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

NEC

RENESAS

V850/SA1

32-Bit Single-chip Microcontroller

Hardware

μPD703014A
μPD703014AY
μPD703014B
μPD703014BY
μPD703015A
μPD703015AY
μPD703015B
μPD703015BY
μPD703017A
μPD703017AY
μPD70F3015B
μPD70F3015BY
μPD70F3017A
μPD70F3017AY

Document No. U12768EJ4V1UD00 (4th edition)
Date Published August 2005 N CP(K)

© NEC Electronics Corporation 1997, 2000, 2002
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.

Windows is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

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 August, 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

Major Revisions in This Edition (1/2)

Page	Description
Throughout	Addition of μPD703014B, 703014BY, 703015B, 703015BY, 70F3015B, and 70F3015BY Deletion of μPD703014AGC, 703014AYGC, 703015AGC, and 703015AYGC
p. 27	Addition of **Table 1-1 List of V850/SA1 Products**
p. 28	Addition of description to minimum instruction execution time in **1.2 Features**
p. 30	Deletion and addition of products in **1.4 Ordering Information**
p. 31	Deletion and addition of products in **1.5 Pin Configuration**
p. 35	Deletion of description in **1.6.2 (2) Bus control unit (BCU)**
p. 38	Addition of **Table 2-1 Pin I/O Buffer Power Supplies**
p. 43	Modification of description in **Table 2-2 Operating States of Pins in Each Operating Mode**
p. 49	Modification of description in **2.3 (7) P60 to P65 (Port 6)**
p. 53	Addition of **2.3 (13) CLKOUT (Clock out)**
p. 55	Addition and modification of description in **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**
p. 58	Modification of **2.5 Pin I/O Circuits**
p. 59	Addition of description to minimum instruction execution time in **3.1 Features**
p. 63	Change of description in **3.2.2 (2) Program status word (PSW)**
p. 80	Modification of **Figure 3-16 Recommended Memory Map**
p. 81	Addition of description in **3.4.8 Peripheral I/O registers**
p. 86	Addition and modification of description in **3.4.9 Specific registers**
p. 113	Addition of description in **5.2.4 Noise elimination of external interrupt request input pin**
p. 114	Addition of description in **5.2.5 Edge detection function of external interrupt request input pin**
p. 122	Addition to **Cautions** in **5.3.4 Interrupt control register (xxICn)**
p. 125	Addition of **Caution** in **5.3.5 In-service priority register (ISPR)**
p. 136	Addition of **5.8.1 Interrupt request valid timing after EI instruction**
p. 137	Addition of **5.9 Bit Manipulation Instruction of Interrupt Control Register During DMA Transfer**
p. 138	Modification of description in **6.1 (1) Main clock oscillator**
p. 138	Modification of description in **6.1 (2) Subclock oscillator**
p. 139	Modification of **Figure 6-1 Clock Generator**
p. 140	Addition to **Notes** in **6.3.1 (1) Processor clock control register (PCC)**
p. 141	Modification of description in **6.3.1 (1) (b) Example of subclock operation → main clock operation setup**
p. 142	Addition to **Notes** and **Cautions** in **6.3.1 (2) Power save control register (PSC)**
p. 148	Modification of description in **6.4.4 (1) Settings and operating states**
p. 151	Addition of **6.6 Notes on Power Save Function**
p. 156	Modification of **Caution** in **7.1.3 (2) Capture/compare registers 00, 10 (CR00, CR10)**
p. 157	Modification of **Caution** in **7.1.3 (3) Capture/compare registers 01, 11 (CR01, CR11)**
p. 185	Change of **Figure 7-27 Data Hold Timing of Capture Register**
p. 185	Addition of **7.2.7 (6) (c) One-shot output function**
p. 189	Addition of **7.3.1 Outline**

The mark ★ shows major revised points.

Major Revisions in This Edition (2/2)

