (1) Processor clock control register (PCC)**).

2. The real-time counter performs a count operation on the subclock when the $\overline{\text{RESET}}$ signal is input.
If a clock resulting from dividing the main clock (f_X) by the baud rate generator (f_{BRG}) is used as the count clock, the count clock is changed to the subclock (f_{XT}).

Table 19-2. Hardware Status on Occurrence of WDTRES

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation continues[Note 1]	
Subclock oscillator (f_{XT})	Oscillation can continue without effect from reset[Note 1].	
Peripheral clock (f_{XX} to f_{XX}/1024), internal system clock (f_{XX}), CPU clock (f_{CPU})	Operation stops	Operation starts
WDT clock (f_{XW})	Operation continues	
Internal RAM	Undefined if writing data to RAM and reset conflict (data loss). Otherwise, retains values immediately before reset input.	
I/O lines (ports)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status	
Real-time counter	Operation continues[Note 2]	
Other on-chip peripheral functions	Operation stops	Operation can be started

Notes 1. The on-chip feedback resistor is “connected” by default (refer to **6.3 (1) Processor clock control register (PCC)**).

2. The real-time counter performs a count operation on the subclock when the $\overline{\text{RESET}}$ signal is input.
If a clock resulting from dividing the main clock (f_X) by the baud rate generator (f_{BRG}) is used as the count clock, the count clock is changed to the subclock (f_{XT}).

Figure 19-2. Hardware Status on $\overline{\text{RESET}}$ Input

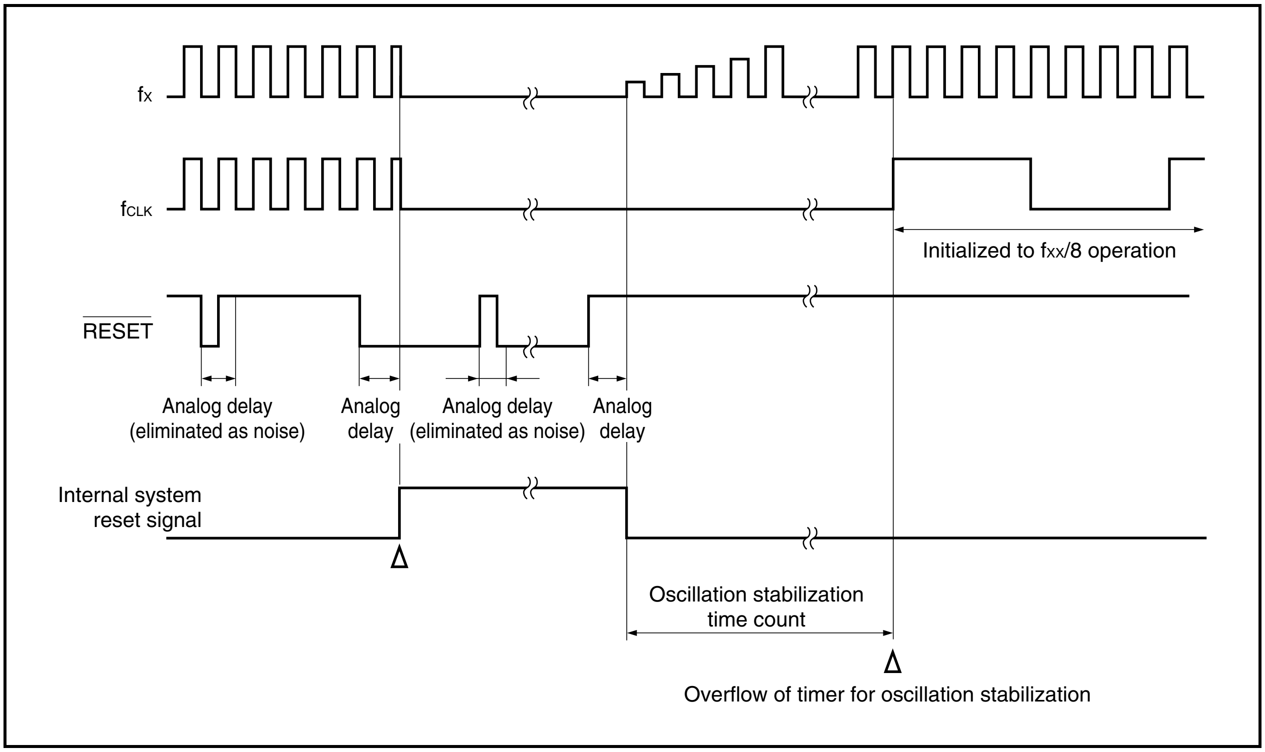
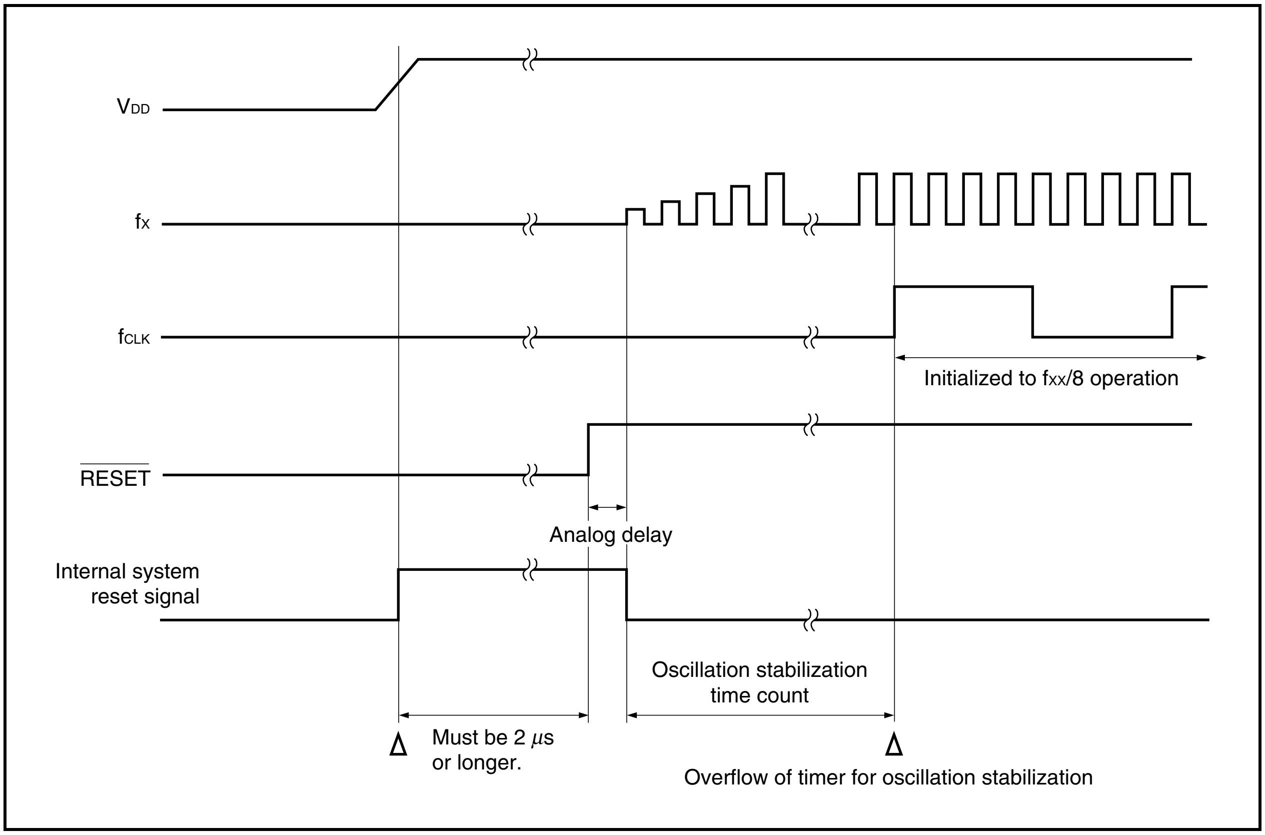


Figure 19-3. Operation on Power Application


VDD
fX
fCLK
Initialized to fXX/8 operation
RESET
Analog delay
Internal system
reset signal
Oscillation stabilization
time count
Must be 2 μs
or longer.
Overflow of timer for oscillation stabilization

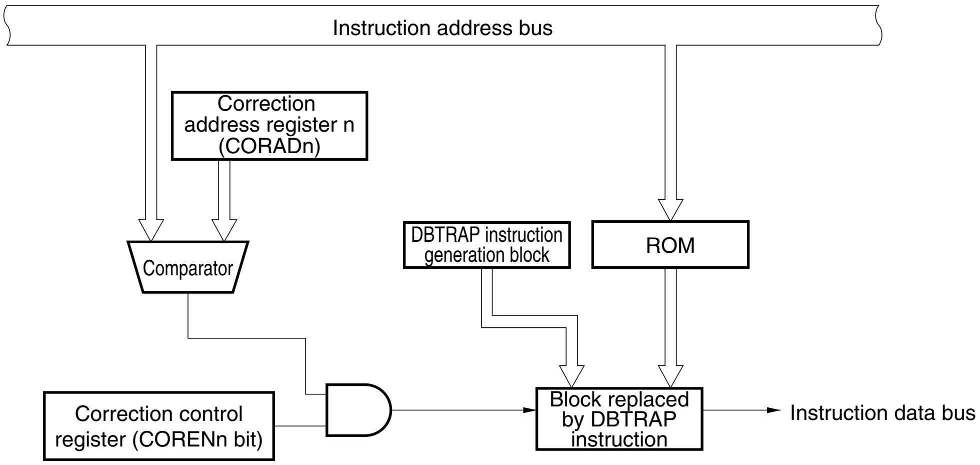
CHAPTER 20 ROM CORRECTION FUNCTION

20.1 Overview

The ROM correction function is used to replace part of the program in the internal ROM with the program of an external RAM or the internal RAM.

By using this function, instruction bugs found in the internal ROM can be corrected at up to four places.

Figure 20-1. Block Diagram of ROM Correction


Instruction address bus
Correction address register n (CORADn)
Comparator
DBTRAP instruction generation block
ROM
Correction control register (CORENn bit)
Block replaced by DBTRAP instruction
Instruction data bus

Remark n = 0 to 3

20.2 Control Registers

20.2.1 Correction address registers 0 to 3 (CORAD0 to CORAD3)

These registers are used to set the first address (correction address) of the instruction to be corrected in the ROM.

The program can be corrected at up to four places because four correction address registers (CORADn) are provided (n = 0 to 3).

The CORADn register can be read or written in 32-bit units.

If the higher 16 bits of the CORADn register are used as the CORADnH register, and the lower 16 bits as the CORADnL register, these registers can be read or written in 16-bit units.

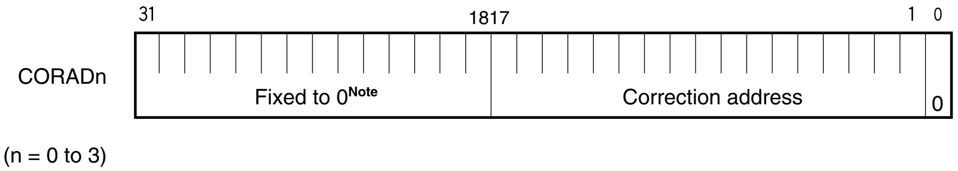
Set correction addresses within the range of 0000000H to 003FFFEH.

Fix bits 0 and 18 to 31[Note] to 0.

These registers are cleared to 00000000H after reset.

Note Fix bits 0 and 17 to 31 to 0 in the μPD703200 and 703200Y.

After reset: 00000000H R/W Address: CORAD0: FFFFF840H CORAD2: FFFFF848H
CORAD1: FFFFF844H CORAD3: FFFFF84CH



Note Bits 17 to 31 in the μPD703200 and 703200Y.

20.2.2 Correction control register (CORCN)

This register disables or enables the correction operation of correction address register n (CORADn) (n = 0 to 3). Each channel can be enabled or disabled by this register.

This register can be read or written in 8-bit or 1-bit units.

This register is cleared to 00H after reset.

After reset: 00H R/W Address: FFFFF880H

	7	6	5	4	3	2	1	0
CORCN	0	0	0	0	COREN3	COREN2	COREN1	COREN0

CORENn	Enables/disables correction operation
0	Disabled
1	Enabled

Remark n = 0 to 3

Table 20-1. Correspondence Between CORCN Register Bits and CORADn Registers

CORCN Register Bit	Corresponding CORADn Register
COREN3	CORAD3
COREN2	CORAD2
COREN1	CORAD1
COREN0	CORAD0

20.3 ROM Correction Operation and Program Flow

<1> If the address to be corrected and the fetch address of the internal ROM match, the fetch code is replaced by the DBTRAP instruction.

<2> When the DBTRAP instruction is executed, execution branches to address 00000060H.

<3> Software processing after branching causes the result of ROM correction to be judged (the fetch address and ROM correction operation are confirmed) and execution to branch to the correction software.

<4> After the correction software has been executed, the return address is set, and return processing is started by the DBRET instruction.

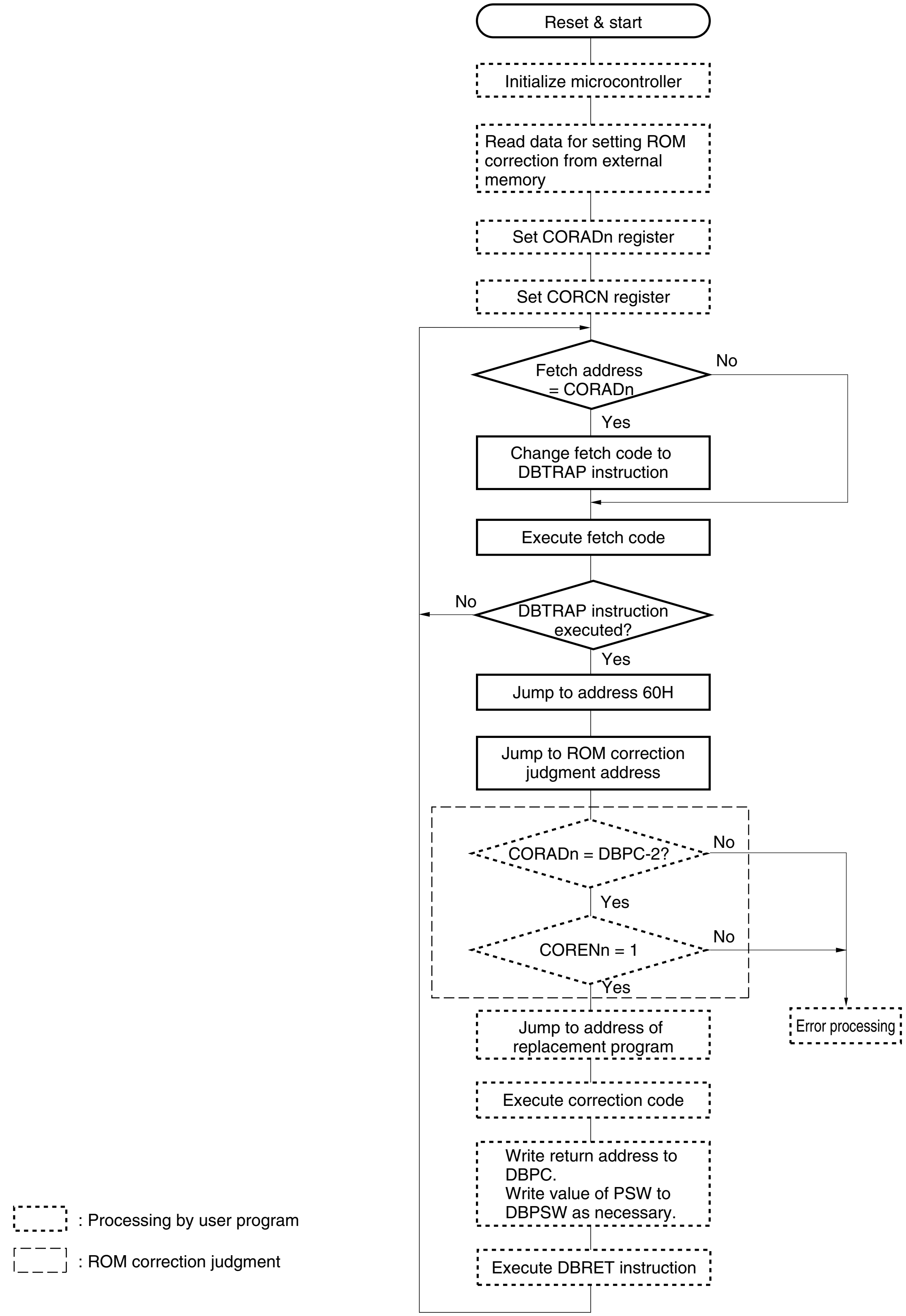
Cautions 1. The software that performs <3> and <4> must be executed in the internal ROM/RAM.

2. Develop the program so that the ROM correction function is not used until data has been completely written to the CORCN register that controls ROM correction.

3. When setting an address to be corrected to the CORADn register, clear the higher bits to 0 in accordance with the capacity of the internal ROM. Before setting, make sure that the CORCN register is 00H.

4. The ROM correction function cannot be used to correct the data of the internal ROM. It can only be used to correct instruction codes. If ROM correction is used to correct data, that data is replaced with the DBTRAP instruction code.

Figure 20-2. ROM Correction Operation and Program Flow


Reset & start
Initialize microcontroller
Read data for setting ROM correction from external memory
Set CORADn register
Set CORCN register
Fetch address = CORADn
No
Yes
Change fetch code to DBTRAP instruction
Execute fetch code
No
DBTRAP instruction executed?
Yes
Jump to address 60H
Jump to ROM correction judgment address
CORADn = DBPC-2?
No
Yes
CORENn = 1
No
Yes
Error processing
Jump to address of replacement program
Execute correction code
Write return address to DBPC.
Write value of PSW to DBPSW as necessary.
Execute DBRET instruction
: Processing by user program
: ROM correction judgment

Remark n = 0 to 3

CHAPTER 21 FLASH MEMORY

The following products are the flash memory versions of the V850ES/SA2 and V850ES/SA3.

Caution There are differences in the amount of noise tolerance and noise radiation between flash memory versions and mask ROM versions. When considering changing from a flash memory version to a mask ROM version during the process from experimental manufacturing to mass production, make sure to sufficiently evaluate commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.

(1) V850ES/SA2

μPD70F3201, 70F3201Y:	256 KB flash memory versions

(2) V850ES/SA3

μPD70F3204, 70F3204Y:	256 KB flash memory versions

In the instruction fetch to this flash memory, 4 bytes can be accessed by a single clock, the same as in the mask ROM version.

Writing to flash memory can be performed with the memory mounted on the target system (on board). A dedicated flash programmer (such as PG-FP4) is connected to the target system to perform writing.

The following can be considered as the development environment and the applications using a flash memory.

- Software can be altered after the V850ES/SA2 or V850ES/SA3 is solder-mounted on the target system.
- Small scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.

21.1 Features

○ Erasure unit
- Chip batch erase
- Block erase

  Blocks 0 to 3: 4 KB
  Blocks 4 to 7: 60 KB

○ Erase/write method
- Dedicated flash programmer mode (CSI0, UART0, CSI0 + HS)
- Flash self programming mode

○ Support functions
- Erase/write prohibit function (security function)

○ Other
- On-board rewrite
- Erase/write with a single power supply
- Interrupt acknowledgment during self programming

21.1.1 Erasure unit

The erasure units for 256 KB flash memory versions are shown below.

(1) Chip erase

The area of xx000000H to xx03FFFFH can be erased at the same time.

(2) Block erase

Erasure can be performed in block units (60 KB × 4, 4 KB × 4).

Block 0:	4 KB
Block 1:	4 KB
Block 2:	4 KB
Block 3:	4 KB
Block 4:	60 KB
Block 5:	60 KB
Block 6:	60 KB
Block 7:	60 KB

★ 21.2 Functional Overview

The on-chip flash memory of the V850ES/SA2 and V850ES/SA3 can be rewritten using the rewriting functions of the dedicated flash programmer (on-board/off-board programming), regardless of whether or not the V850ES/SA2 or V850ES/SA3 has been mounted in the target system. The V850ES/SA2 and V850ES/SA3 support a security function that prohibits rewriting user programs written to the on-chip flash memory, thus protecting the program from being modified by unauthorized parties.

The user-program based rewrite function (self programming) is a rewrite method suited to applications that assume program modifications after manufacturing and shipment of the target system, enabling rewriting under any conditions because interrupt servicing during self programming is supported.

Table 21-1. Rewrite Methods

Rewrite Method	Function	Operating Mode
On-board programming	Rewriting flash memory is possible by using the dedicated flash programmer after the V850ES/SA2 or V850ES/SA3 has been mounted on the target system	Flash memory programming mode
Off-board programming	Rewriting flash memory is possible by using the dedicated flash programmer and dedicated program adapter board (such as the FA Series) before the V850ES/SA2 or V850ES/SA3 is mounted on the target system.	
Self programming mode	Rewriting flash memory is possible[Note] by executing a user program already written in the flash memory via on-board/off-board programming.	Normal operation mode

Note Instruction fetch and data access is not possible from the on-board flash memory area during self programming, so the program for rewriting the internal RAM or external memory must have been transferred and executed beforehand.