Page	Description
p. 196	Change of **Caution** in **7.3.4 (2) 8-bit timer mode control registers 2 to 5 (TMC2 to TMC5)**
p. 242	Modification of description in **10.3.2 (3) IIC clock select register 0 (IICCL0), IIC function expansion register 0 (IICX0)**
pp. 295 to 299	Addition of **Figures 10-25** to **10-29**
p. 313	Modification in **11.3 (1) A/D converter mode register (ADM)**
p. 314	Addition of **Table 11-2 A/D Conversion Time Selection**
p. 326	Addition of **11.6 How to Read A/D Converter Characteristics Table**
p. 330	Change of description in **12.1 Functions**
p. 330	Deletion of **12.2 Transfer Completion Interrupt Request** and addition of **12.2 Features**
p. 331	Addition of **12.3 Configuration**
p. 333	Addition of **Figure 12-2 Correspondence Between DRAn Setting Value and Internal RAM (4 KB)**
p. 334	Addition of **Figure 12-3 Correspondence Between DRAn Setting Value and Internal RAM (8 KB)**
p. 338	Addition of **12.5 Operations**
p. 339	Addition of **12.6 Cautions**
p. 340	Addition of **13.2 Features**
p. 342	Addition of **13.3 (2) Output latch**
p. 345	Modification of description in **13.5 Usage**
p. 347	Addition of description in **13.7 Cautions**
p. 348	Addition of **Table 14-1 Pin I/O Buffer Power Supplies**
p. 380	Addition of **Caution** in **14.2.8 (1) Function of P9 pins**
p. 396	Addition of **14.4 Operation of Port Function**
p. 398	Addition of **Caution** in **CHAPTER 16 FLASH MEMORY**
p. 398	Change of description in **16.1.1 Erasing unit**
p. 400	Addition of **Figure 16-1 Wiring Example of V850/SA1 Flash Writing Adapter (FA-100GC-8EU)**
p. 401	Addition of **Table 16-1 Wiring Table of V850/SA1 Flash Writing Adapter (FA-100GC-8EU)**
p. 402	Addition of **Figure 16-2 Wiring Example of V850/SA1 Flash Writing Adapter (FA-121F1-EA6)**
p. 403	Addition of **Table 16-2 Wiring Table of V850/SA1 Flash Writing Adapter (FA-121F1-EA6)**
p. 413	Addition of **16.7 Flash Memory Programming by Self-Programming**
p. 434	Addition of **CHAPTER 17 ELECTRICAL SPECIFICATIONS**
p. 460	Addition of **CHAPTER 18 PACKAGE DRAWINGS**
p. 462	Addition of **CHAPTER 19 RECOMMENDED SOLDERING CONDITIONS**
p. 466	Addition of **APPENDIX A NOTES ON TARGET SYSTEM DESIGN**
p. 468	Addition of description in **APPENDIX B REGISTER INDEX**
p. 486	Addition of **APPENDIX E REVISION HISTORY**
Major revisions in modification version (U12768EJ4V1UD00)	
Throughout	Addition of lead-free products PD703014AF1- -EA6-A, 703014AYF1- -EA6-A, 703014BBF1- -EA6-A, 703014BGC- -8EU-A, PD703014BYF1- - EA6-A, 703014BYGC- -8EU-A, 703015AF1- -EA6-A, 703015AYF1- -EA6-A, PD703015BF1- -EA6-A, 703015BGC- -8EU-A, 703015BYF1- -EA6-A, 703015BYGC- -8EU-A, PD703017AF1- -EA6-A, 703017AGC- -8EU-A, 703017AYF1- -EA6-A, 703017AYGC- -8EU-A, PD70F3015BF1-EA6-A, 70F3015BGC-8EU-A, 70F3015BYF1-EA6-A, 70F3015BYGC-8EU-A, PD70F3017AF1-EA6-A, 70F3017AGC-8EU-A, 70F3017AYF1-EA6-A, 70F3017AYGC-8EU-A
p.462	Addition of lead-free products to **CHAPTER 19 RECOMMENDED SOLDERING CONDITIONS**

The mark ⋆ shows major revised points.

INTRODUCTION

Readers This manual is intended for users who wish to understand the functions of the V850/SA1 (μPD703014A, 703014AY, 703014B, 703014BY 703015A, 703015AY, 703015B, 703015BY, 703017A, 703017AY, 70F3015B, 70F3015BY, 70F3017A, 70F3017AY) and design application systems using the V850/SA1.

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

Organization The V850/SA1 User's Manual is divided into two parts: hardware (this manual) and architecture (V850 Series User's Manual Architecture).

Hardware	Architecture
• Pin function	• Data type
• CPU function	• Register set
• On-chip peripheral function	• Instruction format and instruction set
• Flash memory programming	• Interrupt and exception
• Electrical specifications	• Pipeline operation

How to Use This Manual It is assumed that the reader of this manual has general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To find out the details of a register whose name is known:
→ Refer to **APPENDIX B REGISTER INDEX**.

To find out the details of a function, etc., whose name is known:
→ Refer to **APPENDIX D INDEX**.

To understand the details of a instruction function:
→ Refer to **V850 Series User's Manual Architecture** available separately.

How to read register formats:
→ Names of bits whose numbers are enclosed in a square are defined in the device file under reserved words.

To understand the overall functions of the V850/SA1:
→ Read this manual in the order of the **CONTENTS**.

To know the electrical specifications of the V850/SA1:
→ Refer to **CHAPTER 17 ELECTRICAL SPECIFICATIONS**.

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 items marked with **Note** in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numerical representation: Binary … xxxx or xxxxB
Decimal … xxxx
Hexadecimal … xxxxH

Prefixes indicating power of 2 (address space, memory capacity):

K (kilo) … $2^{10}=1024$

M (mega) … $2^{20}=1024^2$

G (giga) … $2^{30}=1024^3$

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

Documents related to V850/SA1

Document Name	Document No.
V850 Series Architecture User's Manual	U10243E
V850/SA1 Application Note	U13851E
V850/SA1 Hardware User's Manual	This manual
V850 Series Flash Memory Self Programming User's Manual	U15673E

Documents related to development tools (user's manuals)

Document Name		Document No.
IE-703002-MC (In-Circuit Emulator)		U11595E
IE-703017-MC-EM1 (In-Circuit Emulator Option Board)		U12898E
CA850 Ver. 2.40 or Later C Compiler Package	Operation	U15024E
	C Language	U15025E
	Project Manager	U15026E
	Assembly Language	U15027E
ID850 Ver. 2.40 Integrated Debugger	Operation (Windows™ Based)	U15181E
SM850 Ver. 2.40 System Simulator	Operation (Windows Based)	U15182E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specifications	U14873E
RX850 Ver. 3.13 or Later Real-time OS	Basic	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.13 or Later Real-time OS	Basic	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.0 System Performance Analyzer		U14410E
PG-FP3 Flash Memory Programmer		U13502E
PG-FP4 Flash Memory Programmer		U15260E

CONTENTS

LIST OF FIGURES (1/6)

LIST OF FIGURES (2/6)

LIST OF FIGURES (3/6)

LIST OF FIGURES (4/6)

LIST OF FIGURES (5/6)

LIST OF FIGURES (6/6)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

CHAPTER 1 INTRODUCTION

The V850/SA1 is a low-power series product in the NEC Electronics V850 Series of single-chip microcontrollers designed for real-time control.

1.1 General

The V850/SA1 is a 32-bit single-chip microcontroller that includes the V850 Series CPU core, and peripheral functions such as ROM/RAM, a timer/counter, a serial interface, an A/D converter, and a DMA controller.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850/SA1 has multiply, saturation operation, and bit manipulation instructions realized with a hardware multiplier for digital servo control. Moreover, as a real-time control system, the V850/SA1 enables the realization of extremely high cost-performance for applications that require low power consumption, such as camcorders and other AV equipment, and portable telephone equipment such as cellular phones and PHS phone systems.

★ **Table 1-1. List of V850/SA1 Products**

Product Name	I^2C Function	ROM		RAM Size	Package
		Type	Size		
μPD703014A	Not available	Mask ROM	64 KB	4 KB	121-pin FBGA (12 × 12)
μPD703014AY	Available				
μPD703014B	Not available				100-pin LQFP (14 × 14)
μPD703014BY	Available				
μPD703015A	Not available		128 KB		121-pin FBGA (12 × 12)
μPD703015AY	Available				
μPD703015B	Not available				100-pin LQFP (14 × 14)
μPD703015BY	Available				
μPD703017A	Not available		256 KB	8 KB	100-pin LQFP (14 × 14)/ 121-pin FBGA (12 × 12)
μPD703017AY	Available				
μPD70F3015B	Not available	Flash memory	128 KB	4 KB	100-pin LQFP (14 × 14)
μPD70F3015BY	Available				
μPD70F3017A	Not available		256 KB	8 KB	100-pin LQFP (14 × 14)/ 121-pin FBGA (12 × 12)
μPD70F3017AY	Available				

1.2 Features

○	Number of instructions	74
★ ○	Minimum instruction execution time	50 ns (@20 MHz operation with main clock (f_{XX})) 58.8 ns (@17 MHz operation with main clock (f_{XX})) 30.5 μs (@32.768 kHz operation with subclock (f_{XT}))
○	General-purpose registers	32 bits × 32 registers
○	Instruction set	Signed multiplication (16 × 16 → 32): 100 ns (@20 MHz operation) (able to execute instructions in parallel continuously without creating any register hazards) Saturation operations (overflow and underflow detection functions are included) 32-bit shift instruction: 1 clock Bit manipulation instructions Load/store instructions with long/short format
○	Memory space	16 MB of linear address space (for programs and data) External expandability: Expandable to 4 MB Memory block allocation function: 2 MB per block Programmable wait function Idle state insertion function
○	External bus interface	16-bit data bus (address/data multiplex) Address bus: Separate output enabled Bus hold function External wait function
★ ○	Internal memory	μPD703014A, 703014AY, 703014B, 703014BY (mask ROM: 64 KB/RAM: 4 KB) μPD703015A, 703015AY, 703015B, 703015BY (mask ROM: 128 KB/RAM: 4 KB) μPD703017A, 703017AY (mask ROM: 256 KB/RAM: 8 KB) μPD70F3015B, 70F3015BY (flash memory: 128 KB/RAM: 4 KB) μPD70F3017A, 70F3017AY (flash memory: 256 KB/RAM: 8 KB)
○	Interrupts and exceptions	External interrupts: 8 sources (5 sources[Note]) Internal interrupts: 24 sources Software exceptions: 32 sources Exception trap: 1 source Interrupt priorities can be changed (8 levels)
○	I/O lines	Total: 85 (13 input ports and 72 I/O ports)
○	Timer/counter	16-bit timer: 2 channels (PWM output) 8-bit timer: 4 channels (PWM outputs, cascade connection enabled)
○	Watch timer	When operating under subclock or main clock: 1 channel
○	Watchdog timer	1 channel

Note Number of external interrupts that can release software STOP mode.

○ Serial interface (SIO)	Asynchronous serial interface (UART) Clocked serial interface (CSI)
★	I^2C bus interface (I^2C) (μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only) UART: 1 channel CSI: 1 channel UART/CSI: 1 channel I^2C/CSI: 1 channel UART dedicated baud rate generator: 2 channels
○ A/D converter	10-bit resolution: 12 channels
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 3 channels
○ Real-time output port	8 bits × 1 channel or 4 bits × 2 channels
○ Clock generator	During main clock or subclock operation 5-level CPU clock (including clock-through and sub operations)
○ Power-saving functions	HALT/IDLE/software STOP modes
○ Package	100-pin plastic LQFP (fine pitch, 14 × 14) 121-pin plastic FBGA (12 × 12)
○ CMOS structure	Fully static circuits

1.3 Applications

General battery-driven equipment such as camcorders (including DVC), meters, etc.