Remark The FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 21-2. Basic Functions

Function	Description	Support (○: Supported/×: Not Supported)	
		On-Board/Off-Board Programming	Self Programming
Block erase	Contents of specified block in memory are erased.	○	○
Chip erase	All contents in memory are erased at once.	○	×
Writing	Writing to specified addresses and verify check are executed.	○	○
Verify/checksum	Data read from flash memory and data transferred from flash programmer are compared.	○	×[Note 1]
Blank check	Erasure status of entire memory is checked.	○	○
Security setting	Use of block erase command/chip erase command/program command is disabled.	○	×[Note 2]

Notes 1. Reading is possible by user program.

2. The values set by on-board/off-board programming can only be retained.

A list of security functions is shown below. The block erase command prohibition/chip erase command prohibition/program command prohibition functions are enabled as the initial status after shipment; therefore, making security settings is possible only by setting while rewriting via on-board/off-board programming using the dedicated flash programmer. Security settings can be used in combination.

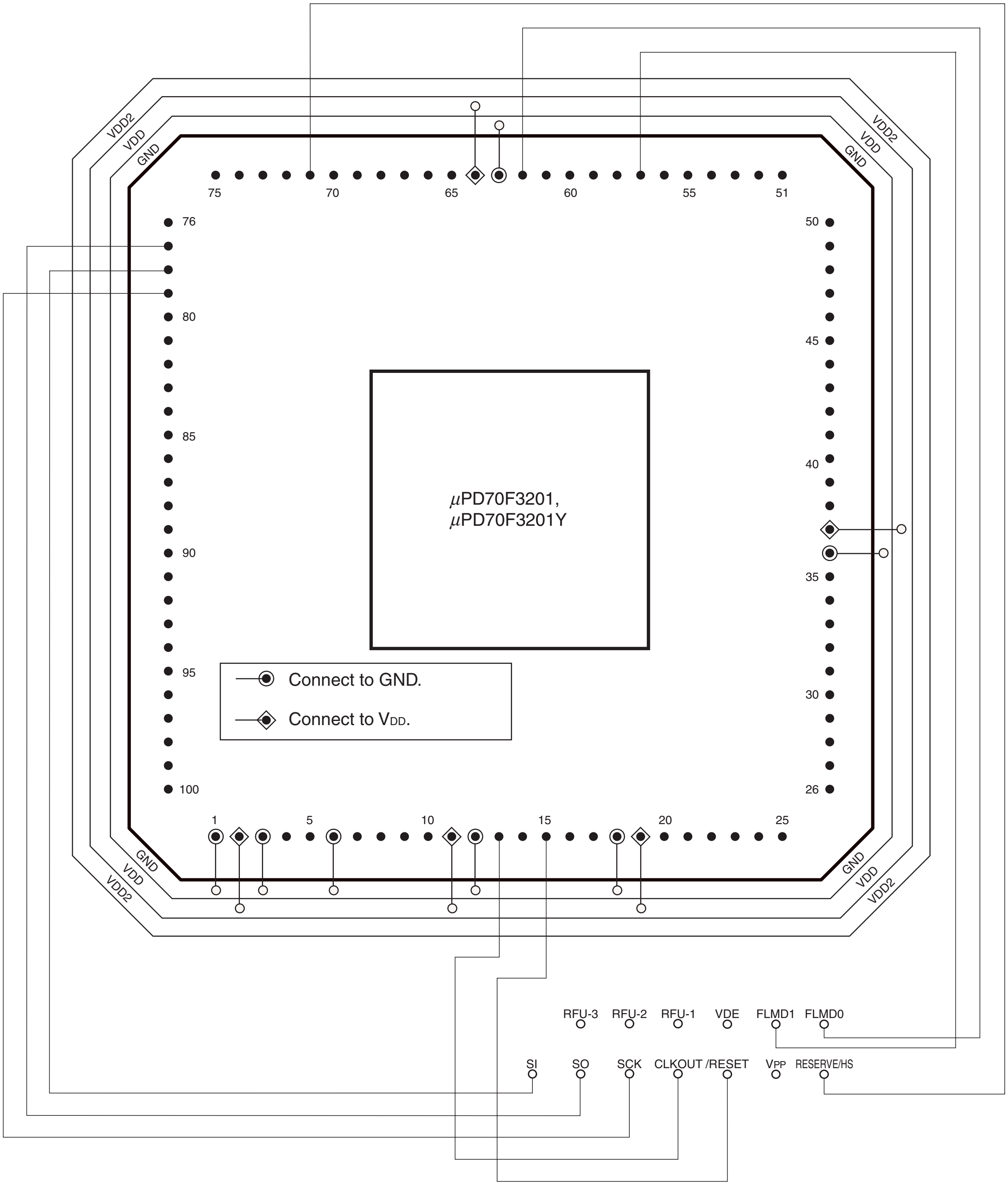
Table 21-3. List of Security Functions

Function	Description	Rewrite Operation When Prohibited	
		On-Board/Off-Board Programming	Self Programming
Block erase command prohibition	Executing the block erase command to all blocks is prohibited. The prohibition setting can be initialized by executing the chip erase command.	Block erase command: ×	Rewriting is possible regardless of prohibition settings.
		Chip erase command: ○	
		Program command: ○	
Chip erase command prohibition	Executing block erasure on all blocks and the chip erase command is prohibited. Once this prohibition function is set, the chip erase command cannot be executed, so no prohibition function can be initialized.	Block erase command: ×	
		Chip erase command: ×	
		Program command: ○	
Program command prohibition	Executing a write command to all blocks and the block erase command is prohibited. The prohibition setting can be initialized by executing the chip erase command.	Block erase command: ×	
		Chip erase command: ○	
		Program command: ×	

Remark ○: Can be executed

×: Cannot be executed

★ **Figure 21-1. Wiring Example of V850ES/SA2 Flash Write Adapter (FA-100GC-8EU-A)**



Remarks 1. Handle the pins not described above in accordance with the specified handling of unused pins (refer to **2.4 Types of Pin I/O Circuits, I/O Buffer Power Supplies, and Connection of Unused Pins**).

When connecting to V_{DD} via a resistor, use of a resistor of 1 kΩ to 10 kΩ is recommended.

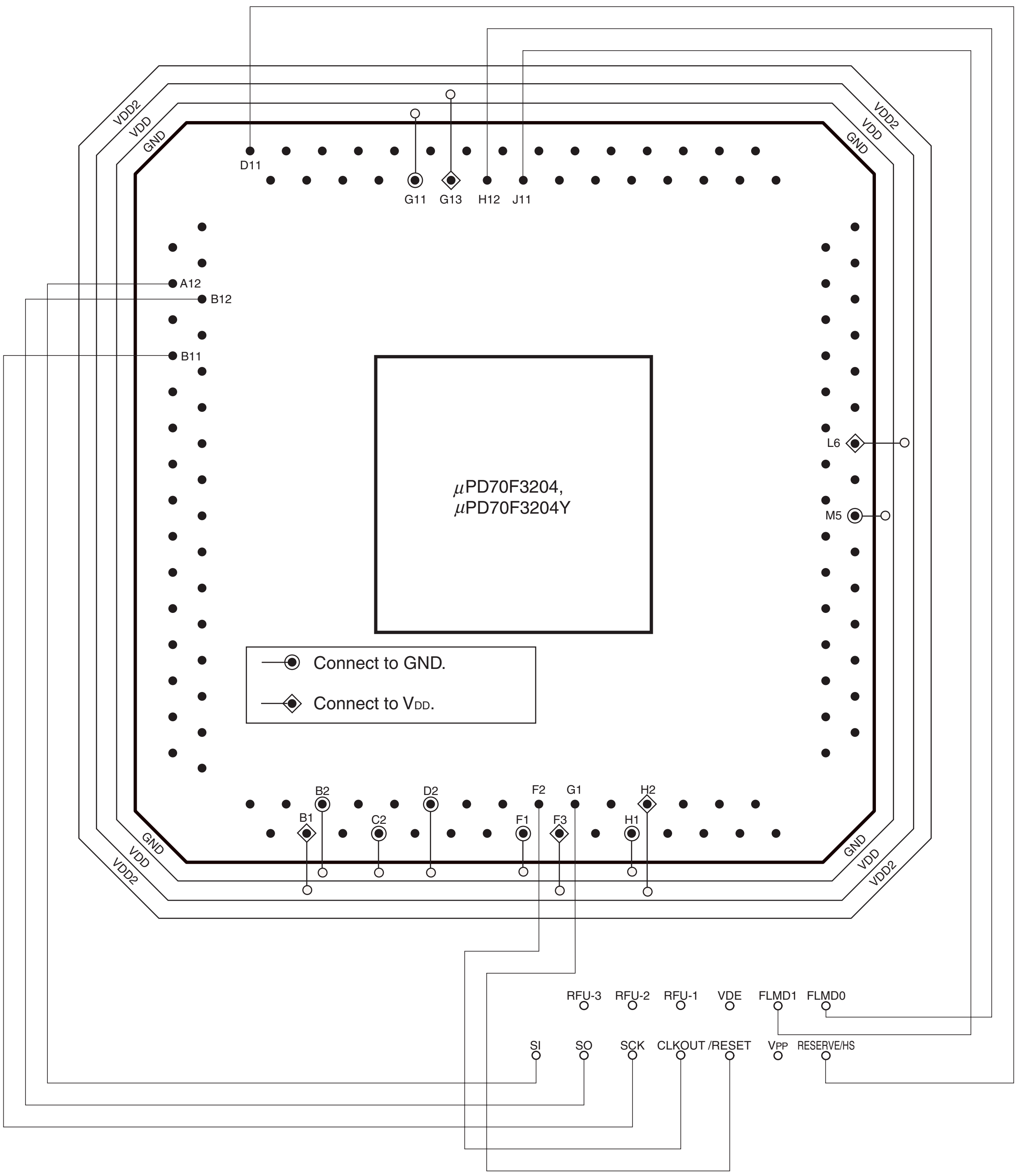
2. This adapter is for a 100-pin plastic LQFP package.

3. This diagram shows the wiring when using a handshake-supporting CSI.

★ **Table 21-4. Wiring of V850ES/SA2 Flash Write Adapter (FA-100GC-8EU-A)**

Pin Configuration of Flash Programmer (PG-FP4)			With CSI0 + HS		With CSI0		With UART0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	P41/SO0/SDA	78	P41/SO0/SDA	78	P31/SO1/TXD0	9
SO/TxD	Output	Transmit signal	P40/SI0	77	P40/SI0	77	P30/SI1/RXD0	8
SCK	Output	Transfer clock	P42/$\overline{\text{SCK0}}$/SCL	79	P42/$\overline{\text{SCK0}}$/SCL	79	Not needed	Not needed
CLKOUT	Output	Clock to V850ES/SA2	X1	13	X1	13	X1	13
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	15	$\overline{\text{RESET}}$	15	$\overline{\text{RESET}}$	15
FLMD0	Input	Write voltage	FLMD0	62	FLMD0	62	FLMD0	62
FLMD1	Input	Write voltage	PLD5/AD5/FLMD1	57	PLD5/AD5/FLMD1	57	PLD5/AD5/FLMD1	57
HS	Input	Handshake signal for CSI0 + HS	PDH0/A16	71	Not needed	Not needed	Not needed	Not needed
VDD	–	VDD voltage generation/ voltage monitor	V_{DD}	11, 19	V_{DD}	11, 19	V_{DD}	11, 19
			AV_{DD}	2	AV_{DD}	2	AV_{DD}	2
			EV_{DD}	37, 64	EV_{DD}	37, 64	EV_{DD}	37, 64
GND	–	Ground	V_{SS}	12, 18	V_{SS}	12, 18	V_{SS}	12, 18
			AV_{SS}	3	AV_{SS}	3	AV_{SS}	3
			AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
			AV_{REF1}	6	AV_{REF1}	6	AV_{REF1}	6
			EV_{SS}	36, 63	EV_{SS}	36, 63	EV_{SS}	36, 63

★ **Figure 21-2. Wiring Example of V850ES/SA3 Flash Write Adapter (FA-121F1-EA6-A)**



Remarks 1. Handle the pins not described above in accordance with the specified handling of unused pins (refer to **2.4 Types of Pin I/O Circuits, I/O Buffer Power Supplies, and Connection of Unused Pins**).

When connecting to V_{DD} via a resistor, use of a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is for a 121-pin plastic FBGA package.

3. This diagram shows the wiring when using a handshake-supporting CSI.

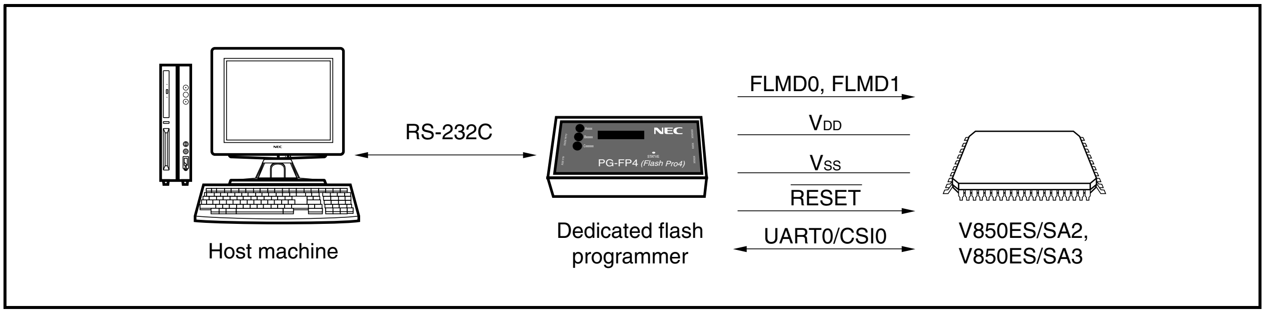
★ **Table 21-5. Wiring of V850ES/SA3 Flash Write Adapter (FA-121F1-EA6-A)**

Pin Configuration of Flash Programmer (PG-FP3/PG-FP4)			With CSI0 + HS		With CSI0		With UART0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	P41/SO0/SDA	A12	P41/SO0/SDA	A12	P31/SO1/TXD0	E2
SO/TxD	Output	Transmit signal	P40/SI0	B12	P40/SI0	B12	P30/SI1/RXD0	E1
SCK	Output	Transfer clock	P42/$\overline{\text{SCK0}}$/SCL	B11	P42/$\overline{\text{SCK0}}$/SCL	B11	Not needed	Not needed
CLK	Output	Clock to V850ES/SA3	X1	F2	X1	F2	X1	F2
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	G1	$\overline{\text{RESET}}$	G1	$\overline{\text{RESET}}$	G1
FLMD0	Output	Write voltage	FLMD0	H12	FLMD0	H12	FLMD0	H12
FLMD1	Output	Write voltage	PDL5/AD5/FLMD1	J11	PDL5/AD5/FLMD1	J11	PDL5/AD5/FLMD1	J11
HS	Input	Handshake signal for CSI0 + HS	PDH0/A16	D11	Not needed	Not needed	Not needed	Not needed
VDD	–	VDD voltage generation/ voltage monitor	V_{DD}	F3, H2	V_{DD}	F3, H2	V_{DD}	F3, H2
			AV_{DD}	B1	AV_{DD}	B1	AV_{DD}	B1
			EV_{DD}	G13, L6	EV_{DD}	G13, L6	EV_{DD}	G13, L6
GND	–	Ground	V_{SS}	F1, H1	V_{SS}	F1, H1	V_{SS}	F1, H1
			AV_{SS}	C2	AV_{SS}	C2	AV_{SS}	C2
			AV_{REF0}	B2	AV_{REF0}	B2	AV_{REF0}	B2
			AV_{REF1}	D2	AV_{REF1}	D2	AV_{REF1}	D2
			EV_{SS}	G11, M5	EV_{SS}	G11, M5	EV_{SS}	G11, M5

21.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of V850ES/SA2 and V850ES/SA3.

Figure 21-3. Environment Required for Writing Programs to Flash Memory



A host machine is required for controlling the dedicated flash programmer.

UART0 or CSI0 is used for the interface between the dedicated flash programmer and the V850ES/SA2 or V850ES/SA3 to perform writing, erasing, etc. A dedicated program adapter (FA Series) required for off-board writing.

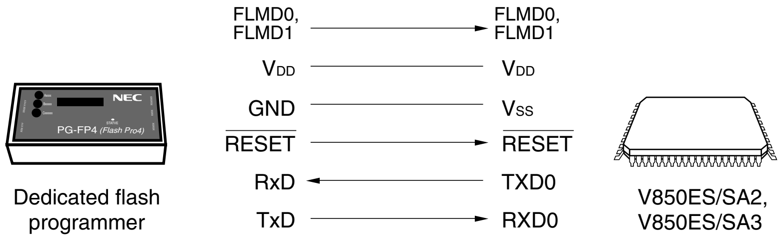
21.4 Communication Mode

The communication between the dedicated flash programmer and the V850ES/SA2 or V850ES/SA3 is performed by serial communication using UART0 or CSI0 of the V850ES/SA2, V850ES/SA3.

(1) UART0

Transfer rate: 9,600 to 153,600 bps

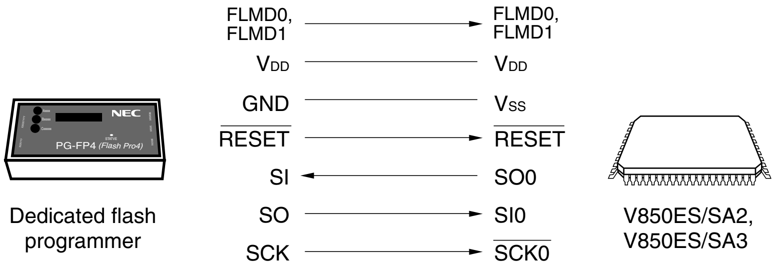
Figure 21-4. Communication with Dedicated Flash Programmer (UART0)



(2) CSI0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

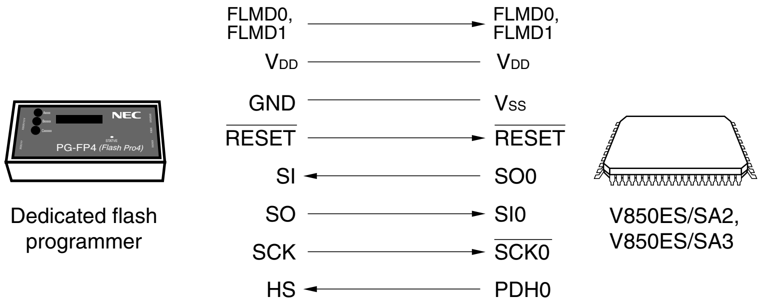
Figure 21-5. Communication with Dedicated Flash Programmer (CSI0)



(3) CSI0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 21-6. Communication with Dedicated Flash Programmer (CSI0 + HS)



The dedicated flash programmer outputs the transfer clock, and the V850ES/SA2 and V850ES/SA3 operate as slaves.

When the PG-FP4 is used as the dedicated flash programmer, it generates the following signals to the V850ES/SA2 or V850ES/SA3. For details, refer to the PG-FP4 manual.

Table 21-6. Signal Generation of Dedicated Flash Programmer (PG-FP4)

PG-FP4			V850ES/SA2, V850ES/SA3	Connection Handling		
Signal Name	I/O	Pin Function	Pin Name	CSI0	UART0	CSI0 + HS
FLMD0	Output	Mode setting, writing enable/ disable	FLMD0	◎	◎	◎
FLMD1	Output	Mode setting	FLMD1	◎ Note 1	◎ Note 1	◎ Note 1
V_{DD}	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD}	× Note 2	× Note 2	× Note 2
GND	–	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output to V850ES/SA2, V850ES/SA3	X1	× Note 3	× Note 3	× Note 3
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI/RxD	Input	Receive signal	SO0/TxD0	◎	◎	◎
SO/TxD	Output	Transmit signal	SI0/RxD0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK0}}$	◎	×	◎
HS	Input	Handshake signal of CSI0 + HS	PDH0	×	×	◎

Notes 1. Wire as shown in Figures 21-1 and 21-2, or connect to GND via a pull-down resistor on-board.

2. Connect when power is supplied from the PG-FP4.

3. Connect when the clock is supplied from the PG-FP4 (wire as shown in Figures 21-1 and 21-2, or generate an oscillator on-board to supply the clock).

Remark ◎: Always connected
×: Does not need to be connected

21.5 Pin Connection

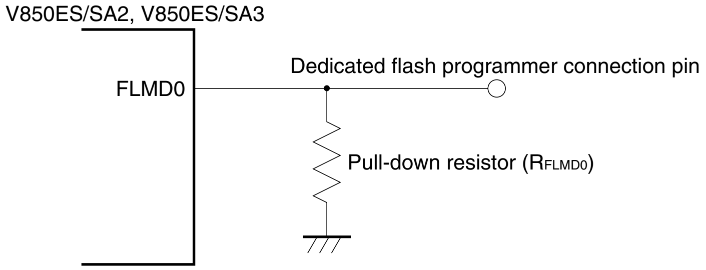
When performing on-board writing, mount a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

When switched to the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, all the ports enter the output high-impedance status, so that pin handling is required when the external device does not acknowledge the output high-impedance status.

21.5.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. The following shows an example of the connection of the FLMD0 pin.

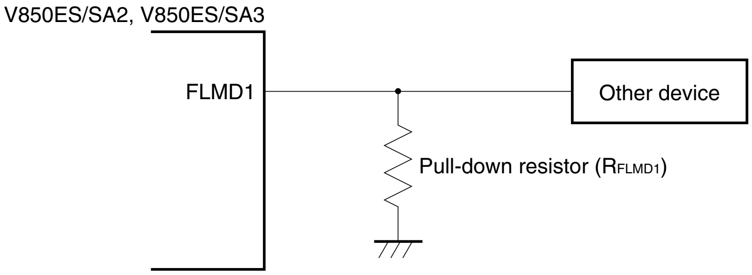
Figure 21-7. FLMD0 Pin Connection Example


V850ES/SA2, V850ES/SA3
FLMD0
Dedicated flash programmer connection pin
Pull-down resistor (R_{FLMD0})

★ 21.5.2 FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.

Figure 21-8. FLMD1 Pin Connection Example



Caution If the V_{DD} signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

Table 21-7. Operation Mode According to FLMD0 and FLMD1 Pin Settings

FLMD0	FLMD1	Operation Mode
0	Don’t care	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

21.5.3 Serial interface pin

The following shows the pins used by each serial interface.

Table 21-8. Pins Used by Serial Interfaces

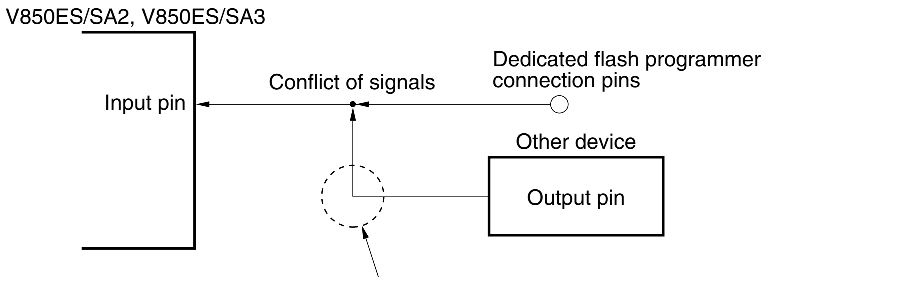
Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$
CSI0 + HS	SO0, SI0, $\overline{SCK0}$, PDH0
UART0	TXD0, RXD0

When connecting a dedicated flash programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(1) Conflict of signals

When the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

Figure 21-9. Conflict of Signals (Serial Interface Input Pin)

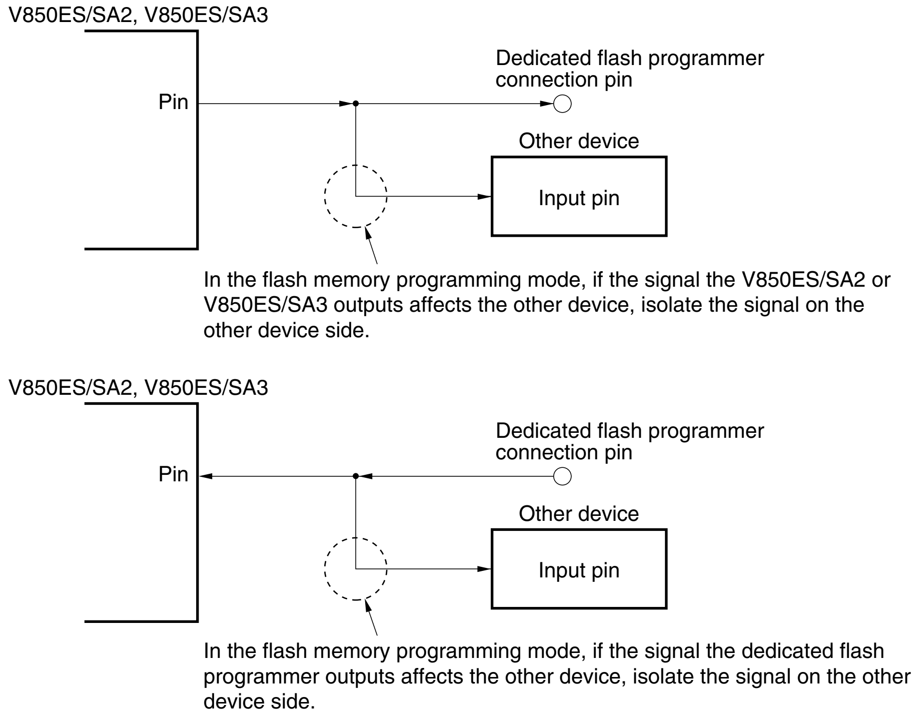


In the flash memory programming mode, the signal that the dedicated flash programmer sends out conflicts with signals another device outputs. Therefore, isolate the signals on the other device side.

(2) Malfunction of other device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.

Figure 21-10. Malfunction of Other Device

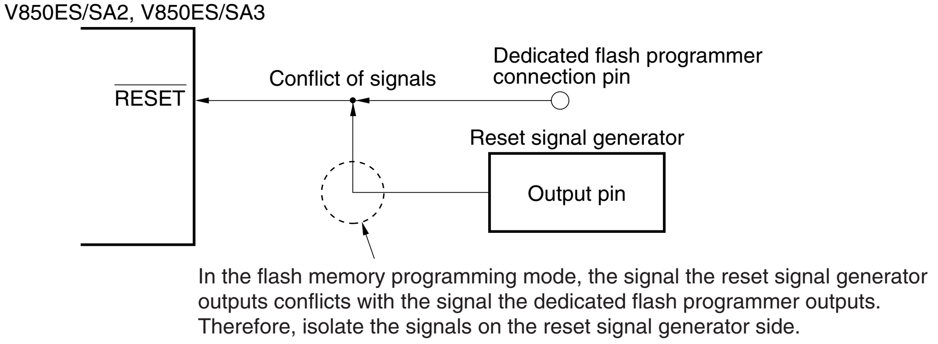

V850ES/SA2, V850ES/SA3
Pin
Dedicated flash programmer connection pin
Other device
Input pin
In the flash memory programming mode, if the signal the V850ES/SA2 or V850ES/SA3 outputs affects the other device, isolate the signal on the other device side.
V850ES/SA2, V850ES/SA3
Pin
Dedicated flash programmer connection pin
Other device
Input pin
In the flash memory programming mode, if the signal the dedicated flash programmer outputs affects the other device, isolate the signal on the other device side.

21.5.4 $\overline{\text{RESET}}$ pin

When the reset signals of the dedicated flash programmer are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

Figure 21-11. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, the signal the reset signal generator outputs conflicts with the signal the dedicated flash programmer outputs. Therefore, isolate the signals on the reset signal generator side.

21.5.5 Port pins (including NMI)

When the flash memory programming mode is set, all the port pins except the pins that communicate with the dedicated flash programmer enter the output high-impedance status. If problems such as disabling output high-impedance status should occur in the external devices connected to the port, connect the port pins to V_{DD} or V_{SS} via resistors.

21.5.6 Other signal pins

Connect X1, X2, XT1, XT2, AV_{REF0}, and AV_{REF1} to the same status as that in the normal operation mode.

21.5.7 Power supply

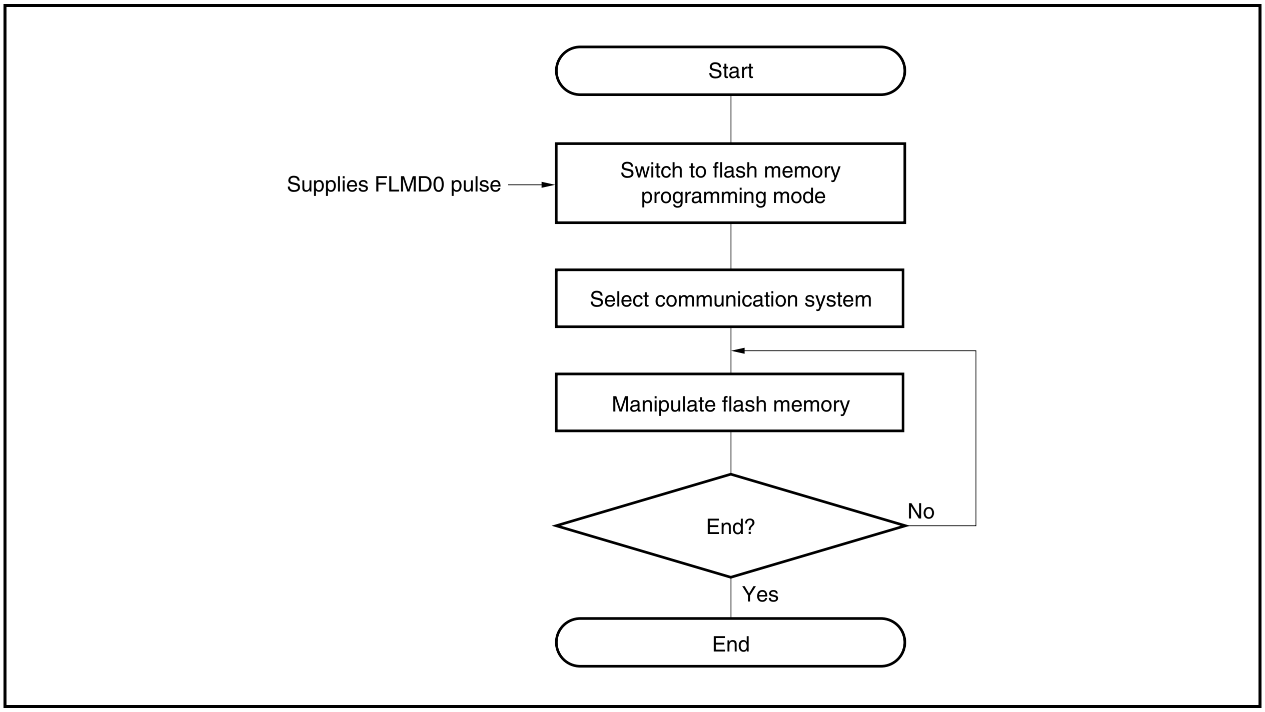
Supply the same power (V_{DD}, V_{SS}, EV_{DD}, EV_{SS}, AV_{DD}, AV_{SS}) as in normal operation mode.

21.6 Programming Method

21.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 21-12. Procedure for Manipulating Flash Memory

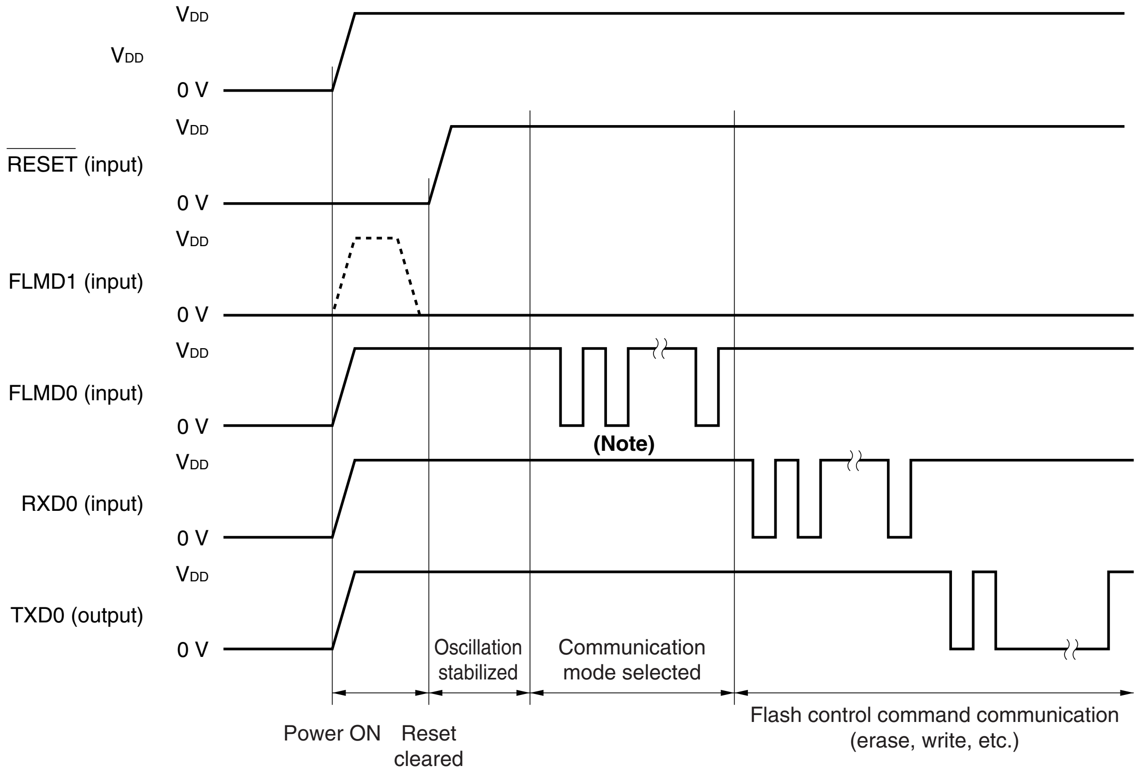

Start
Supplies FLMD0 pulse
Switch to flash memory programming mode
Select communication system
Manipulate flash memory
End?
No
Yes
End

21.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850ES/SA2 or V850ES/SA3 to the flash memory programming mode. When switching modes, set the FLMD0 and FLMD1 pins before releasing reset.

When performing on-board writing, change modes using a jumper, etc.

Figure 21-13. Flash Memory Programming Mode



Note The number of clocks to be inserted differs depending on the communication mode. For details, refer to **Table 21-9**.

21.6.3 Selection of communication mode

In the V850ES/SA2 and V850ES/SA3, the communication mode is selected by inputting pulses (16 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

Table 21-9. List of Communication Modes

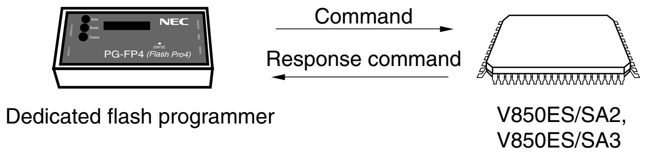
FLMD0 Pulse	Communication Mode	Remarks
0	CSI0	V850ES/SA2 and V850ES/SA3 perform slave operation, MSB first
3	CSI0 + HS	V850ES/SA2 and V850ES/SA3 perform slave operation, MSB first
8	UART0	Communication rate: 9,600 bps (at reset), LSB first
Others	RFU	Setting prohibited

Caution When UART is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the FLMD0 pulse.

21.6.4 Communication command

The V850ES/SA2 and V850ES/SA3 communicate with the dedicated flash programmer by means of commands. The command sent from the dedicated flash programmer to the V850ES/SA2 or V850ES/SA3 is called a “command”. The response signal sent from the V850ES/SA2 or V850ES/SA3 to the dedicated flash programmer is called a “response command”.

Figure 21-14. Communication Command



The following shows the commands for flash memory control of the V850ES/SA2 and V850ES/SA3. All of these commands are issued from the dedicated flash programmer, and the V850ES/SA2 and V850ES/SA3 perform the various processing corresponding to the commands.

Table 21-10. Flash Memory Control Command

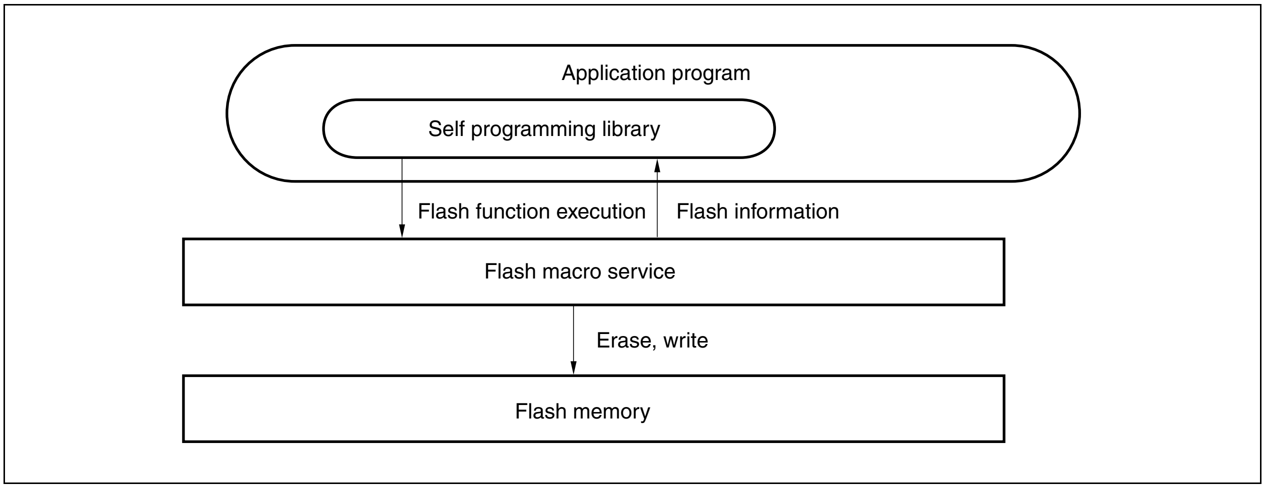
Category	Command Name	Support			Function
		CSI0	CSI0 + HS	UART0	
Blank check	Block blank check command	√	√	√	Checks the erase state of the entire memory.
Erase	Chip erase command	√	√	√	Erases the contents of the entire memory.
	Block erase command	√	√	√	Erases the contents of the specified block memory.
Write	Write command	√	√	√	Writes data according to the specification of the write address and the number of bytes to be written, and executes a verify check.
Verify	Verify command	√	√	√	Compares the contents of the entire memory and the input data.
System setting and control	Reset command	√	√	√	Escapes from each state.
	Oscillating frequency setting command	√	√	√	Sets the oscillation frequency.
	Baud rate setting command	–	–	√	Sets the baud rate when using UART.
	Silicon signature command	√	√	√	Reads the silicon signature information.
	Version acquisition command	√	√	√	Reads the version information of the device.
	Status command	√	√	–	Acquires the operation status.
	Security setting command	√	√	√	Erases chip and blocks, and sets security of write.

★ ## 21.7 Rewriting by Self Programming

21.7.1 Overview

The V850ES/SA2 and V850ES/SA3 support a flash macro service that allows the internal flash memory to be rewritten by the user program itself. By using this interface and a self programming library that is used to rewrite the flash memory via a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data can be rewritten in the field.

Figure 21-15. Concept of Self Programming



21.7.2 Features

(1) Interrupt support

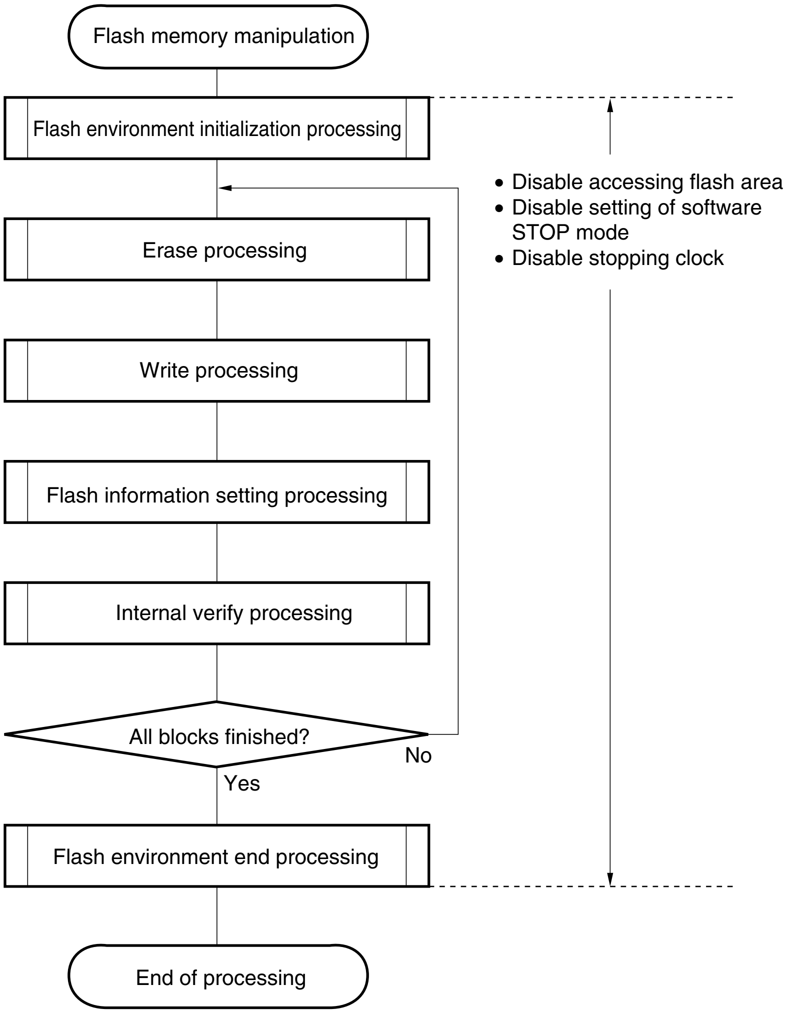
Instructions cannot be fetched from the flash memory during self programming. Conventionally, therefore, a user handler written to the flash memory could not be used even if an interrupt occurred. With the V850ES/SA2 and V850ES/SA3, however, a user handler can be registered to an entry RAM area by using a library function, so that interrupt servicing can be performed by internal RAM or external memory execution.