★ 1.4 Ordering Information

Part Number	Package	Internal ROM
PD703014AF1- -EA6	121-pin plastic FBGA (12×12)	64 KB (mask ROM)
PD703014AF1- -EA6-A	121-pin plastic FBGA (12×12)	64 KB (mask ROM)
PD703014AYF1- -EA6	121-pin plastic FBGA (12×12)	64 KB (mask ROM)
PD703014AYF1- -EA6-A	121-pin plastic FBGA (12×12)	64 KB (mask ROM)
PD703014BF1- -EA6-A	121-pin plastic FBGA (12×12)	64 KB (mask ROM)
PD703014BGC- -8EU	100-pin plastic LQFP (fine pitch) (14×14)	64 KB (mask ROM)
PD703014BGC- -8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	64 KB (mask ROM)
PD703014BYF1- -EA6-A	121-pin plastic FBGA (12×12)	64 KB (mask ROM)
PD703014BYGC- -8EU	100-pin plastic LQFP (fine pitch) (14×14)	64 KB (mask ROM)
PD703014BYGC- -8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	64 KB (mask ROM)
PD703015AF1- -EA6	121-pin plastic FBGA (12×12)	128 KB (mask ROM)
PD703015AF1- -EA6-A	121-pin plastic FBGA (12×12)	128 KB (mask ROM)
PD703015AYF1- -EA6	121-pin plastic FBGA (12×12)	128 KB (mask ROM)
PD703015AYF1- -EA6-A	121-pin plastic FBGA (12×12)	128 KB (mask ROM)
PD703015BF1- -EA6-A	121-pin plastic FBGA (12×12)	128 KB (mask ROM)
PD703015BGC- -8EU	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (mask ROM)
PD703015BGC- -8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (mask ROM)
PD703015BYF1- -EA6-A	121-pin plastic FBGA (12×12)	128 KB (mask ROM)
PD703015BYGC- -8EU	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (mask ROM)
PD703015BYGC- -8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (mask ROM)
PD703017AF1- -EA6	121-pin plastic FBGA (12×12)	256 KB (mask ROM)
PD703017AF1- -EA6-A	121-pin plastic FBGA (12×12)	256 KB (mask ROM)
PD703017AGC- -8EU	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (mask ROM)
PD703017AGC- -8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (mask ROM)
PD703017AYF1- -EA6	121-pin plastic FBGA (12×12)	256 KB (mask ROM)
PD703017AYF1- -EA6-A	121-pin plastic FBGA (12×12)	256 KB (mask ROM)
PD703017AYGC- -8EU	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (mask ROM)
PD703017AYGC- -8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (mask ROM)
PD70F3015BF1-EA6-A	121-pin plastic FBGA (12×12)	128 KB (flash memory)
PD70F3015BGC-8EU	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (flash memory)
PD70F3015BGC-8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (flash memory)
PD70F3015BYF1-EA6-A	121-pin plastic FBGA (12×12)	128 KB (flash memory)
PD70F3015BYGC-8EU	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (flash memory)
PD70F3015BYGC-8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	128 KB (flash memory)
PD70F3017AF1-EA6	121-pin plastic FBGA (12×12)	256 KB (flash memory)
PD70F3017AF1-EA6-A	121-pin plastic FBGA (12×12)	256 KB (flash memory)
PD70F3017AGC-8EU	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (flash memory)
PD70F3017AGC-8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (flash memory)
PD70F3017AYF1-EA6	121-pin plastic FBGA (12×12)	256 KB (flash memory)
PD70F3017AYF1-EA6-A	121-pin plastic FBGA (12×12)	256 KB (flash memory)
PD70F3017AYGC-8EU	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (flash memory)
PD70F3017AYGC-8EU-A	100-pin plastic LQFP (fine pitch) (14×14)	256 KB (flash memory)

Remarks 1. “ ” indicates ROM code suffix.

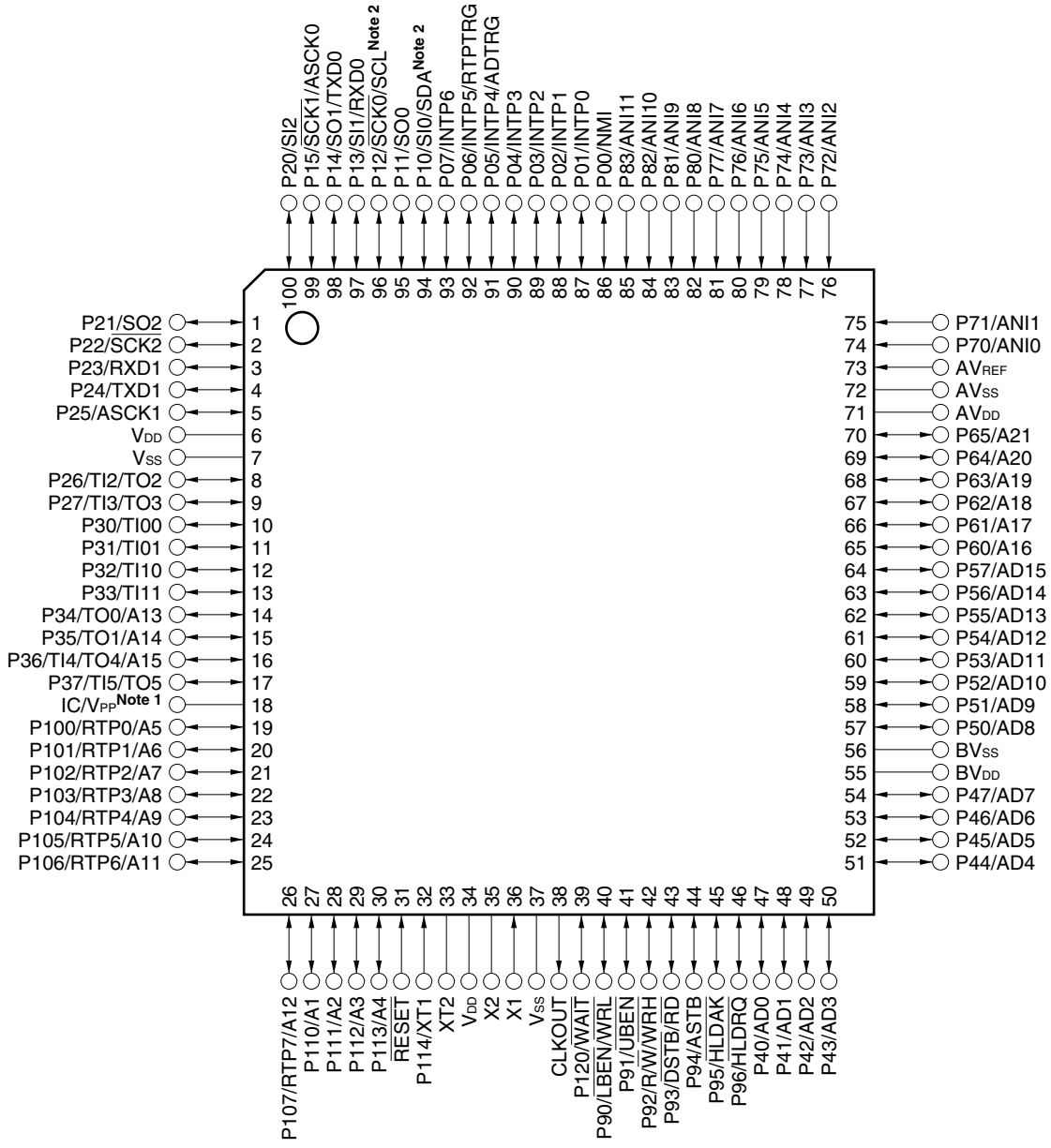
2. The V850/SA1 does not include any ROMless versions.