(2) Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

If a security setting is not performed, flash information setting processing does not have to be executed.

Figure 21-16. Standard Self Programming Flow


Flash memory manipulation
Flash environment initialization processing
Erase processing
Write processing
Flash information setting processing
Internal verify processing
All blocks finished?
No
Yes
Flash environment end processing
End of processing
• Disable accessing flash area
• Disable setting of software STOP mode
• Disable stopping clock

(3) Flash functions

Table 21-12. Flash Function List

Function Name	Description	Support
FlashEnv	Initialization of flash control macro	○
FlashBlockErase	Erasure of only specified block	○
FlashWordWrite	Writing from specified address	○
FlashBlockIVerify	Internal verification of specified block	○
FlashBlockBlankCheck	Blank check of specified block	○
FlashFLMDCheck	Check of FLMD pin	○
FlashStatusCheck	Status check of operation specified immediately before	○
FlashGetInfo	Reading of flash information	○
FlashSetInfo	Setting of flash information	○
FlashBootSwap	Swapping of boot area	×
FlashSetUserHandler	User interrupt handler registration function	○

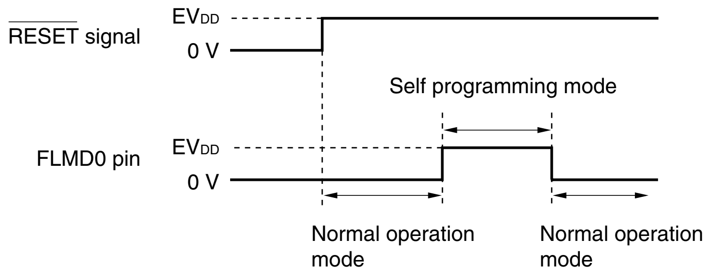
Remark ○: Supported
×: Not supported

(4) Pin processing

(a) FLMD0 pin

Keep the voltage applied to the FLMD0 pin at 0 V after reset release and while normal operation is performed. Keep the voltage of the FLMD0 pin at the EV_{DD} level voltage only in the self programming mode (the voltage must be stabilized during self programming). When self programming has been completed, the voltage of the FLMD0 pin must be returned to 0 V.

Figure 21-17. Mode Switch Timing



(5) Internal resources used

The following table lists the internal resources used for self programming. These internal resources can also be used freely for purposes other than self programming.

Table 21-13. Internal Resources Used

Resource Name	Description
Entry RAM area (124 bytes in either internal RAM or external RAM)	Routines and parameters used for the flash macro service are located in this area. The entry program and default parameters are copied by calling a library initialization function.
Stack area (user stack + 300 bytes)	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Library code (1900 bytes)	Program entity of library (can be used anywhere other than the flash memory block to be manipulated).
Application program	Executed as user application. Calls flash functions.
Maskable interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status, the interrupt servicing start address must be registered in advance by a registration function.
NMI interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status, the interrupt servicing start address must be registered in advance by a registration function.

★

CHAPTER 22 ELECTRICAL SPECIFICATIONS

Absolute maximum ratings (T_A = 25°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +3.5	V
	AV_{DD}		–0.5 to +3.5	V
	EV_{DD}		–0.5 to +3.3	V
	AV_{SS}		–0.5 to +0.5	V
	EV_{SS}		–0.5 to +0.5	V
Input voltage	V_I	Other than X1, XT1, and port 7	–0.5 to EV_{DD} + 0.3$^{\text{Note}}$	V
Clock input voltage	V_K	X1	–0.5 to V_{DD} + 0.3$^{\text{Note}}$	V
	V_{KT}	XT1	–0.5 to V_{DD} + 0.3$^{\text{Note}}$	V
Analog input voltage	V_{IAN}	Port 7	–0.5 to AV_{DD} + 0.3$^{\text{Note}}$	V
Analog reference voltage	AV_{REF}	AV_{REF0}, AV_{REF1}	–0.5 to AV_{DD} + 0.3$^{\text{Note}}$	V
Output current, low	I_{OL}	Per pin	4	mA
		Total for all pins	50	mA
Output current, high	I_{OH}	Per pin	–4	mA
		Total for all pins	–50	mA
Output voltage	V_O	V_{DD} = 2.5 V ±0.2 V	–0.5 to V_{DD} + 0.3 V$^{\text{Note}}$	V
Operating ambient temperature	T_A	Normal operation mode	–40 to +85	°C
		Flash programming mode	–20 to +85	°C
Storage temperature	T_{stg}		–40 to +125	°C

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Capacitance (T_A = 25°C, V_{DD} = AV_{DD} = EV_{DD} = V_{SS} = AV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	fx = 1 MHz Unmeasured pins returned to 0 V			10	pF
I/O capacitance	C_{IO}				10	pF
Output capacitance	C_O				10	pF

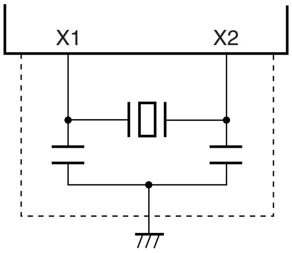
Operating conditions (V_{DD} = AV_{DD} = EV_{DD})

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal system clock frequency	f_{CLK}	@ V_{DD} = 2.2 to 2.7 V, operation with main clock	0.0625		20	MHz
		@ V_{DD} = 2.2 to 2.7 V, operation with subclock	32		35	kHz

Recommended oscillator

(1) Main clock oscillator (T_A = –40 to +85°C)

(a) Connection of ceramic resonator or crystal resonator



Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_X	V_{DD} = 2.2 to 2.7 V	2		20	MHz
Oscillation stabilization time		Upon reset release		$2^{19}/f_X$		s
		Upon STOP mode release		**Note**		s

Note The TYP. value differs depending on the setting of the oscillation stabilization time select register (OSTS).

Caution Ensure that the duty of the oscillation waveform is between 45% and 55%.

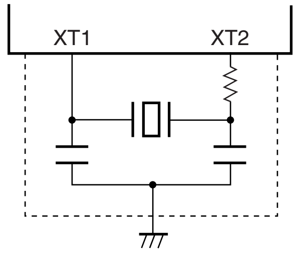
Remarks 1. Connect the oscillator as close as possible to the X1 and X2 pins.

2. Do not route the wiring near broken lines.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

(2) Subclock oscillator (T_A = –40 to +85°C)

(a) Connection of crystal resonator



Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{XT}		32	32.768	35	kHz
Oscillation stabilization time				10		s

Caution Ensure that the duty of the oscillation waveform is between 45% and 55%.

Remarks 1. Connect the oscillator as close as possible to the XT1 and XT2 pins.

2. Do not route the wiring near broken lines.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC characteristics

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V) (1/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**		$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	**Note 2**		$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	**Note 3**		$0.7AV_{DD}$		AV_{DD}	V
	V_{IH4}	X1		$0.8V_{DD}$		V_{DD}	V
	V_{IH5}	XT1, XT2		$0.8V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**		EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	**Note 2**		EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	**Note 3**		AV_{SS}		$0.3AV_{DD}$	V
	V_{IL4}	X1		V_{SS}		$0.2V_{DD}$	V
	V_{IL5}	XT1, XT2		V_{SS}		$0.2V_{DD}$	V
Output voltage, high	V_{OH1}	**Note 4**	I_{OH} = –1 mA	$0.7EV_{DD}$			V
Output voltage, low	V_{OL1}	**Note 4** (except pins P40 and P42)	I_{OL} = 1.6 mA			0.4	V
	V_{OL2}	P40, P42	I_{OL} = 3 mA			0.4	V
Input leakage current, high	I_{LIH}	V_{IN} = V_{DD} = EV_{DD}				5	μA
Input leakage current, low	I_{LIL}	V_{IN} = 0 V				–5	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD} = EV_{DD}				5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–5	μA
Pull-up resistor	R_L	V_{IN} = 0 V		10	30	100	kΩ

Notes 1. P21, P31, P90, P91, P94 to P97, P99, P911, P914, PCD1 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDH0 to PDH7, PDL0 to PDL15 (and their alternate-function pins)

2. $\overline{\text{RESET}}$, P00 to P05, P20, P22, P30, P32, P40 to P46, P92, P93, P98, P910, P912, P913, P915 (and their alternate-function pins)

3. P70 to P715, P80, P81

4. P00 to P05, P20 to P22, P30 to P32, P40 to P46, P90 to P915, PCD1 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDH0 to PDH7, PDL0 to PDL15 (and their alternate-function pins)

DC characteristics

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V) **(2/3)**

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note] V850ES/SA2 (μPD70F3201, μPD70F3201Y) V850ES/SA3 (μPD70F3204, μPD70F3204Y)	I_{DD1}	Normal operation All peripheral functions operating	V_{DD} = 2.2 to 2.7 V@2 to 20 MHz			0.8 × f_{XX} + 2.5	1.58 × f_{XX} + 6.0	mA
				f_{XX} = 2 MHz		4.1	9.2	mA
				f_{XX} = 4 MHz		5.7	12.3	mA
				f_{XX} = 8 MHz		8.9	18.6	mA
				f_{XX} = 10 MHz		10.5	21.8	mA
				f_{XX} = 12 MHz		12.1	25.0	mA
				f_{XX} = 17 MHz		16.1	31.9	mA
				f_{XX} = 20 MHz		18.5	37.6	mA
	I_{DD2}	HALT mode All peripheral functions operating	V_{DD} = 2.2 to 2.7 V@2 to 20 MHz			0.37 × f_{XX} + 2.0	0.7 × f_{XX} + 3.0	mA
				f_{XX} = 2 MHz		2.7	4.4	mA
				f_{XX} = 4 MHz		3.5	5.8	mA
				f_{XX} = 8 MHz		5.0	8.6	mA
				f_{XX} = 10 MHz		6.2	10.0	mA
				f_{XX} = 12 MHz		6.9	11.4	mA
				f_{XX} = 17 MHz		8.2	14.9	mA
				f_{XX} = 20 MHz		9.4	17.0	mA
	I_{DD3}	IDLE mode RTC operation	V_{DD} = 2.2 to 2.7 V@2 to 20 MHz			40 × f_{XX} + 300	70 × f_{XX} + 500	μA
				f_{XX} = 2 MHz		380	640	μA
				f_{XX} = 4 MHz		460	780	μA
				f_{XX} = 8 MHz		620	1060	μA
				f_{XX} = 10 MHz		700	1200	μA
				f_{XX} = 12 MHz		780	1340	μA
				f_{XX} = 17 MHz		980	1690	μA
				f_{XX} = 20 MHz		1100	1900	μA
	I_{DD4}	STOP mode	Sub oscillator, Only RTC operates	T_A = 25°C		4	20	μA
				T_A = 85°C		–	60	μA
			Sub oscillator stopped (XT1 = V_{SS})	T_A = 25°C		1	10	μA
				T_A = 85°C		–	50	μA
	I_{DD5}	Subclock operation mode (f_{XT} = 32.768 kHz)				150	250	μA
	I_{DD6}	Sub-IDLE mode (f_{XT} = 32.768 kHz) Main clock oscillator stopped, RTC operates		T_A = 25°C		4	20	μA
				T_A = 85°C		–	60	μA

Note The typical value of V_{DD} is 2.5 V. The value does not include the current consumed at the output buffer.

Remark f_{XX}: Main clock frequency (MHz)

DC characteristics

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V) **(3/3)**

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note] V850ES/SA2 (μPD703200, μPD703200Y μPD703201, μPD703201Y) V850ES/SA3 (μPD703204, μPD703204Y)	I_{DD1}	Normal operation All peripheral functions operating	V_{DD} = 2.2 to 2.7 V@2 to 20 MHz			0.7 × f_{XX} + 1.0	1.38 × f_{XX} + 1.5	mA
				f_{XX} = 2 MHz		2.4	4.3	mA
				f_{XX} = 4 MHz		3.8	7.0	mA
				f_{XX} = 8 MHz		6.6	12.5	mA
				f_{XX} = 10 MHz		8.0	15.3	mA
				f_{XX} = 12 MHz		9.4	18.1	mA
				f_{XX} = 17 MHz		12.9	25.0	mA
				f_{XX} = 20 MHz		15.0	29.1	mA
	I_{DD2}	HALT mode All peripheral functions operating	V_{DD} = 2.2 to 2.7 V@2 to 20 MHz			0.36 × f_{XX} + 1.0	0.67 × f_{XX} + 1.5	mA
				f_{XX} = 2 MHz		1.7	2.8	mA
				f_{XX} = 4 MHz		2.4	4.2	mA
				f_{XX} = 8 MHz		3.9	6.9	mA
				f_{XX} = 10 MHz		4.6	8.2	mA
				f_{XX} = 12 MHz		5.3	9.5	mA
				f_{XX} = 17 MHz		7.1	12.9	mA
				f_{XX} = 20 MHz		8.2	14.9	mA
	I_{DD3}	IDLE mode RTC operation	V_{DD} = 2.2 to 2.7 V@2 to 20 MHz			40 × f_{XX} + 300	70 × f_{XX} + 500	μA
				f_{XX} = 2 MHz		380	640	μA
				f_{XX} = 4 MHz		460	780	μA
				f_{XX} = 8 MHz		620	1060	μA
				f_{XX} = 10 MHz		700	1200	μA
				f_{XX} = 12 MHz		780	1340	μA
				f_{XX} = 17 MHz		980	1690	μA
				f_{XX} = 20 MHz		1100	1900	μA
	I_{DD4}	STOP mode	Sub oscillator, Only RTC operates	T_A = 25°C		4	20	μA
				T_A = 85°C		–	60	μA
			Sub oscillator stopped ($XT1 = V_{SS}$)	T_A = 25°C		1	10	μA
				T_A = 85°C		–	50	μA
	I_{DD5}	Subclock operation mode (f_{XT} = 32.768 kHz)				40	100	μA
	I_{DD6}	Sub-IDLE mode (f_{XT} = 32.768 kHz) Main clock oscillator stopped, RTC operates		T_A = 25°C		4	20	μA
				T_A = 85°C		–	60	μA

Note The typical value of V_{DD} is 2.5 V. The value does not include the current consumed at the output buffer.

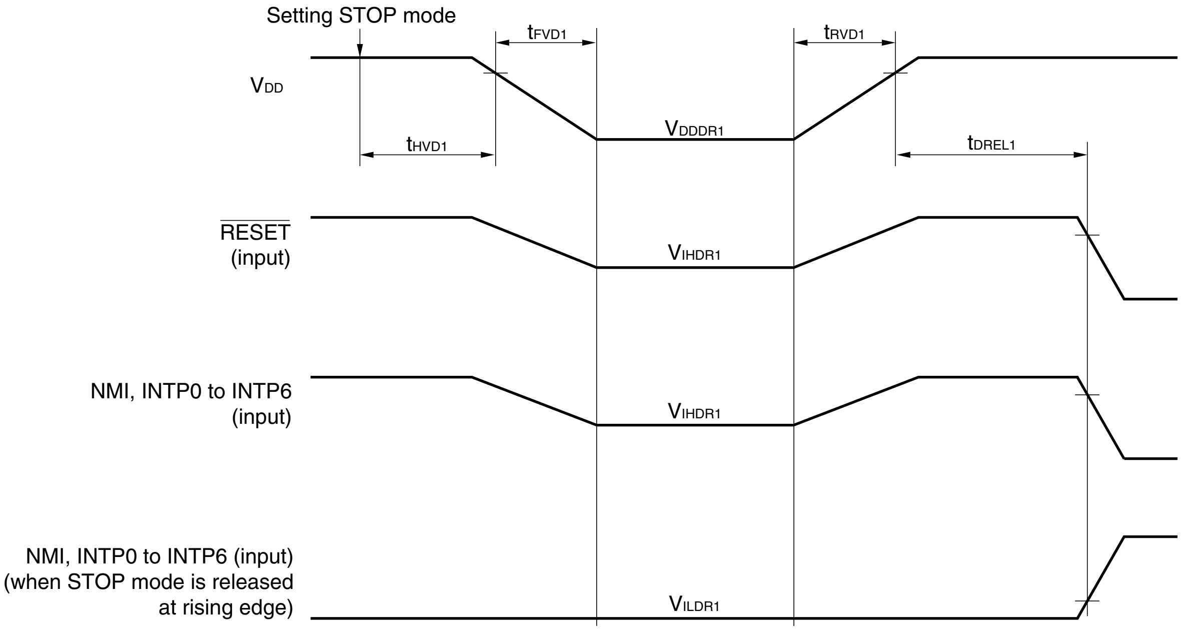
Remark f_{XX}: Main clock frequency (MHz)

Data retention characteristics

(1) In STOP mode (T_A = –40 to +85°C, V_{SS} = AV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR1}	STOP mode	1.5		2.7	V
Data retention current	I_{DDDR1}	V_{DD} = AV_{DD} = EV_{DD} = V_{DDDR1}		1	60	μA
Supply voltage rise time	t_{RVD1}		200			μs
Supply voltage fall time	t_{FVD1}		200			μs
Supply voltage hold time (from STOP mode setting)	t_{HVD1}		0			ms
STOP release signal input time	t_{DREL1}		0			ms
Data retention high-level input voltage	V_{IHDR1}	All input ports	V_{IHn}		V_{DDDR1}	V
Data retention low-level input voltage	V_{ILDR1}	All input ports	0		V_{ILn}	V

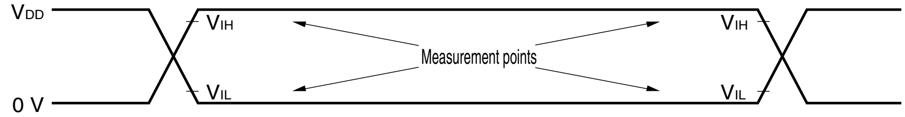
Remark n = 1 to 5


Setting STOP mode
t_{FVD1}
t_{RVD1}
V_{DD}
V_{DDDR1}
t_{HVD1}
t_{DREL1}
$\overline{\text{RESET}}$ (input)
V_{IHDR1}
NMI, INTP0 to INTP6 (input)
V_{IHDR1}
NMI, INTP0 to INTP6 (input) (when STOP mode is released at rising edge)
V_{ILDR1}

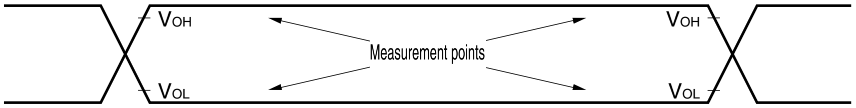
Caution Shifting to STOP mode and restoring from STOP mode must be performed at V_{DD} = 2.2 V min. (f_{CLK} = 20 MHz).

AC characteristics

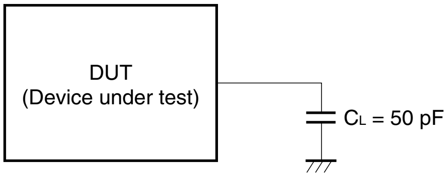
AC test input measurement points (V_{DD}, AV_{DD}, EV_{DD})



AC test output measurement points



Load conditions



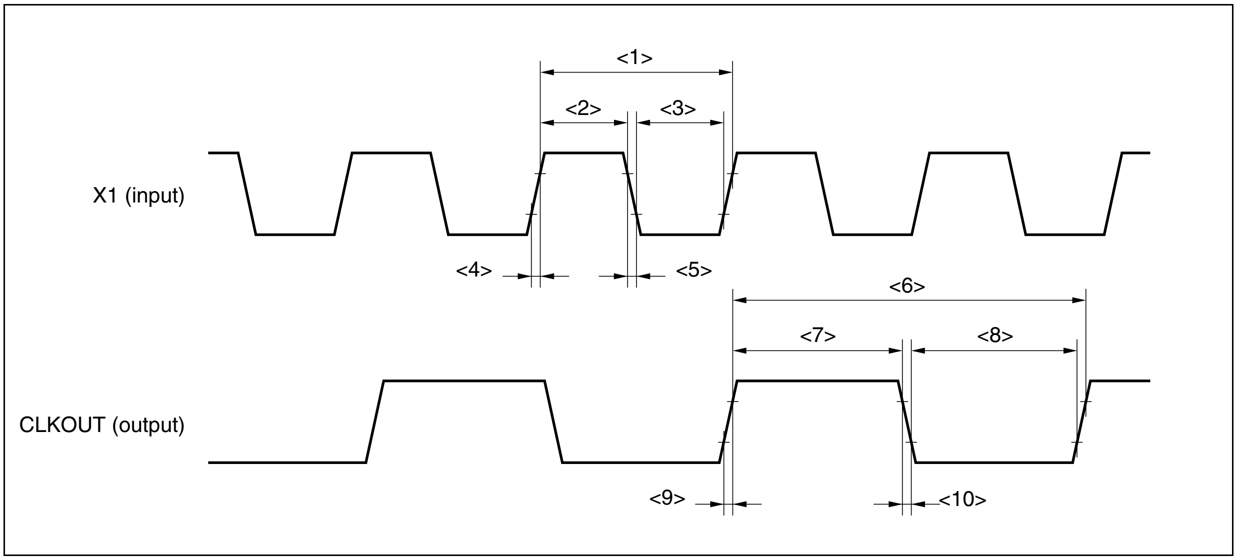
Caution If the load capacitance exceeds 50 pF due to the circuit configuration, reduce the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

Clock timing

(1) Operating conditions (T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
X1 input cycle	t_{CYX}	<1>		50	500	ns
X1 input high-level width	t_{WXH}	<2>		25	250	ns
X1 input low-level width	t_{WXL}	<3>		25	250	ns
X1 input rise time	t_{XR}	<4>			0.5(t_{CYX} – t_{WXH} – t_{WXL})	ns
X1 input fall time	t_{XF}	<5>			0.5(t_{CYX} – t_{WXH} – t_{WXL})	ns
CLKOUT output cycle	t_{CYK}	<6>		50 ns	16 μs	
CLKOUT high-level width	t_{WKH}	<7>		t_{CYK}/2 – 5		ns
CLKOUT low-level width	t_{WKL}	<8>		t_{CYK}/2 – 5		ns
CLKOUT rise time	t_{KR}	<9>			5	ns
CLKOUT fall time	t_{KF}	<10>			5	ns

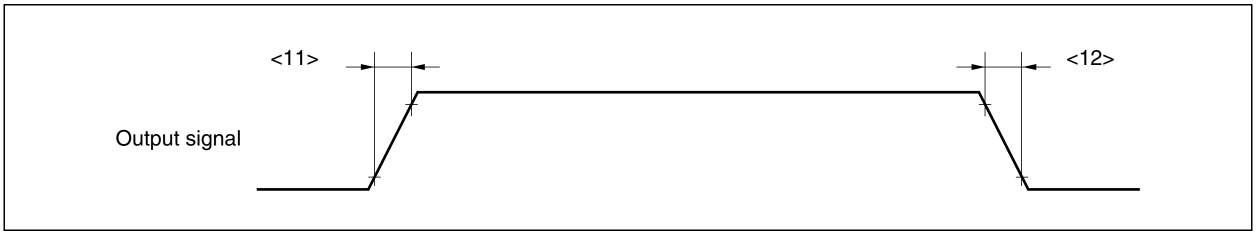
Remark Ensure that the duty of the oscillation waveform is between 45% and 55%.



Timing of output signal (excluding external bus interface)

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<11>			20	ns
Output fall time	t_{OF}	<12>			20	ns



Bus timing

(1) Multiplexed bus mode

(a) CLKOUT asynchronous: In multiplexed bus mode

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	t_{SAST}	<13>		0.5T – 20		ns
Address hold time (from ASTB↓)	t_{HSTA}	<14>		0.5T – 15		ns
Delay time from $\overline{\text{RD}}$↓ to address float	t_{FRDA}	<15>			2	ns
Data input setup time from address	t_{SAID}	<16>			(2 + n)T – 30	ns
Data input setup time from $\overline{\text{RD}}$↓	t_{SRID}	<17>			(1 + n)T – 25	ns
Delay time from ASTB↓ to $\overline{\text{RD}}$↓, $\overline{\text{WRm}}$↓	$t_{DSTRDWR}$	<18>		0.5T – 15		ns
Data input hold time (from $\overline{\text{RD}}$↑)	t_{HRDID}	<19>		0		ns
Address output time from $\overline{\text{RD}}$↑	t_{DRDA}	<20>		(1 + i)T – 15		ns
Delay time from $\overline{\text{RD}}$, $\overline{\text{WRm}}$↑ to ASTB↑	$t_{DRDWRST}$	<21>		0.5T – 15		ns
Delay time from $\overline{\text{RD}}$↑ to ASTB↓	t_{DRDST}	<22>		(1.5 + i)T – 15		ns
$\overline{\text{RD}}$, $\overline{\text{WRm}}$ low-level width	t_{WRDWRL}	<23>		(1 + n)T – 15		ns
ASTB high-level width	t_{WSTH}	<24>		T – 15		ns
Data output time from $\overline{\text{WRm}}$↓	t_{DWROD}	<25>			15	ns
Data output setup time (to $\overline{\text{WRm}}$↑)	t_{SODWR}	<26>		(1 + n)T – 20		ns
Data output hold time (from $\overline{\text{WRm}}$↑)	t_{HWROD}	<27>		T – 15		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{SAWT1}	<28>	n ≥ 1		1.5T – 30	ns
	t_{SAWT2}	<29>	n ≥ 1		(1.5 + n)T – 30	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT1}	<30>	n ≥ 1	(0.5 + n)T		ns
	t_{HAWT2}	<31>	n ≥ 1	(1.5 + n)T		ns
$\overline{\text{WAIT}}$ setup time (to ASTB↓)	t_{SSTWT1}	<32>	n ≥ 1		T – 25	ns
	t_{SSTWT2}	<33>	n ≥ 1		(1 + n)T – 25	ns
$\overline{\text{WAIT}}$ hold time (from ASTB↓)	t_{HSTWT1}	<34>	n ≥ 1	nT		ns
	t_{HSTWT2}	<35>	n ≥ 1	(1 + n)T		ns
$\overline{\text{HLDRQ}}$ high-level width	t_{WHQH}	<36>		T + 10		ns
$\overline{\text{HLDAK}}$ low-level width	t_{WHAL}	<37>		T – 15		ns
Delay time from $\overline{\text{HLDAK}}$↑ to bus output	t_{DHAC}	<38>		–3		ns
Delay time from $\overline{\text{HLDRQ}}$↓ to $\overline{\text{HLDAK}}$↓	t_{DHQHA1}	<39>			(2n + 7.5)T + 25	ns
Delay time from $\overline{\text{HLDRQ}}$↑ to $\overline{\text{HLDAK}}$↑	t_{DHQHA2}	<40>		0.5T	1.5T + 25	ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

2. n: Number of wait clocks inserted in the bus cycle.
The sampling timing changes when a programmable wait is inserted.

3. m = 0, 1

4. i: Number of idle states inserted after the read cycle (0 or 1).

5. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

(b) CLKOUT synchronous: In multiplexed bus mode

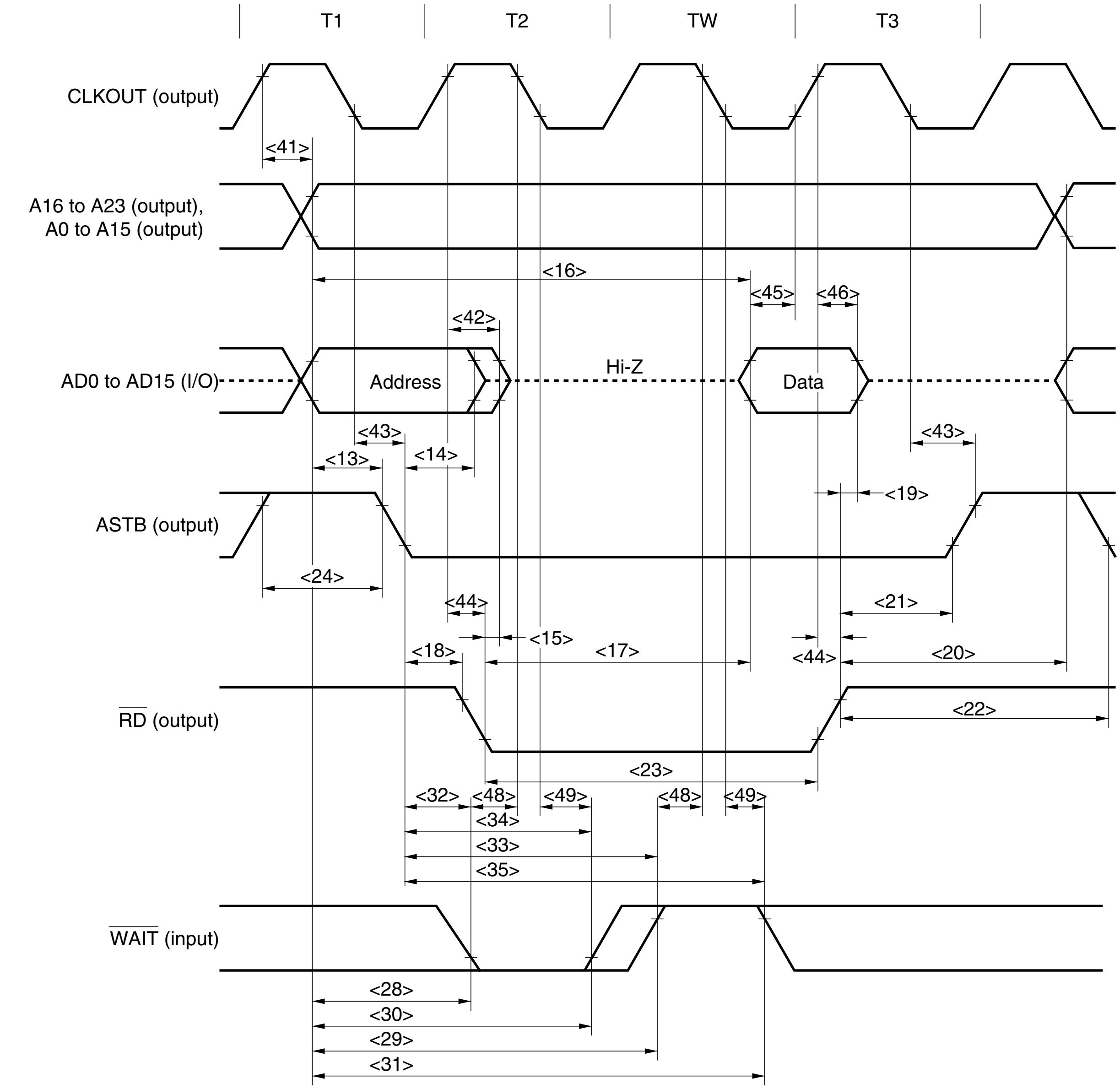
(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA}	<41>		0	19	ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<42>		0	19	ns
Delay time from CLKOUT↓ to ASTB	t_{DKST}	<43>		–5	14	ns
Delay time from CLKOUT↑ to $\overline{RD}$, $\overline{WRm}$	t_{DKRDWR}	<44>		–9	10	ns
Data input setup time (to CLKOUT↑)	t_{SIDK}	<45>		15		ns
Data input hold time (from CLKOUT↑)	t_{HKID}	<46>		5		ns
Data output delay time from CLKOUT↑	t_{DKOD}	<47>			19	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	t_{SWTK}	<48>		20		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT}	<49>		5		ns
$\overline{HLDRQ}$ setup time (to CLKOUT↓)	t_{SHQK}	<50>		20		ns
$\overline{HLDRQ}$ hold time (from CLKOUT↓)	t_{HKHQ}	<51>		5		ns
Delay time from CLKOUT↑ to bus float	t_{DKF}	<52>			19	ns
Delay time from CLKOUT↑ to $\overline{HLDAK}$	t_{DKHA}	<53>			19	ns

Remarks 1. m = 0, 1

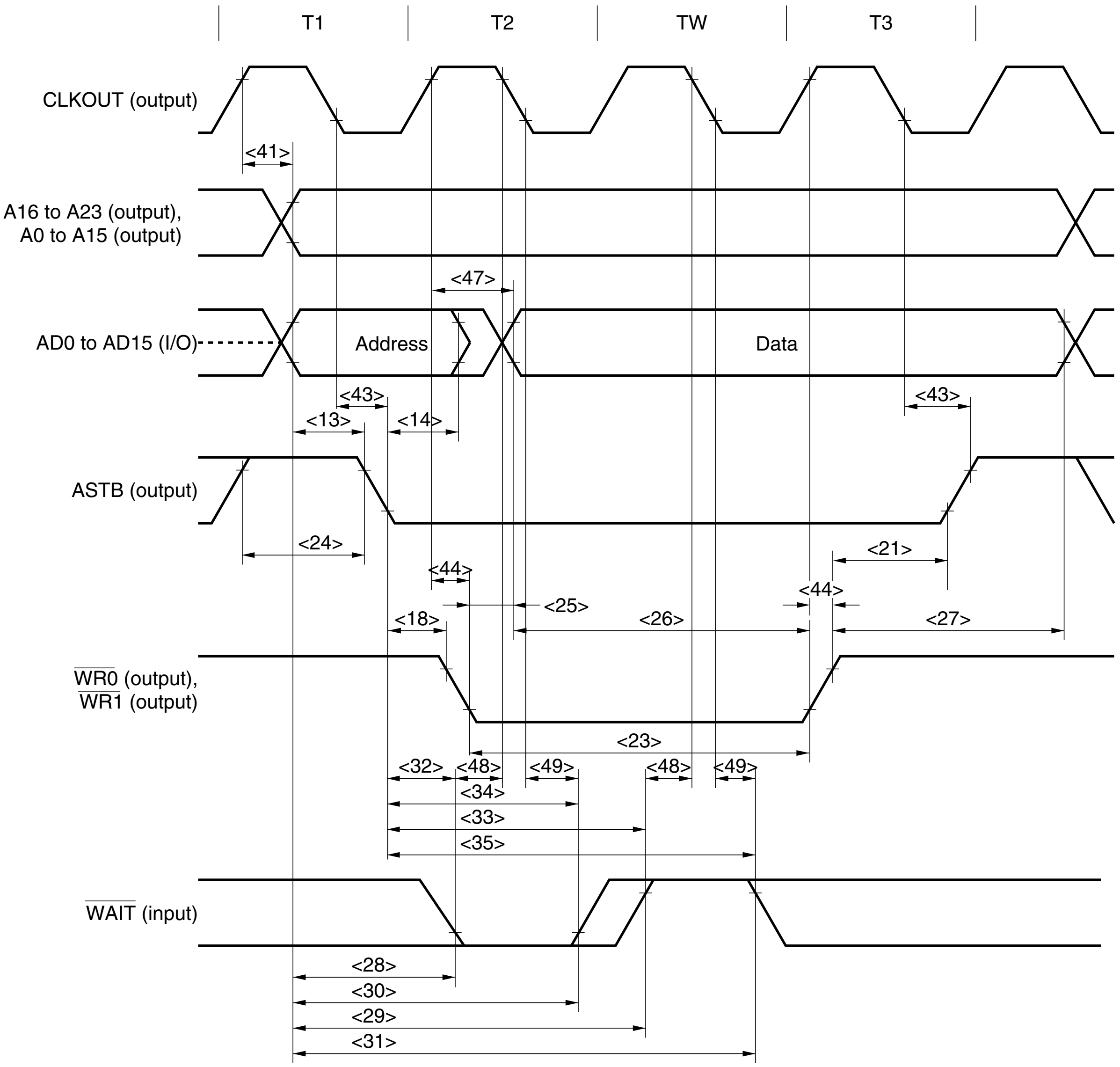
2. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

Read cycle (CLKOUT synchronous/asynchronous, 1 wait): In multiplexed bus mode



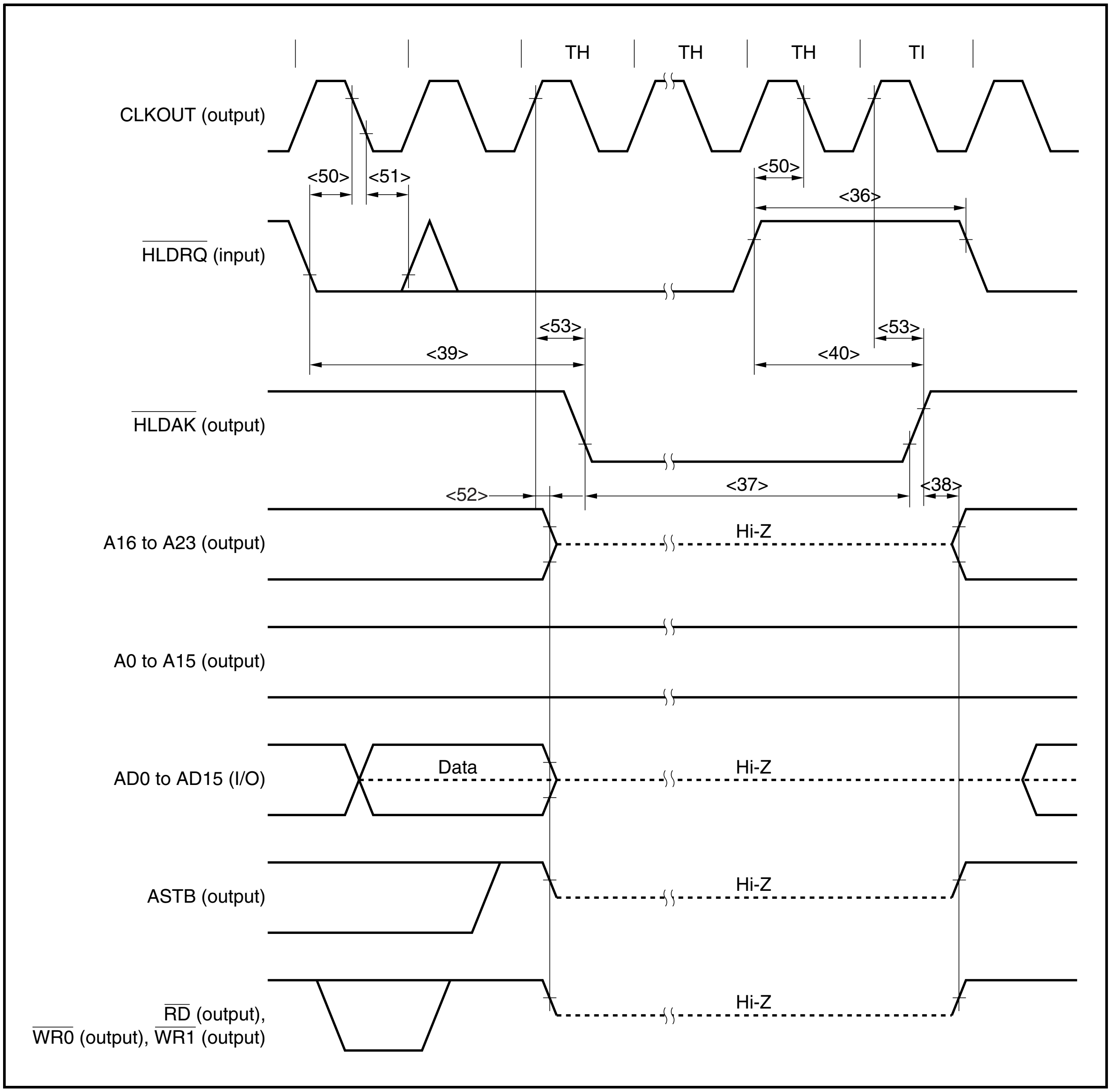
Remark $\overline{\text{WR0}}$ and $\overline{\text{WR1}}$ are high level.

Write cycle (CLKOUT synchronous/asynchronous, 1 wait): In multiplexed bus mode



Remark $\overline{\text{RD}}$ is high level.

Bus hold: In multiplexed bus mode



(2) In separate bus mode

(a) Read cycle (CLKOUT asynchronous): In separate bus mode

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to $\overline{RD}$↓)	t_{SARD}	<54>		0.5T – 20		ns
Address hold time (from $\overline{RD}$↑)	t_{HARD}	<55>		0		ns
$\overline{RD}$ low-level width	t_{WRDL}	<56>		(1.5 + n)T – 10		ns
Data setup time (to $\overline{RD}$↑)	t_{SISD}	<57>		20		ns
Data hold time (from $\overline{RD}$↑)	t_{HISD}	<58>		0		ns
Data setup time (to address)	t_{SAID}	<59>			(2 + n)T – 25	ns
$\overline{WAIT}$ setup time (to $\overline{RD}$↓)	t_{SRDWT1}	<60>			0.5T – 20	ns
	t_{SRDWT2}	<61>			(0.5 + n)T – 20	ns
$\overline{WAIT}$ hold time (from $\overline{RD}$↓)	t_{HRDWT1}	<62>		0.5T		ns
	t_{HRDWT2}	<63>		(0.5 + n)T		ns
$\overline{WAIT}$ setup time (to address)	t_{SAWT1}	<64>			T – 30	ns
	t_{SAWT2}	<65>			(1 + n)T – 30	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT1}	<66>		T		ns
	t_{HAWT2}	<67>		(1 + n)T		ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

2. n: Number of wait clocks inserted in bus cycle
The sampling timing changes when a programmable wait is inserted.

3. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

(b) Read cycle (CLKOUT synchronous): In separate bus mode

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address, CS	t_{DKSA}	<68>		0	19	ns
Data input setup time (to CLKOUT↑)	t_{SISDK}	<69>		20		ns
Data input hold time (from CLKOUT↑)	t_{HKISD}	<70>		5		ns
Delay time from CLKOUT↓↑ to $\overline{RD}$	t_{DKSR}	<71>		0	19	ns
$\overline{WAIT}$ setup time (to CLKOUT↑)	t_{SWTK}	<72>		20		ns
$\overline{WAIT}$ hold time (from CLKOUT↑)	t_{HKWT}	<73>		5		ns

Remark The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

(c) Write cycle (CLKOUT asynchronous): In separate bus mode
(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to $\overline{\text{WRm}}$↓)	t_{SAW}	<74>		T – 20		ns
Address hold time (from $\overline{\text{WRm}}$↑)	t_{HAW}	<75>		0.5T – 10		ns
$\overline{\text{WRm}}$ low-level width	t_{WWRL}	<76>		(0.5 + n)T – 10		ns
Data output time from $\overline{\text{WRm}}$↓	t_{DOSDW}	<77>		–5		ns
Data setup time (to $\overline{\text{WRm}}$↑)	t_{SOSDW}	<78>		(0.5 + n)T – 15		ns
Data hold time (from $\overline{\text{WRm}}$↑)	t_{HOSDW}	<79>		0.5T – 10		ns
Data setup time (to address)	t_{SAOD}	<80>		T – 25		ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{WRm}}$↓)	t_{SWRWT1}	<81>		20		ns
	t_{SWRWT2}	<82>		nT – 20		ns
$\overline{\text{WAIT}}$ hold time (from $\overline{\text{WRm}}$↓)	t_{HWRWT1}	<83>		0		ns
	t_{HWRWT2}	<84>		nT		ns
$\overline{\text{WAIT}}$ setup time (to address)	t_{SAWT1}	<85>			T – 25	ns
	t_{SAWT2}	<86>			(1 + n)T – 20	ns
$\overline{\text{WAIT}}$ hold time (from address)	t_{HAWT1}	<87>		T		ns
	t_{HAWT2}	<88>		(1 + n)T		ns

Remarks 1. m = 0, 1

2. T = 1/f_{CPU} (f_{CPU}: CPU operation clock frequency)

3. n: Number of wait clocks inserted in bus cycle
The sampling timing changes when a programmable wait is inserted.

4. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

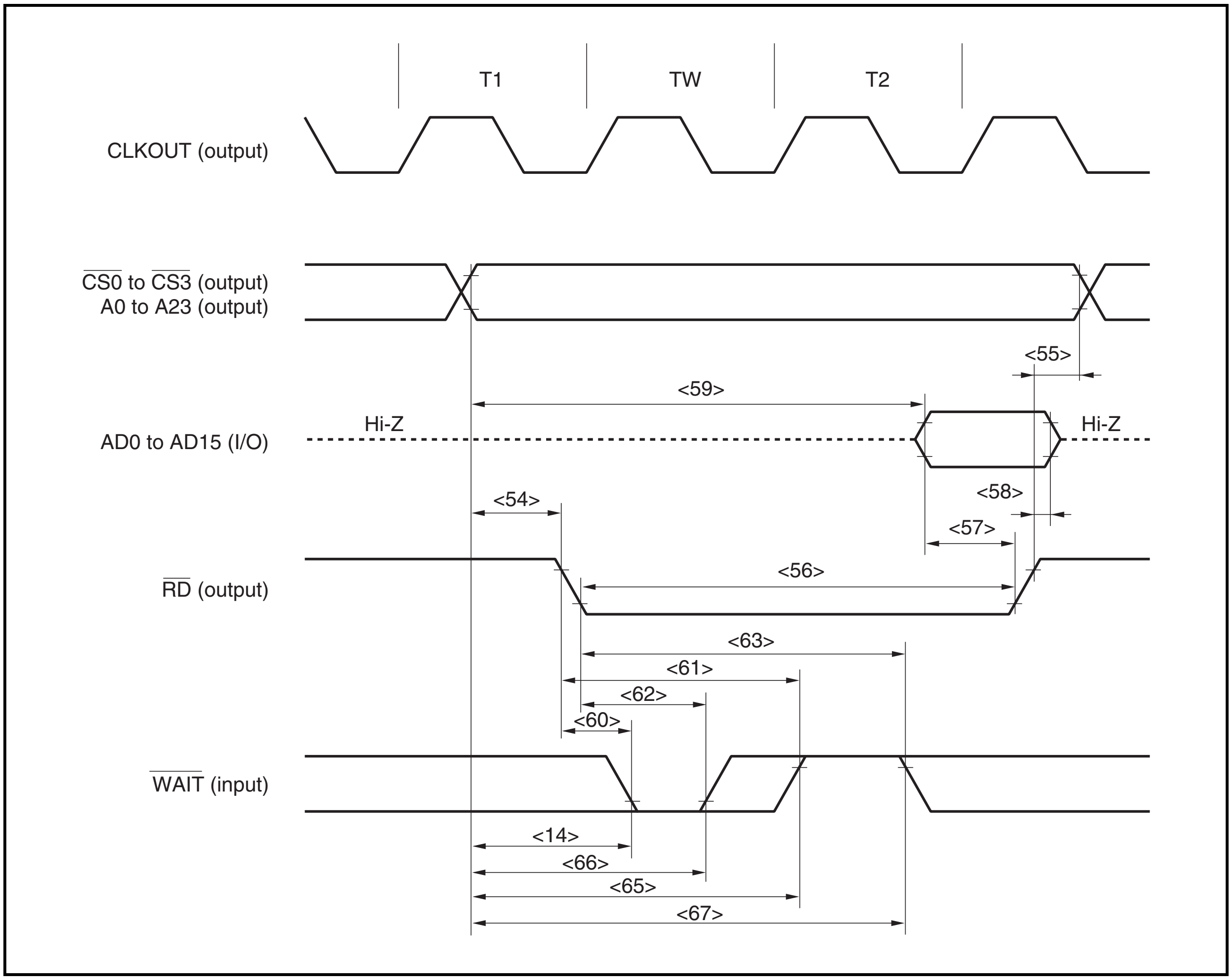
(d) Write cycle (CLKOUT synchronous): In separate bus mode
(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address, CS	t_{DKSA}	<89>		0	19	ns
Delay time from CLKOUT↑ to data output	t_{DKSD}	<90>		0	19	ns
Delay time from CLKOUT↑↓ to $\overline{\text{WRm}}$	t_{DKSW}	<91>		0	19	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↑)	t_{SWTK}	<92>		20		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↑)	t_{HKWT}	<93>		0		ns

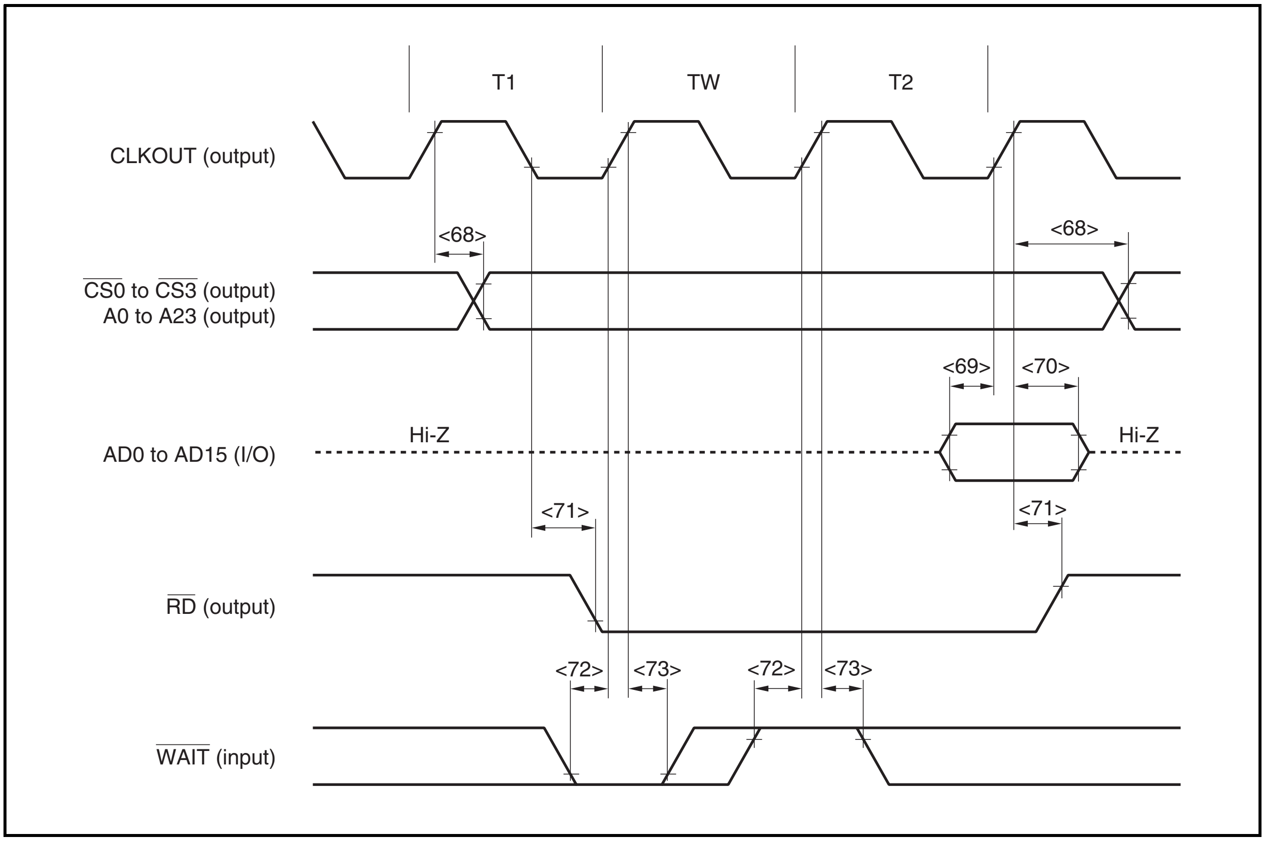
Remarks 1. m = 0, 1

2. The values in the above specifications are the values for when clocks with a 1:1 duty ratio are input from X1.

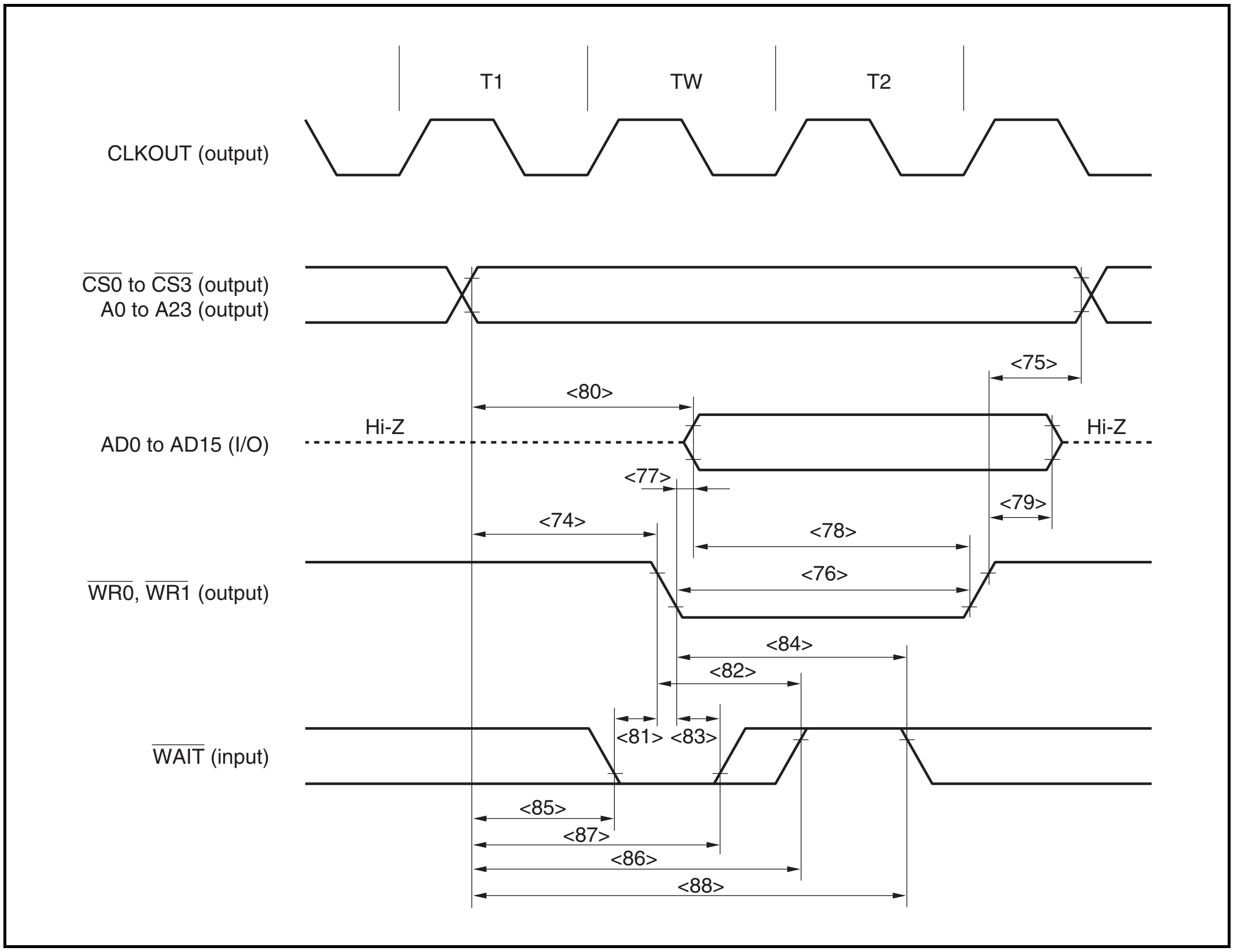
Read cycle (CLKOUT asynchronous, 1 wait): In separate bus mode



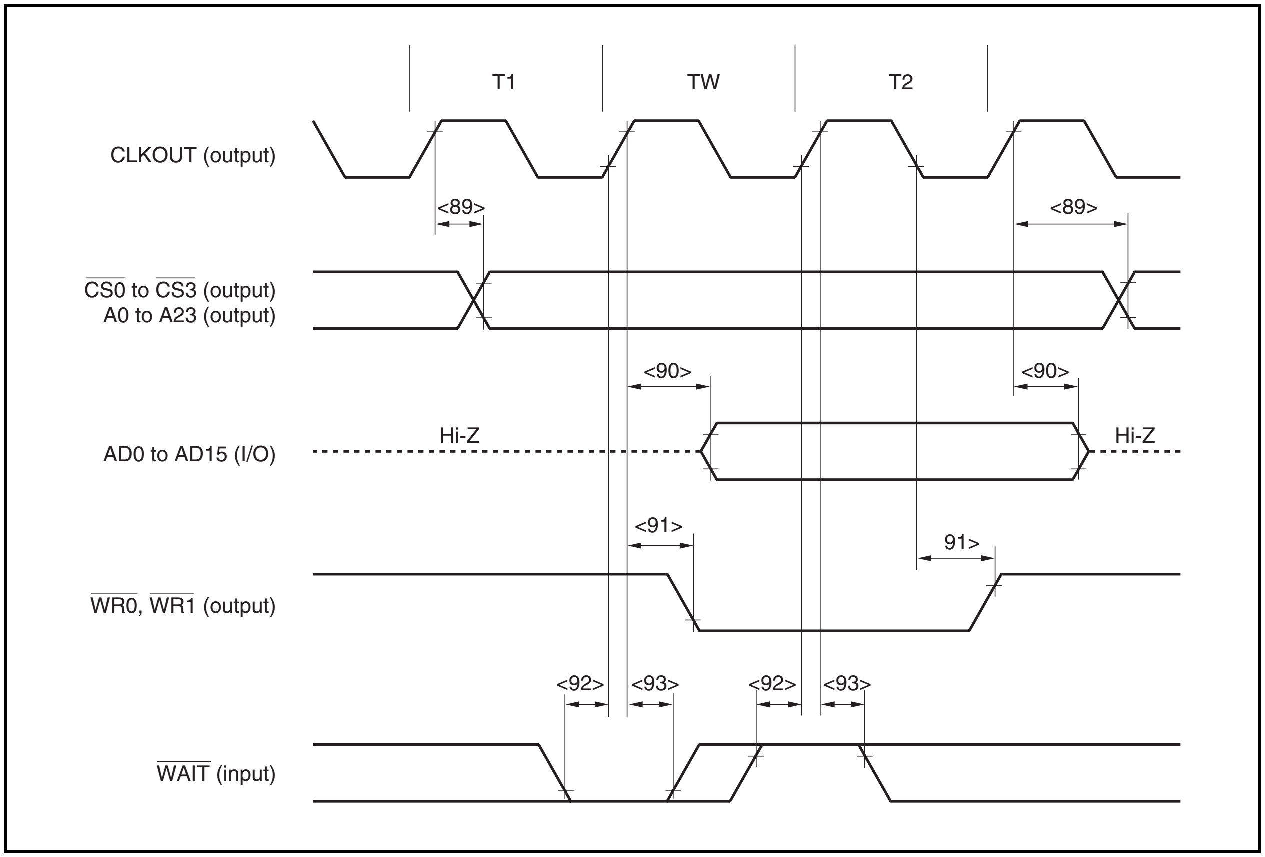
Read cycle (CLKOUT synchronous, 1 wait): In separate bus mode



Write cycle (CLKOUT asynchronous, 1 wait): In separate bus mode



Write cycle (CLKOUT synchronous, 1 wait): In separate bus mode



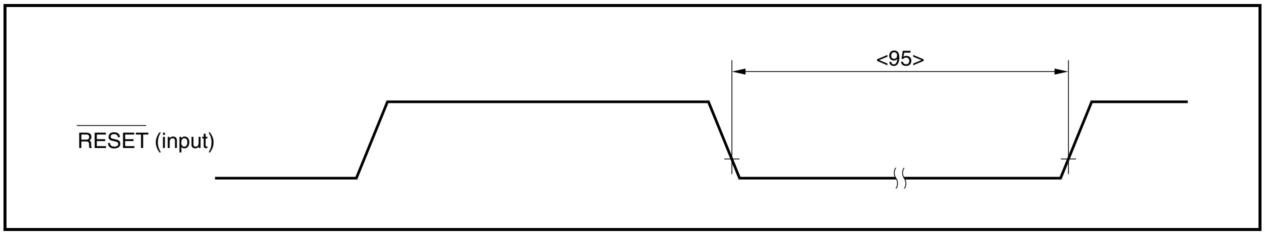
Reset/interrupt timing

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

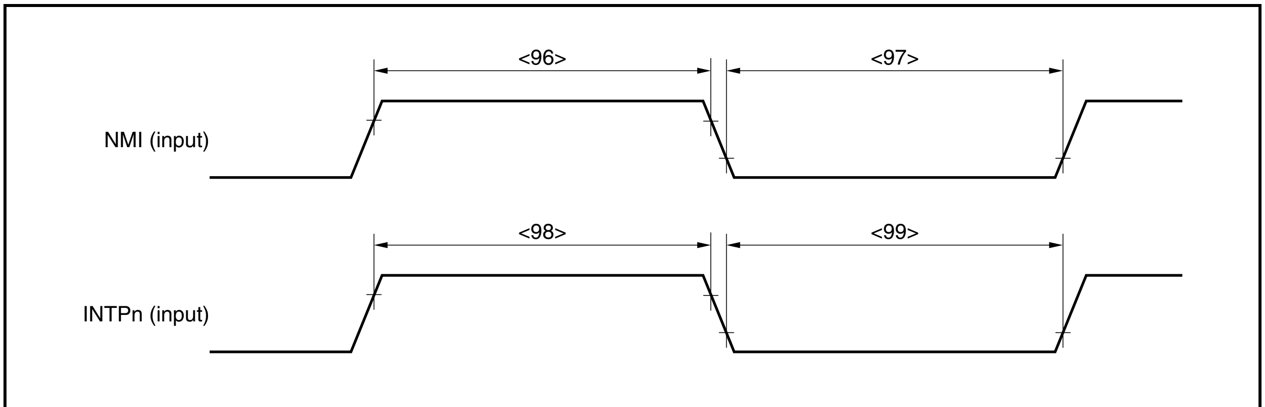
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<95>	At power-on, while erasing/writing flash memory	2		μs
			Other than above	500		ns
NMI high-level width	t_{WNIH}	<96>		500		ns
NMI low-level width	t_{WNIL}	<97>		500		ns
INTPn high-level width	t_{WITH}	<98>	n = 0 to 6 (analog noise elimination)	500		ns
INTPn low-level width	t_{WITL}	<99>	n = 0 to 6 (analog noise elimination)	500		ns

Remark T = 1/fxx

Reset



Interrupt



Remark n = 0 to 6

Timer timing

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TIn high-level width		n = 0, 1	2T + 20		ns
		n = 2 to 5	40		ns
TIn low-level width		n = 0, 1	2T + 20		ns
		n = 2 to 5	40		ns
TCLRn high-level width		n = 0, 1	2T + 20		ns
TCLRn low-level width		n = 0, 1	2T + 20		ns
INTPnm high-level width	t_{WITH}	nm = 00, 01, 10, 11	2T + 20		ns
INTPnm low-level width	t_{WITL}	nm = 00, 01, 10, 11	2T + 20		ns

Remark T = 1/fxx

CSI timing

(1) Master mode

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKn}}$ cycle time	t_{KCY1}	<100>	Output	200		ns
$\overline{\text{SCKn}}$ high-/low-level width	t_{KH1}	<101>	Output	t_{KCY1}/2 – 10		ns
SIn setup time (to $\overline{\text{SCKn}}$↑)	t_{SIK1}	<102>		30		ns
SIn hold time (from $\overline{\text{SCKn}}$↑)	t_{KSI1}	<103>		30		ns
Delay time from $\overline{\text{SCKn}}$↓ to SOn output	t_{KSO1}	<104>			30	ns