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

1.5 Pin Configuration

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

- μPD703014BGC-×××-8EU
- μPD703014BGC-×××-8EU-A
- μPD703014BYGC-×××-8EU
- μPD703014BYGC-×××-8EU-A
- μPD703015BGC-×××-8EU
- μPD703015BGC-×××-8EU-A
- μPD703015BYGC-×××-8EU
- μPD703015BYGC-×××-8EU-A
- μPD703017AGC-×××-8EU
- μPD703017AGC-×××-8EU-A
- μPD703017AYGC-×××-8EU
- μPD703017AYGC-×××-8EU-A
- μPD70F3015BGC-8EU
- μPD70F3015BGC-8EU-A
- μPD70F3015BYGC-8EU
- μPD70F3015BYGC-8EU-A
- μPD70F3017AGC-8EU
- μPD70F3017AGC-8EU-A
- μPD70F3017AYGC-8EU
- μPD70F3017AYGC-8EU-A

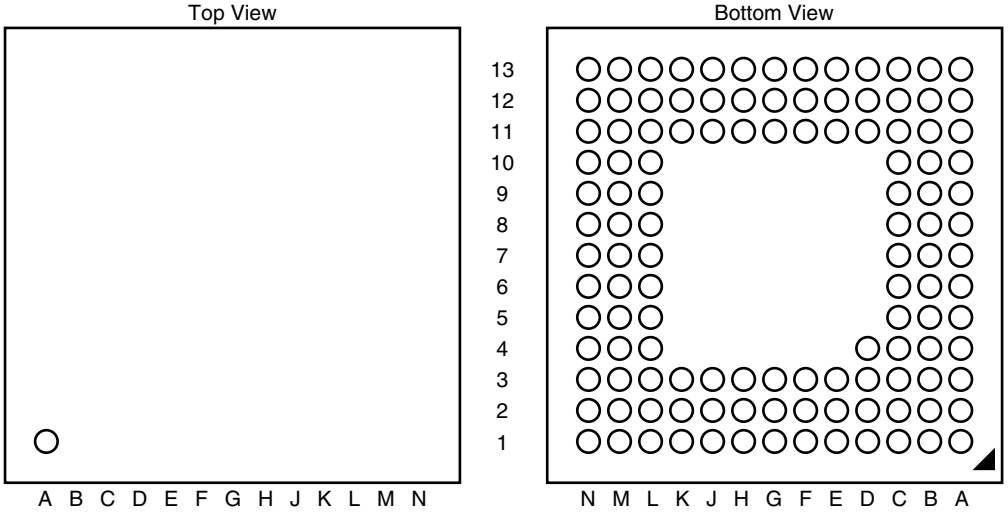


Notes 1. μPD703014B, 703014BY, 703015B, 703015BY, 703017A, 703017AY: IC (connect directly to VSS)
μPD70F3015B, 70F3015BY, 70F3017A, 70F3017AY: VPP (connect to VSS in normal operation mode)

2. SCL and SDA are available only in the μPD703014BY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

★ 121-pin plastic FBGA (12 × 12)

- μPD703014AF1-×××-EA6
- μPD703014AF1-×××-EA6-A
- μPD703014AYF1-×××-EA6
- μPD703014AYF1-×××-EA6-A
- μPD703014BF1-×××-EA6-A
- μPD703014BYF1-×××-EA6-A
- μPD703015AF1-×××-EA6
- μPD703015AF1-×××-EA6-A
- μPD703015AYF1-×××-EA6
- μPD703015AYF1-×××-EA6-A
- μPD703015BF1-×××-EA6-A
- μPD703015BYF1-×××-EA6-A
- μPD703017AF1-×××-EA6
- μPD703017AF1-×××-EA6-A
- μPD703017AYF1-×××-EA6
- μPD703017AYF1-×××-EA6-A
- μPD70F3015BF1-EA6-A
- μPD70F3015BYF1-EA6-A
- μPD70F3017AF1-EA6
- μPD70F3017AF1-EA6-A
- μPD70F3017AYF1-EA6
- μPD70F3017AYF1-EA6-A



Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
A1	P20	B8	P83	D2	V_{DD}	G11	P60	K13	BV_{DD}	M7	V_{SS}
A2	P15	B9	P80	D3	V_{SS}	G12	P56	L1	P104	M8	V_{SS}
A3	V_{SS}	B10	P75	D11	AV_{DD}	G13	P57	L2	P105	M9	P92
A4	P13	B11	AV_{SS}	D12	AV_{DD}	H1	P34	L3	$\overline{\text{RESET}}$	M10	P95
A5	P11	B12	AV_{SS}	D13	AV_{DD}	H2	P37	L4	V_{DD}	M11	P41
A6	P06	B13	P71	E1	P25	H3	P35	L5	V_{SS}	M12	P45
A7	P03	C1	P22	E2	V_{DD}	H11	P55	L6	X2	M13	P44
A8	P00	C2	P23	E3	P30	H12	P53	L7	P90	N1	P107
A9	P81	C3	V_{SS}	E11	AV_{DD}	H13	P54	L8	P120	N2	P110
A10	P76	C4	P24	E12	P64	J1	IC/V_{PP}[Note]	L9	P93	N3	P112
A11	P73	C5	P07	E13	P65	J2	IC/V_{PP}[Note]	L10	P96	N4	V_{DD}
A12	P72	C6	P04	F1	P26	J3	P100	L11	BV_{SS}	N5	XT1
A13	AV_{SS}	C7	P01	F2	P27	J11	P52	L12	BV_{SS}	N6	V_{SS}
B1	P21	C8	P82	F3	P33	J12	P50	L13	BV_{SS}	N7	V_{SS}
B2	P14	C9	P77	F11	P63	J13	P51	M1	P106	N8	CLKOUT
B3	V_{SS}	C10	P74	F12	P61	K1	P101	M2	P111	N9	P91
B4	P12	C11	AV_{SS}	F13	P62	K2	P102	M3	P113	N10	P94
B5	P10	C12	P70	G1	P31	K3	P103	M4	V_{DD}	N11	P40
B6	P05	C13	AV_{REF}	G2	P32	K11	P46	M5	XT2	N12	P42
B7	P02	D1	V_{DD}	G3	P36	K12	P47	M6	X1	N13	P43

Note μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, μPD703017A, 703017AY: IC (connect directly to V_{SS})
μPD70F3015B, 70F315BY, 70F3017A, 70F3017AY: V_{PP} (connect to V_{SS} in normal operation mode)

Remarks 1. Alternate pin names are omitted. Alternate pins are identical to the 100-pin plastic LQFP. However, SCL and SDA are available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 70F3015BY, 703017AY, and 70F3017AY.

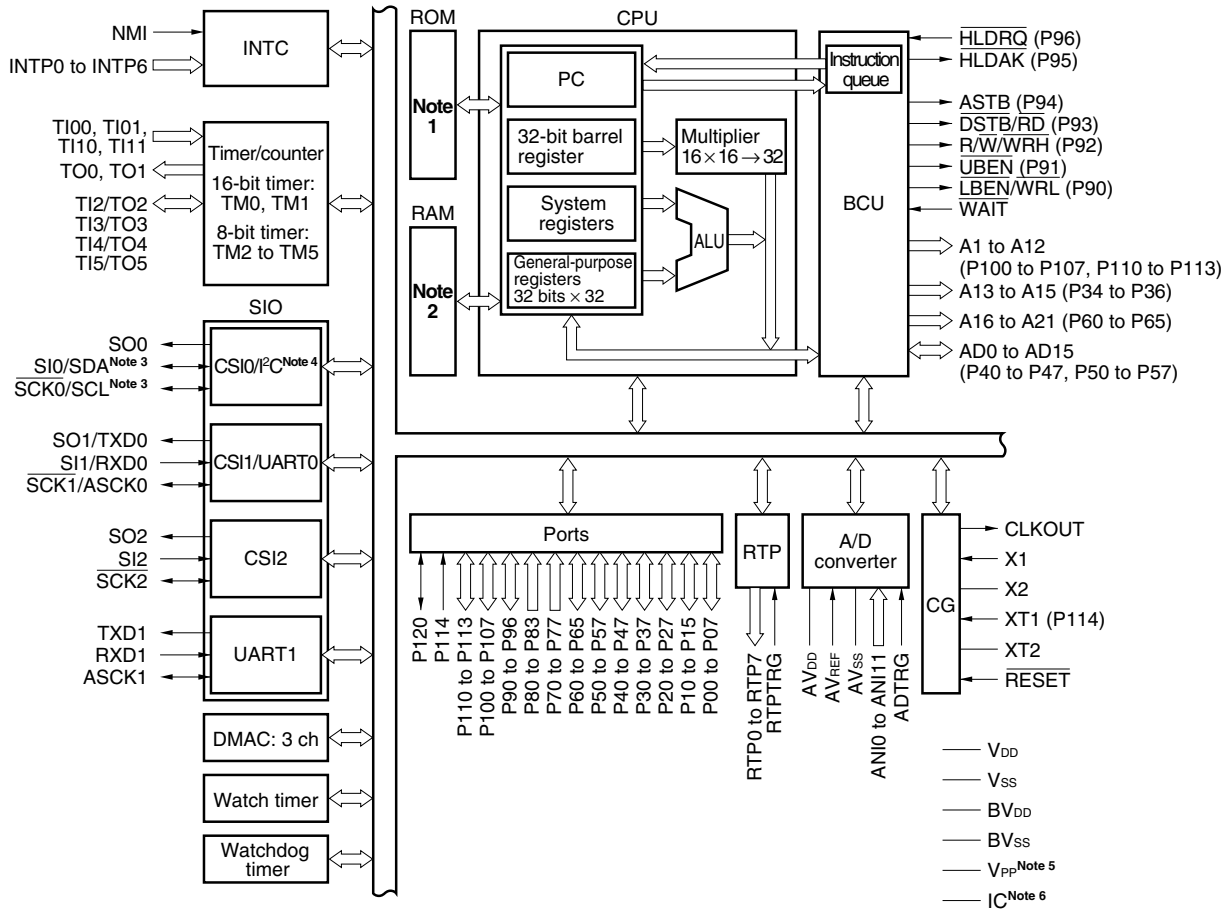
2. Connect the D4 pin directly to V_{SS}.

Pin Identification

A1 to A21:	Address bus	P90 to P96:	Port 9
AD0 to AD15:	Address/data bus	P100 to P107:	Port 10
ADTRG:	A/D trigger input	P110 to P114:	Port 11
ANI0 to ANI11:	Analog input	P120	Port 12
ASCK0, ASCK1:	Asynchronous serial clock	$\overline{RD}$:	Read strobe
ASTB:	Address strobe	$\overline{RESET}$:	Reset
AV_{DD}:	Power supply for analog	RTP0 to RTP7:	Real-time output port
AV_{REF}:	Analog reference voltage	RTPTRG:	RTP trigger
AV_{SS}:	Ground for analog	R/$\overline{W}$:	Read/write status
BV_{DD}:	Power supply for bus interface	RXD0, RXD1:	Receive data
BV_{SS}:	Ground for bus interface	$\overline{SCK0}$ to $\overline{SCK2}$:	Serial clock
CLKOUT:	Clock output	SCL:	Serial clock
$\overline{DSTB}$:	Data strobe	SDA:	Serial data
$\overline{HLDAK}$:	Hold acknowledge	SI0 to SI2:	Serial input
$\overline{HLDRQ}$:	Hold request	SO0 to SO2:	Serial output
IC:	Internally connected	TI00, TI01, TI10,	
INTP0 to INTP6:	Interrupt request from peripherals	TI11, TI2 to TI5:	Timer input
$\overline{LBEN}$:	Lower byte enable	TO0 to TO5:	Timer output
NMI:	Non-maskable interrupt request	TXD0, TXD1:	Transmit data
P00 to P07:	Port 0	$\overline{UBEN}$:	Upper byte enable
P10 to P15:	Port 1	V_{DD}:	Power supply
P20 to P27:	Port 2	V_{PP}:	Programming power supply
P30 to P37:	Port 3	V_{SS}:	Ground
P40 to P47:	Port 4	$\overline{WAIT}$:	Wait
P50 to P57:	Port 5	$\overline{WRH}$:	Write strobe high level data
P60 to P65:	Port 6	$\overline{WRL}$:	Write strobe low level data
P70 to P77:	Port 7	X1, X2:	Crystal for main clock
P80 to P83:	Port 8	XT1, XT2:	Crystal for subclock

1.6 Function Blocks

⋆ 1.6.1 Internal block diagram



Notes 1. μPD703014A, 703014AY, 703014B, 703014BY: 64 KB (mask ROM)
μPD703015A, 703015AY, 703015B, 703015BY: 128 KB (mask ROM)
μPD703017A, 703017AY: 256 KB (mask ROM)
μPD70F3015B, 70F3015BY: 128 KB (flash memory)
μPD70F3017A, 70F3017AY: 256 KB (flash memory)

2. μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 70F3015B, 70F3015BY: 4 KB
μPD703017A, 703017AY, 70F3017A, 70F3017AY: 8 KB

3. The SCL and SDA pins are available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

4. The I^2C function is available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

5. μPD70F3015B, 70F3015BY, 70F3017A and 70F3017AY

6. μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, and 703017AY

1.6.2 Internal units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as the multiplier (16 bits × 16 bits → 32 bits) and the 32-bit barrel shifter help accelerate processing of complex instructions.

★ **(2) Bus control unit (BCU)**

The BCU starts a required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory space and the CPU does not send a bus cycle start request, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is stored in an instruction queue.

★ **(3) ROM**

This consists of a mask ROM or flash memory mapped to the address space starting at 00000000H.

ROM can be accessed by the CPU in one clock cycle during instruction fetch. The internal ROM capacity and internal ROM area vary as follows according to the product.

Product Name	Internal ROM Capacity	Internal ROM Area
μPD703014A, 703014AY, 703014B, 703014BY	64 KB (mask ROM)	xx000000H to xx00FFFFH
μPD703015A, 703015AY, 703015B, 703015BY	128 KB (mask ROM)	xx000000H to xx01FFFFH
μPD70F3015B, 70F3015BY	128 KB (flash memory)	
μPD703017A, 703017AY	256 KB (mask ROM)	xx000000H to xx03FFFFH
μPD70F3017A, 70F3017AY	256 KB (flash memory)	

★ **(4) RAM**

The internal RAM capacity and internal RAM area vary as follows according to the product. RAM can be accessed by the CPU in one clock cycle during data access.

Product Name	Internal RAM Capacity	Internal RAM Area
μPD703014A, 703014AY, 703015B, 703015BY	4 KB	xxFFE000H to xxFFEFFFH
μPD703015A, 703015AY, 703015B, 703015BY		
μPD70F3015B, 703015BY		
μPD703017A, 703017AY	8 KB	xxFFD000H to xxFFEFFFH
μPD70F3017A, 70F3017AY		

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0 to INTP6) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple servicing control can be performed for interrupt sources.

(6) Clock generator (CG)

The clock generator includes two types of oscillators: one each for the main clock (f_{XX}) and subclock (f_{XT}), generates five types of clocks (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, and f_{XT}), and supplies one of them as the operating clock for the CPU (f_{CPU}).

(7) Timer/counter

A two-channel 16-bit timer/event counter and a four-channel 8-bit timer/event counter are equipped, enabling measurement of pulse intervals and frequency as well as programmable pulse output.

The two-channel 8-bit timer/event counter can be connected via a cascade connection to enable use as a 16-bit timer.

(8) Watch timer

This timer generates an interrupt of the reference time period (0.5 seconds) for counting the clock (the 32.768 kHz subclock or the 16.777 MHz main clock). At the same time, the watch timer can be used as an interval timer for the main clock.