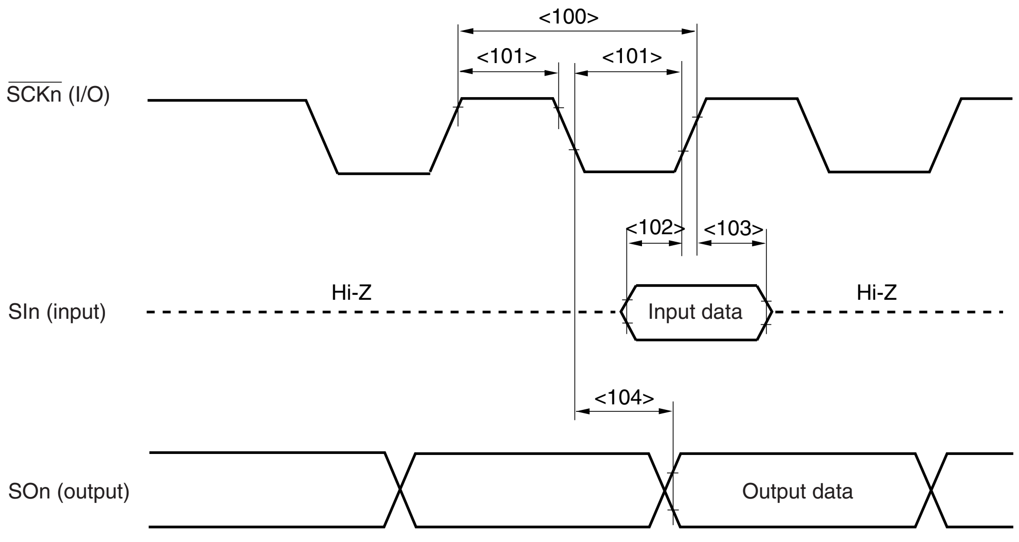
Remark n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

(2) Slave mode

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKn}}$ cycle time	t_{KCY2}	<100>	Output	200		ns
$\overline{\text{SCKn}}$ high-/low-level width	t_{KH2}	<101>	Output	90		ns
SIn setup time (to $\overline{\text{SCKn}}$↑)	t_{SIK2}	<102>		50		ns
SIn hold time (from $\overline{\text{SCKn}}$↑)	t_{KSI2}	<103>		50		ns
Delay time from $\overline{\text{SCKn}}$↓ to SOn output	t_{KSO2}	<104>			50	ns

Remark n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)



Remark n = 0 to 3 (V850ES/SA2), n = 0 to 4 (V850ES/SA3)

I^{2}C bus mode (µPD703200Y, 703201Y, 703204Y, 70F3201Y, 70F3204Y only)

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V)

Parameter		Symbol		Normal Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		f_{CLK}		0	100	0	400	kHz
Bus-free time (between stop/start conditions)		t_{BUF}	<105>	4.7	–	1.3	–	µs
Hold time$^{\text{Note 1}}$		$t_{HD:STA}$	<106>	4.0	–	0.6	–	µs
SCL clock low-level width		t_{LOW}	<107>	4.7	–	1.3	–	µs
SCL clock high-level width		t_{HIGH}	<108>	4.0	–	0.6	–	µs
Setup time for start/restart conditions		$t_{SU:STA}$	<109>	4.7	–	0.6	–	µs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	<110>	5.0	–	–	–	µs
	I^{2}C mode			0$^{\text{Note 2}}$	–	0$^{\text{Note 2}}$	0.9$^{\text{Note 3}}$	µs
Data setup time		$t_{SU:DAT}$	<111>	250	–	100$^{\text{Note 4}}$	–	ns
SDA and SCL signal rise time		t_R	<112>	–	1,000	20 + 0.1Cb$^{\text{Note 5}}$	300	ns
SDA and SCL signal fall time		t_F	<113>	–	300	20 + 0.1Cb$^{\text{Note 5}}$	300	ns
Stop condition setup time		$t_{SU:STO}$	<114>	4.0	–	0.6	–	µs
Pulse width with spike suppressed by input filter		t_{SP}	<115>	–	–	0	50	ns
Capacitance load of each bus line		Cb		–	400	–	400	pF

Notes 1. At the start condition, the first clock pulse is generated after the hold time.

2. The system requires a minimum of 300 ns hold time internally for the SDA signal (at $V_{IHmin.}$ of SCL signal) in order to occupy the undefined area at the falling edge of SCL.

3. If the system does not extend the SCL signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

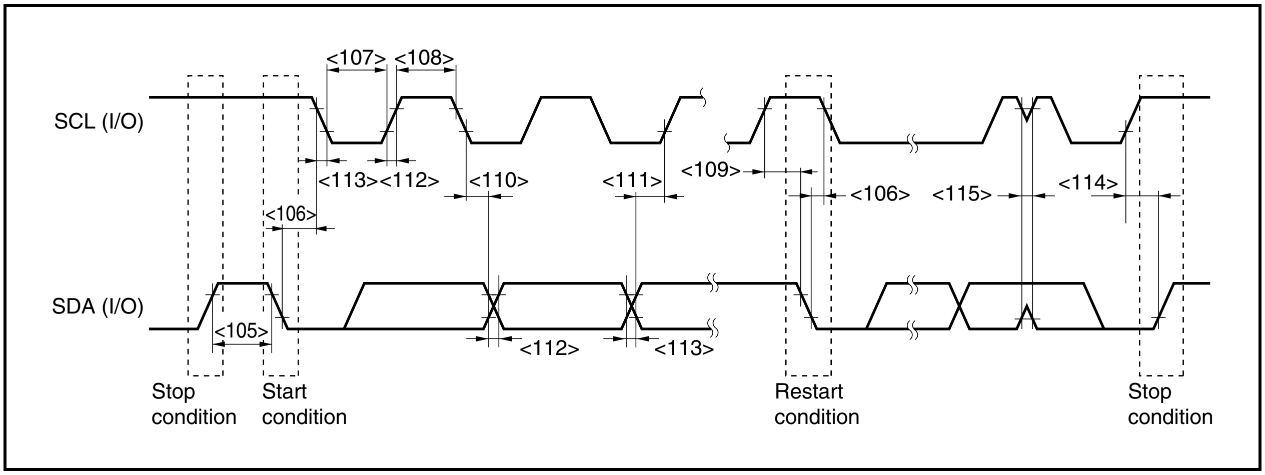
4. The high-speed-mode I^{2}C bus can be used in a normal-mode I^{2}C bus system. In this case, set the high-speed-mode I^{2}C bus so that it meets the following conditions.

- If the system does not extend the SCL signal's low state hold time:
  $t_{SU:DAT} \geq 250$ ns
- If the system extends the SCL signal's low state hold time:
  Transmit the next data bit to the SDA line prior to releasing the SCL line ($t_{Rmax.}$ + $t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Normal mode I^{2}C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

Remark The maximum operating frequency of the µPD703200Y, 703201Y, 703204Y, 70F3201Y, and 70F3204Y is f_{XX} = 17 MHz.

I^{2}C bus mode (μPD703200Y, 703201Y, 703204Y, 70F3201Y, 70F3204Y only)



A/D converter

(T_A = −40 to +85°C, V_{DD} = AV_{DD} = AV_{REF0} = 2.2 to 2.7 V, AV_{SS} = V_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 1]				±0.1	±0.3	%FSR
Conversion time	t_{CONV}		6.3		150	μs
Zero-scale error[Note 1]					±0.2	%FSR
Full-scale error[Note 1]					±0.2	%FSR
Integral linearity error[Note 2]					±2	LSB
Differential linearity error[Note 2]					±1	LSB
Analog reference voltage	AV_{REF}	$AV_{REF0} = AV_{DD}$	2.2		2.7	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF}	V
AV_{REF0} current	AI_{REF0}			10		μA
AV_{DD} power supply current	AI_{DD}			400	800	μA

Notes 1. Excluding quantization error (±0.05%FSR)

2. Excluding quantization error (±0.5 LSB)

Remark LSB: Least significant bit

FSR: Full-scale range

D/A converter

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = AV_{REF1} = 2.2 to 2.7 V, AV_{SS} = V_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Overall error[Note]		Load conditions: 2 MΩ, 30 pF AV_{REF1} = V_{DD}			±0.8	%FSR
Settling time					20	μs
Output resistance				10		kΩ
Analog reference voltage	AV_{REF}	AV_{REF1} = V_{DD}	2.2		2.7	V
AV_{REF1} current	AV_{REF1}	Per channel, during D/A operation		500		μA

Note Excluding quantization error (±0.2%FSR).

Flash memory programming characteristics (μPD70F3201, 70F3201Y, 70F3204, and 70F3204Y only)

(1) Basic characteristics

(V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{XX}	2.2 to 2.7 V	2		20	MHz
Supply voltage	V_{DD}		2.2		2.7	V
Number of rewrites	C_{WRT}	**Note**		100		Times
Supply current	I_{DD}	When writing			6	mA
		When erasing			6	mA
Programming temperature	T_{PRG}	2.2 to 2.7 V	–20		+85	°C

Note When writing data for the first time to a shipped product, a product that can be "erased then written" and a product that can be "only written" can be both be rewritten once.

Example: P write, E: erase
Shipped product → P → E → P → E → P: Number of rewrites: 3
Shipped product → E →P → E → P → E → P: Number of rewrites: 3

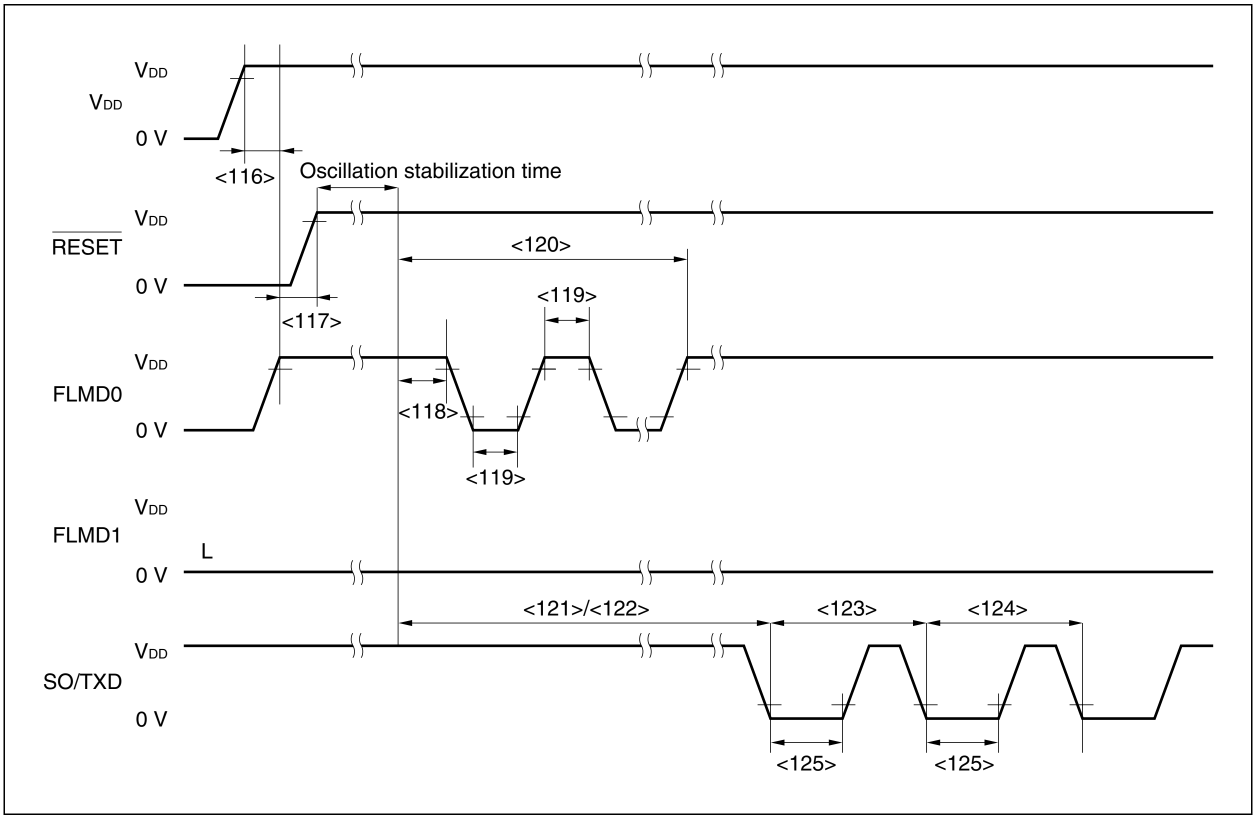
(2) Serial write operation characteristics

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = EV_{DD} = 2.2 to 2.7 V, V_{SS} = AV_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

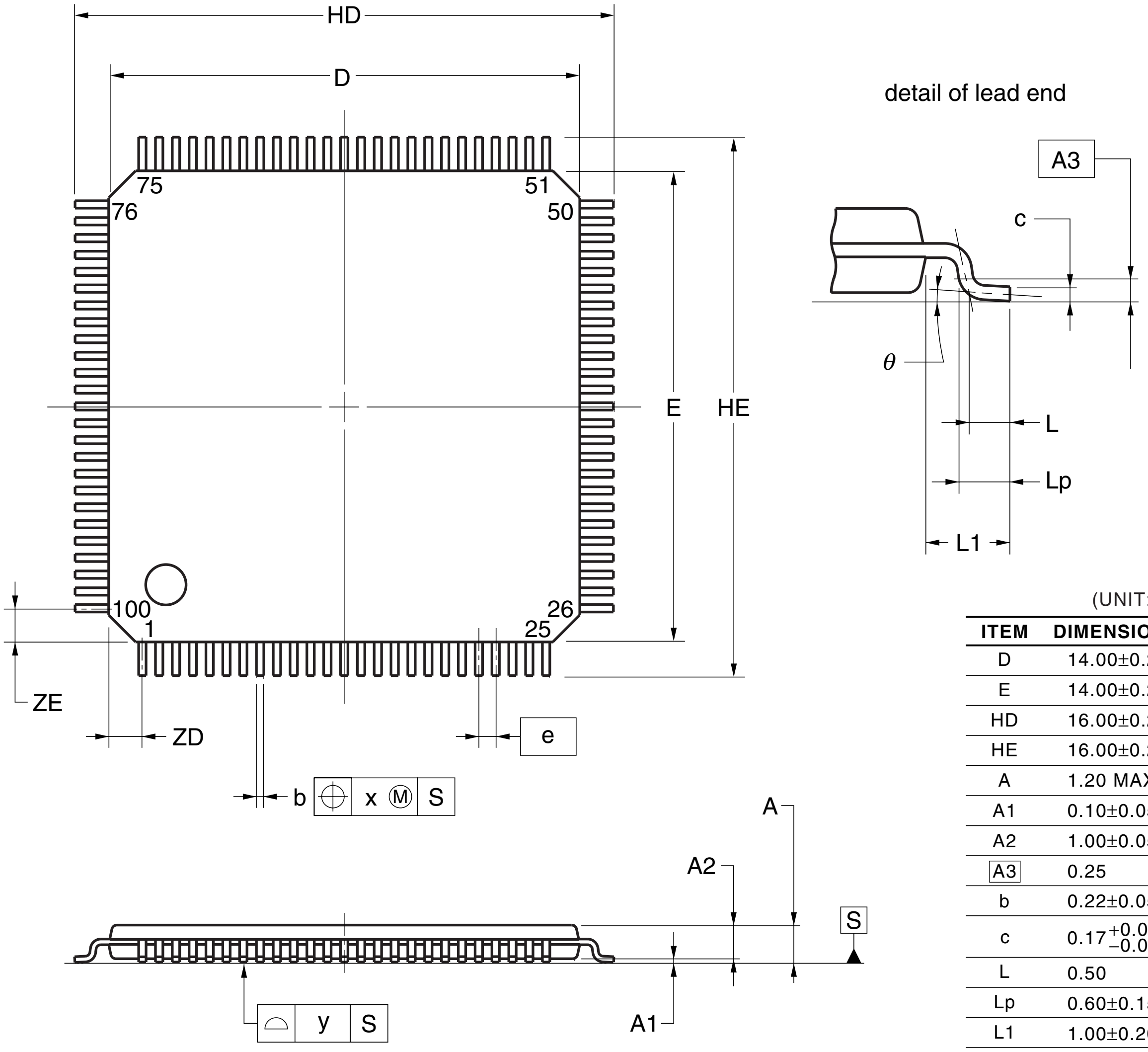
Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
Set time from VDD↑ to FLMD0↑	t_{DP}	<116>		10			μs
_RESET release time from FLMD0↑	t_{PR}	<117>		2			μs
Time from oscillation stabilization time end to FLMD0 pulse output start	t_{RP}	<118>	f_{XX} = 20 MHz	230			μs
FLMD0 pulse high-/low-level width	t_{PW}	<119>		10		100	μs
Time from oscillation stabilization time end to FLMD0 pulse input signal end	t_{RPE}	<120>	f_{XX} = 20 MHz			100	ms
Time from oscillation stabilization time end to reset command input[Note 1]	t_{RC}	<121>	f_{XX} = 20 MHz	t_{RPE} + 50			ms
Time from oscillation stabilization time end to low data input 1[Note 2]	t_{RI}	<122>	f_{XX} = 20 MHz	t_{RPE} + 50			ms
Time from low data input 1 to low data input 2[Note 2]	t_{I2}	<123>	f_{XX} = 20 MHz	1			ms
Time from low data input 2 to reset command input 2[Note 2]	t_{2C}	<124>	f_{XX} = 20 MHz	1			ms
Low data input width[Note 2]	t_{L1}, t_{L2}	<125>			"00H" @9600 bps		

Notes 1. When CSI0 or CSI0 + HS is selected.

2. When UART0 is selected.


VDD
VDD
0 V
<116>
Oscillation stabilization time
VDD
RESET
0 V
<117>
<120>
<119>
VDD
FLMD0
0 V
<118>
<119>
VDD
FLMD1
L
0 V
<121>/<122>
<123>
<124>
VDD
SO/TXD
0 V
<125>
<125>

⋆ 100-PIN PLASTIC TQFP (FINE PITCH) (14x14)



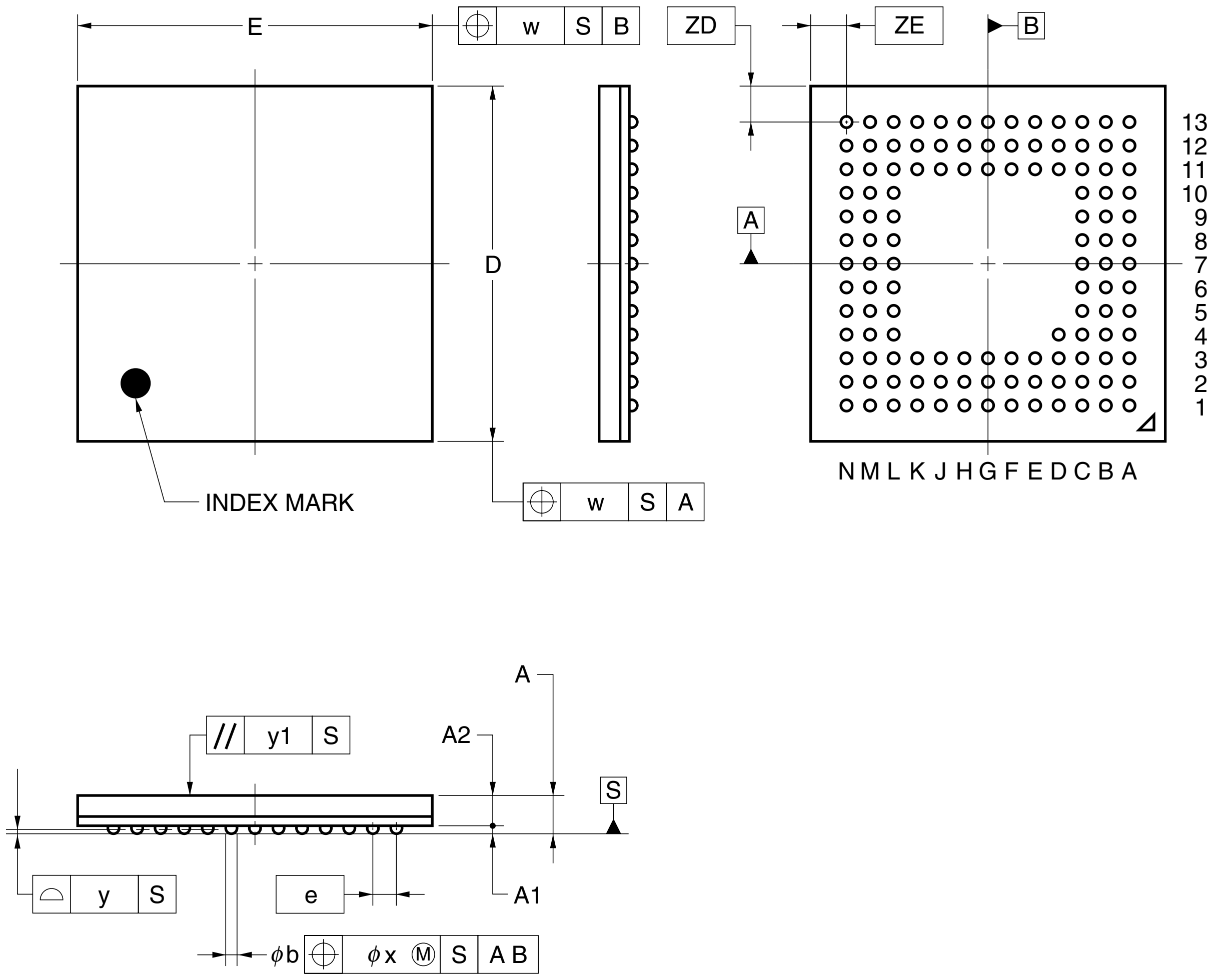
(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.20 MAX.
A1	0.10±0.05
A2	1.00±0.05
$\boxed{A3}$	0.25
b	0.22±0.05
c	$0.17^{+0.03}_{-0.07}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
$\boxed{e}$	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-YEU

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

121-PIN PLASTIC FBGA (12x12)



ITEM	MILLIMETERS
D	12.00±0.10
E	12.00±0.10
w	0.20
A	1.48±0.10
A1	0.35±0.06
A2	1.13
e	0.80
b	$0.50^{+0.05}_{-0.10}$
x	0.08
y	0.10
y1	0.20
ZD	1.20
ZE	1.20

P121F1-80-EA6

★

CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS

The V850ES/SA2 and V850ES/SA3 should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, consult an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 24-1. Surface Mounting Type Soldering Conditions(1/2)

(1) μPD703201GC-xxx-YEU-A 100-pin plastic TQFP (fine pitch) (14 × 14)
μPD70F3201GC-YEU-A 100-pin plastic TQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 36 to 72 hours)	IR60-363-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remark Products with -A at the end of the part number are lead-free products.

(2) μPD703201YGC-xxx-YEU-A 100-pin plastic TQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remark Products with -A at the end of the part number are lead-free products.

Table 24-1. Surface Mounting Type Soldering Conditions(2/2)

(3)	μPD703204F1-xxx-EA6-A	121-pin plastic FBGA (12 × 12)
	μPD703204YF1-xxx-EA6-A	121-pin plastic FBGA (12 × 12)
	μPD70F3204F1-EA6-A	121-pin plastic FBGA (12 × 12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-203-3

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remark Products with -A at the end of the part number are lead-free products.

(4)	μPD703200GC-xxx-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)
	μPD703200YGC-xxx-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)
	μPD70F3201YGC-YEU-A	100-pin plastic TQFP (fine pitch) (14 × 14)
	μPD70F3204YF1-EA6-A	121-pin plastic FBGA (12 × 12)

Undefined

★

APPENDIX A REGISTER INDEX

(1/7)

Symbol	Name	Unit	Page
ADCR	A/D conversion result register	ADC	286
ADCRH	A/D conversion result register H	ADC	286
ADIC	Interrupt control register	INTC	441
ADM	A/D converter mode register	ADC	282
ADS	Analog input channel specification register	ADC	284
ASIF0	UART0 transmit status register	UART	307
ASIF1	UART1 transmit status register	UART	307
ASIM0	UART0 operation mode register	UART	303
ASIM1	UART1 operation mode register	UART	303
ASIS0	UART0 reception error status register	UART	306
ASIS1	UART1 reception error status register	UART	306
AWC	Address wait control register	BCU	191
BCC	Bus cycle control register	BCU	192
BRGC0	Baud rate generator compare register 0	BRG	325
BRGC1	Baud rate generator compare register 1	BRG	325
BRGIC	Interrupt control register	INTC	441
BSC	Bus size configuration register	BCU	181
CC00	Capture/compare register 00	Timer	215
CC01	Capture/compare register 01	Timer	215
CC10	Capture/compare register 10	Timer	215
CC11	Capture/compare register 11	Timer	215
CCIC00	Interrupt control register	INTC	441
CCIC01	Interrupt control register	INTC	441
CCIC10	Interrupt control register	INTC	441
CCIC11	Interrupt control register	INTC	441
CKSR0	Clock select register 0	BRG	324
CKSR1	Clock select register 1	BRG	324
CORAD0	Correction address register 0	ROMC	477
CORAD0H	Correction address register 0H	ROMC	477
CORAD0L	Correction address register 0L	ROMC	477
CORAD1	Correction address register 1	ROMC	477
CORAD1H	Correction address register 1H	ROMC	477
CORAD1L	Correction address register 1L	ROMC	477
CORAD2	Correction address register 2	ROMC	477
CORAD2H	Correction address register 2H	ROMC	477
CORAD2L	Correction address register 2L	ROMC	477
CORAD3	Correction address register 3	ROMC	477
CORAD3H	Correction address register 3H	ROMC	477
CORAD3L	Correction address register 3L	ROMC	477

(2/7)

Symbol	Name	Unit	Page
CORCN	Correction control register	ROMC	478
CR2	Compare register 2	Timer	241
CR3	Compare register 3	Timer	241
CR4	Compare register 4	Timer	241
CR5	Compare register 5	Timer	241
CSIC0	Clocked serial interface clock selection register 0	CSI	338
CSIC1	Clocked serial interface clock selection register 1	CSI	338
CSIC2	Clocked serial interface clock selection register 2	CSI	338
CSIC3	Clocked serial interface clock selection register 3	CSI	338
CSIC4	Clocked serial interface clock selection register 4	CSI	338
CSIIC0	Interrupt control register	INTC	441
CSIIC1	Interrupt control register	INTC	441
CSIIC2	Interrupt control register	INTC	441
CSIIC3	Interrupt control register	INTC	441
CSIIC4	Interrupt control register	INTC	441
CSIM0	Clocked serial interface mode register 0	CSI	336
CSIM1	Clocked serial interface mode register 1	CSI	336
CSIM2	Clocked serial interface mode register 2	CSI	336
CSIM3	Clocked serial interface mode register 3	CSI	336
CSIM4	Clocked serial interface mode register 4	CSI	336
CTBP	CALLT base pointer	CPU	61
CTPC	CALLT execution status saving register	CPU	60
CTPSW	CALLT execution status saving register	CPU	60
DACS0	D/A converter conversion value setting register 0	DAC	296
DACS1	D/A converter conversion value setting register 1	DAC	296
DADC0	DMA addressing control register 0	DMAC	413
DADC1	DMA addressing control register 1	DMAC	413
DADC2	DMA addressing control register 2	DMAC	413
DADC3	DMA addressing control register 3	DMAC	413
DAM	D/A converter mode register	DAC	295
DAY	Day count register	RTC	264
DAYB	Day count setting register	RTC	264
DBC0	DMA transfer count register 0	DMAC	412
DBC1	DMA transfer count register 1	DMAC	412
DBC2	DMA transfer count register 2	DMAC	412
DBC3	DMA transfer count register 3	DMAC	412
DBPC	Exception/debug trap status saving register	CPU	61
DBPSW	Exception/debug trap status saving register	CPU	61
DCHC0	DMA channel control register 0	DMAC	414
DCHC1	DMA channel control register 1	DMAC	414
DCHC2	DMA channel control register 2	DMAC	414
DCHC3	DMA channel control register 3	DMAC	414
DDA0H	DMA destination address register 0H	DMAC	411
DDA0L	DMA destination address register 0L	DMAC	411

(3/7)

Symbol	Name	Unit	Page
DDA1H	DMA destination address register 1H	DMAC	411
DDA1L	DMA destination address register 1L	DMAC	411
DDA2H	DMA destination address register 2H	DMAC	411
DDA2L	DMA destination address register 2L	DMAC	411
DDA3H	DMA destination address register 3H	DMAC	411
DDA3L	DMA destination address register 3L	DMAC	411
DMAIC0	Interrupt control register	INTC	441
DMAIC1	Interrupt control register	INTC	441
DMAIC2	Interrupt control register	INTC	441
DMAIC3	Interrupt control register	INTC	441
DSA0H	DMA source address register 0H	DMAC	410
DSA0L	DMA source address register 0L	DMAC	410
DSA1H	DMA source address register 1H	DMAC	410
DSA1L	DMA source address register 1L	DMAC	410
DSA2H	DMA source address register 2H	DMAC	410
DSA2L	DMA source address register 2L	DMAC	410
DSA3H	DMA source address register 3H	DMAC	410
DSA3L	DMA source address register 3L	DMAC	410
DTFR0	DMA trigger factor register 0	DMAC	415
DTFR1	DMA trigger factor register 1	DMAC	415
DTFR2	DMA trigger factor register 2	DMAC	415
DTFR3	DMA trigger factor register 3	DMAC	415
DWC0	Data wait control register 0	BCU	188
ECR	Interrupt source register	CPU	58
EIPC	Interrupt status saving register	CPU	57
EIPSW	Interrupt status saving register	CPU	57
EXIMC	External bus interface mode control register	BCU	180
FEPC	NMI status saving register	CPU	58
FEPSW	NMI status saving register	CPU	58
HOUR	Hour count register	RTC	263
HOURB	Hour count setting register	RTC	264
IIC	IIC shift register	I^{2}C	363
IICC	IIC control register	I^{2}C	353
IICCL	IIC clock select register	I^{2}C	361
IICIC	Interrupt control register	INTC	441
IICS	IIC status register	I^{2}C	358
IICX	IIC function expansion register	I^{2}C	362
IMR0	Interrupt mask register 0	INTC	442
IMR0H	Interrupt mask register 0H	INTC	442
IMR0L	Interrupt mask register 0L	INTC	442
IMR1	Interrupt mask register 1	INTC	442
IMR1H	Interrupt mask register 1H	INTC	442
IMR1L	Interrupt mask register 1L	INTC	442
IMR2	Interrupt mask register 2	INTC	442

(4/7)

Symbol	Name	Unit	Page
IMR2L	Interrupt mask register 2L	INTC	442
INTF0	External interrupt falling edge specification register 0	INTC	95, 446
INTF9	External interrupt falling edge specification register 9	INTC	135, 447
INTF9L	External interrupt falling edge specification register 9L	INTC	135, 447
INTR0	External interrupt rising edge specification register 0	INTC	96, 446
INTR9	External interrupt rising edge specification register 9	INTC	136, 447
INTR9L	External interrupt rising edge specification register 9L	INTC	136, 447
ISPR	In-service priority register	INTC	443
MIN	Minute count register	RTC	263
MINB	Minute count setting register	RTC	263
OSTS	Oscillation stabilization time selection register	WDT	271, 276
OVFIC0	Interrupt control register	INTC	441
OVFIC1	Interrupt control register	INTC	441
P0	Port register 0	Port	93
P2	Port register 2	Port	100
P3	Port register 3	Port	107
P4	Port register 4	Port	114
P7	Port register 7	Port	124
P7H	Port register 7H	Port	124
P7L	Port register 7L	Port	124
P8	Port register 8	Port	125
P9	Port register 9	Port	128
P9H	Port register 9H	Port	128
P9L	Port register 9L	Port	128
PC	Program counter	CPU	55
PCC	Processor clock control register	CG	205
PCD	Port register CD	Port	144
PCM	Port register CM	Port	147
PCS	Port register CS	Port	153
PCT	Port register CT	Port	158
PDH	Port register DH	Port	163
PDL	Port register DL	Port	167
PDLH	Port register DLH	Port	167
PDLL	Port register DLL	Port	167
PF2	Port function register 2	Port	101
PF3	Port function register 3	Port	109
PF4	Port function register 4	Port	117
PF9	Port function register 9	Port	134
PF9H	Port function register 9H	Port	134
PFC3	Port function control register 3	Port	108
PFC4	Port function control register 4	Port	116
PFC9	Port function control register 9	Port	132
PFC9H	Port function control register 9H	Port	132
PFC9L	Port function control register 9L	Port	132

(5/7)

Symbol	Name	Unit	Page
PFM	Power fail comparison mode register	ADC	285
PFT	Power fail comparison threshold value register	ADC	285
PIC0	Interrupt control register	INTC	441
PIC1	Interrupt control register	INTC	441
PIC2	Interrupt control register	INTC	441
PIC3	Interrupt control register	INTC	441
PIC4	Interrupt control register	INTC	441
PIC5	Interrupt control register	INTC	441
PIC6	Interrupt control register	INTC	441
PM0	Port mode register 0	Port	93
PM2	Port mode register 2	Port	100
PM3	Port mode register 3	Port	107
PM4	Port mode register 4	Port	114
PM9	Port mode register 9	Port	129
PM9H	Port mode register 9H	Port	129
PM9L	Port mode register 9L	Port	129
PMC0	Port mode control register 0	Port	94
PMC2	Port mode control register 2	Port	101
PMC3	Port mode control register 3	Port	108
PMC4	Port mode control register 4	Port	115
PMC9	Port mode control register 9	Port	130
PMC9H	Port mode control register 9H	Port	130
PMC9L	Port mode control register 9L	Port	130
PMCCM	Port mode control register CM	Port	148
PMCCS	Port mode control register CS	Port	154
PMCCT	Port mode control register CT	Port	159
PMCD	Port mode register CD	Port	144
PMCDH	Port mode control register DH	Port	164
PMCDL	Port mode control register DL	Port	168
PMCDLH	Port mode control register DLH	Port	168
PMCDLL	Port mode control register DLL	Port	168
PMCM	Port mode register CM	Port	147
PMCS	Port mode register CS	Port	153
PMCT	Port mode register CT	Port	158
PMDH	Port mode register DH	Port	163
PMDL	Port mode register DL	Port	168
PMDLH	Port mode register DLH	Port	168
PMDLL	Port mode register DLL	Port	168
PRCMD	Command register	CPU	85
PRSCM	Prescaler compare register	CG	211
PRSM	Prescaler mode register	CG	210
PSC	Power save control register	CG	207, 471
PSMR	Power save mode register	CG	207, 471
PSW	Program status word	CPU	59

(6/7)

Symbol	Name	Unit	Page
PU0	Pull-up resistor option register 0	Port	95
PU2	Pull-up resistor option register 2	Port	102
PU3	Pull-up resistor option register 3	Port	109
PU4	Pull-up resistor option register 4	Port	117
PU9	Pull-up resistor option register 9	Port	135
PU9H	Pull-up resistor option register 9H	Port	135
PU9L	Pull-up resistor option register 9L	Port	135
ROVIC	Interrupt control register	INTC	441
RTCC0	RTC operation control register 0	RTC	260
RTCC1	RTC operation control register 1	RTC	261
RTCIC	Interrupt control register	INTC	441
RXB0	Receive buffer register 0	UART	308
RXB1	Receive buffer register 1	UART	308
SEC	Second count register	RTC	262
SECB	Second count setting register	RTC	262
SES0	Valid edge select register 0	Timer	221
SES1	Valid edge select register 1	Timer	221
SIO0	Serial I/O shift register 0	CSI	340
SIO1	Serial I/O shift register 1	CSI	340
SIO2	Serial I/O shift register 2	CSI	340
SIO3	Serial I/O shift register 3	CSI	340
SIO4	Serial I/O shift register 4	CSI	340
SIOE0	Receive-only serial I/O shift register 0	CSI	340
SIOE1	Receive-only serial I/O shift register 1	CSI	340
SIOE2	Receive-only serial I/O shift register 2	CSI	340
SIOE3	Receive-only serial I/O shift register 3	CSI	340
SIOE4	Receive-only serial I/O shift register 4	CSI	340
SOTB0	Clocked serial interface transmit buffer register 0	CSI	341
SOTB1	Clocked serial interface transmit buffer register 1	CSI	341
SOTB2	Clocked serial interface transmit buffer register 2	CSI	341
SOTB3	Clocked serial interface transmit buffer register 3	CSI	341
SOTB4	Clocked serial interface transmit buffer register 4	CSI	341
SREIC0	Interrupt control register	INTC	441
SREIC1	Interrupt control register	INTC	441
SRIC0	Interrupt control register	INTC	441
SRIC1	Interrupt control register	INTC	441
STIC0	Interrupt control register	INTC	441
STIC1	Interrupt control register	INTC	441
SUBC	Sub-count register	RTC	262
SVA	Slave address register	I^2C	363
SYS	System status register	CPU	86
TCL2	Timer clock selection register 2	Timer	242
TCL3	Timer clock selection register 3	Timer	242
TCL4	Timer clock selection register 4	Timer	243

(7/7)

Symbol	Name	Unit	Page
TCL5	Timer clock selection register 5	Timer	243
TM0	Timer 0	Timer	214
TM1	Timer 1	Timer	214
TM2	Timer counter 2	Timer	241
TM3	Timer counter 3	Timer	241
TM4	Timer counter 4	Timer	241
TM5	Timer counter 5	Timer	241
TMC00	Timer control register 00	Timer	217
TMC01	Timer control register 01	Timer	219
TMC10	Timer control register 10	Timer	217
TMC11	Timer control register 11	Timer	217
TMC2	Timer mode control register 2	Timer	244
TMC3	Timer mode control register 3	Timer	244
TMC4	Timer mode control register 4	Timer	244
TMC5	Timer mode control register 5	Timer	244
TMIC2	Interrupt control register	INTC	441
TMIC3	Interrupt control register	INTC	441
TMIC4	Interrupt control register	INTC	441
TMIC5	Interrupt control register	INTC	441
TXB0	Transmit buffer register 0	UART	309
TXB1	Transmit buffer register 1	UART	309
VSWC	System wait control register	BCU	87
WDCS	Watchdog timer clock selection register	WDT	272
WDTIC	Interrupt control register	INTC	441
WDTM	Watchdog timer mode register	WDT	273, 444
WEEK	Week count register	RTC	265
WEEKB	Week count setting register	RTC	265

★

APPENDIX B INSTRUCTION SET LIST

B.1 Conventions

(1) Symbols used to describe operands

Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the condition codes
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Symbols used to describe opcodes

Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list

(3) Symbols used in operations

Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Half-word	Halfword (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
||	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Symbols used in execution clock

Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Symbols used in flag operations

Symbol	Explanation
(Blank)	No change
0	Clear to 0
×	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z	0 0 1 0	Z = 1	Zero
NZ	1 0 1 0	Z = 0	Not zero
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
S/N	0 1 0 0	S = 1	Negative
NS/P	1 1 0 0	S = 0	Positive
T	0 1 0 1	—	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (In Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Half-word))		4	4	4					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extend(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (returned PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR Note 5	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110dddddd ddddddddddddddd0 Note 7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110dddddd ddddddddddddddd0 Note 7	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 Notes 8, 10	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Half-word))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Half-word)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii IIIIIIIIIIIIIIII	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2][Note 6]xGR[reg1][Note 6]		1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2][Note 6]xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1][Note 6]ximm16		1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm$^{\text{Note 15}}$	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp–zero-extend (imm5) ep←sp/imm	n+2 Note 4 Note17	n+2 Note 4 Note17	n+2 Note 4 Note17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16))	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	if conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Half-word))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Half-word)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Half-word)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR[reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Half-word))) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend(GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend(GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	00000111111iiiii 0000000100000000	EIPC ←PC+4 (Return PC) EIPSW ←PSW ECR.EICC ←Interrupt Code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. dddddddd: Higher 8 bits of disp9.

2. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

★

APPENDIX C REVISION HISTORY

C.1 Major Revisions in This Edition

Page	Description
Throughout	Addition of products (μPD703200 and 703200Y)
Throughout	Modification of minimum instruction execution time
Throughout	Deletion of backup mode
p.20	Change of pin names from V_{DDBU} and V_{SSBU} to V_{DD} and V_{SS} respectively in **1.5 Pin Configuration**
p.84	Modification of **3.4.8 (1) Setting data to special registers**
p.87	Addition of **3.4.9 (2) Access to special internal peripheral I/O registers**
p.165	Modification of **Figure 4-32 Block Diagram of PDH0 to PDH7**
p.178	Modification of **5.2.1 Pin status when internal ROM, internal RAM, or internal peripheral I/O is accessed**
p.196	Modification of **5.11 Bus Timing**
p.268	Modification of **9.3.4 Notes**
p.283	Modification of **Tables 11-2** and **11-3 Example of Setting A/D Conversion Time**
p.289	Addition of **11.5 Notes on Use**
p.290	Addition of **11.6 How to Read A/D Converter Characteristics Table**
p.331	Modification of **13.7 Cautions**
p.422	Modification of **16.11 Precautions**
p.440	Addition of **Caution** to **17.3.4 Interrupt control register (xxICn)**
p.443	Addition of **Caution** to **17.3.6 In-service priority register (ISPR)**
p.453	Addition of **17.6.2 Debug trap**
p.482	Addition of **21.2 Functional Overview**
p.484	Addition of **Figure 21-1 Wiring Example of V850ES/SA2 Flash Write Adapter (FA-100GC-8EU-A)**
p.485	Addition of **Table 21-4 Wiring of V850ES/SA2 Flash Write Adapter (FA-100GC-8EU-A)**
p.486	Addition of **Figure 21-2 Wiring Example of V850ES/SA3 Flash Write Adapter (FA-121F1-EA6-A)**
p.487	Addition of **Table 21-5 Wiring of V850ES/SA3 Flash Write Adapter (FA-121F1-EA6-A)**
p.491	Addition of **21.5.2 FLMD1 pin**
p.499	Addition of **21.7 Rewriting by Self Programming**
p.503	Modification of **CHAPTER 22 ELECTRICAL SPECIFICATIONS**
p.532	Addition of 100-pin plastic TQFP (fine pitch) to **CHAPTER 23 PACKAGE DRAWINGS**
p.534	Addition of **CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS**
p.536	Addition of **APPENDIX A REGISTER INDEX**
p.543	Addition of **APPENDIX B INSTRUCTION SET LIST**
p.553	Addition of **APPENDIX C REVISION HISTORY**
Major revisions in modification version (U15905EJ2V1UD00)	
p.19	Modification of **1.4 Ordering information**
p.534	Modification of **CHAPTER 24 RECOMMENDED SOLDERING CONDITIONS**