(9) Watchdog timer

A watchdog timer is equipped to detect program loops, system abnormalities, etc.

It can also be used as an interval timer.

When used as a watchdog timer, it generates a non-maskable interrupt request (INTWDT) after an overflow occurs. When used as an interval timer, it generates a maskable interrupt request (INTWDTM) after an overflow occurs.

(10) Serial interface (SIO)

The V850/SA1 includes four serial interface channels: for the asynchronous serial interface (UART0, UART1), clocked serial interface (CSI0 to CSI2), and I^2C bus interface. One of these channels is switchable between the UART and CSI and another is switchable between CSI and I^2C. Two channels are fixed to UART and CSI, respectively.

For UART 0 and UART1, data is transferred via the TXD0, TXD1, RXD0, and RXD1 pins.

For CSI0 to CSI2, data is transferred via the SO0 to SO2, SI0 to SI2, and $\overline{SCK0}$ to $\overline{SCK2}$ pins.

★ For I^2C, data is transferred via the SDA and SCL pins. I^2C is equipped only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

UART also has a two-channel dedicated baud rate generator.

(11) A/D converter

This high-speed, high-resolution 10-bit A/D converter includes 12 analog input pins. Conversion uses the successive approximation method.

(12) DMA controller

A three-channel DMA controller is equipped. This controller transfers data between the internal RAM and on-chip peripheral I/O devices in response to interrupt requests sent by on-chip peripheral I/O.

(13) Real-time output port (RTP)

The RTP is a real-time output function that transfers previously set 8-bit data to an output latch when an external trigger signal or timer compare register match signal occurs. It can also be used in a 4-bit $\times$ 2-channel configuration.

(14) Ports

As shown below, the following ports have general-purpose port functions and control pin functions.

Port	I/O	Port Function	Control Function
Port 0	8-bit I/O	General-purpose port	NMI, external interrupt, A/D converter trigger, RTP trigger
Port 1	6-bit I/O		Serial interface
Port 2	8-bit I/O		Serial interface, timer I/O
Port 3	8-bit I/O		Timer I/O, external address bus
Port 4	8-bit I/O		External address/data bus
Port 5			
Port 6	6-bit I/O		External address bus
Port 7	8-bit input		A/D converter analog input
Port 8	4-bit input		
Port 9	7-bit I/O		External bus interface control signal I/O
Port 10	8-bit I/O		Real-time output port, external address bus
Port 11	4-bit I/O, 1-bit input		External address bus, subclock input
Port 12	1-bit I/O		Wait control

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The names and functions of the pins of the V850/SA1 are described below divided into port pins and non-port pins.

There are three types of power supplies for the pin I/O buffers: AV_{DD}, BV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is described below.

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

Power Supply	Corresponding Pins	Usable Voltage Range
AV_{DD}	Port 7, port 8	2.7 V ≤ AV_{DD} ≤ 3.6 V
BV_{DD}	Port 4, port 5, port 6, port 9, port 12, CLKOUT	2.7 V ≤ BV_{DD} ≤ 3.6 V
V_{DD}	Port 0, port 1, port 2, port 3, port 10, port 11, $\overline{\text{RESET}}$	2.7 V ≤ V_{DD} ≤ 3.6 V

(1) Port pins

(1/3)

Pin Name	I/O	PULL	Function	Alternate Function
P00	I/O	Yes	Port 0 8-bit I/O port Input/output mode can be specified in 1-bit units.	NMI
P01				INTP0
P02				INTP1
P03				INTP2
P04				INTP3
P05				INTP4/ADTRG
P06				INTP5/RTPTRG
P07				INTP6
P10	I/O	Yes	Port 1 6-bit I/O port Input/output mode can be specified in 1-bit units.	SI0/SDA[Note]
P11				SO0
P12				$\overline{\text{SCK0}}$/SCL[Note]
P13				SI1/RXD0
P14				SO1/TXD0
P15				$\overline{\text{SCK1}}$/ASCK0
P20	I/O	Yes	Port 2 8-bit I/O port Input/output mode can be specified in 1-bit units.	SI2
P21				SO2
P22				$\overline{\text{SCK2}}$
P23				RXD1
P24				TXD1
P25				ASCK1
P26				TI2/TO2
P27				TI3/TO3

★ **Note** μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only

Remark PULL: On-chip pull-up resistor

(2/3)

Pin Name	I/O	PULL	Function	Alternate Function
P30	I/O	Yes	Port 3 8-bit I/O port Input/output mode can be specified in 1-bit units.	TI00
P31				TI01
P32				TI10
P33				TI11
P34				TO0/A13
P35				TO1/A14
P36				TI4/TO4/A15
P37				TI5/TO5
P40	I/O	No	Port 4 8-bit I/O port Input/output mode can be specified in 1-bit units.	AD0
P41				AD1
P42				AD2
P43				AD3
P44				AD4
P45				AD5
P46				AD6
P47				AD7
P50	I/O	No	Port 5 8-bit I/O port Input/output mode can be specified in 1-bit units.	AD8
P51				AD9
P52				AD10
P53				AD11
P54				AD12
P55				AD13
P56				AD14
P57				AD15
P60	I/O	No	Port 6 6-bit I/O port Input/output mode can be specified in 1-bit units.	A16
P61				A17
P62				A18
P63				A19
P64				A20
P65				A21

Remark PULL: On-chip pull-up resistor

(3/3)

Pin Name	I/O	PULL	Function	Alternate Function
P70	Input	No	Port 7 8-bit input port	ANI0
P71				ANI1
P72				ANI2
P73				ANI3
P74				ANI4
P75				ANI5
P76				ANI6
P77				ANI7
P80	Input	No	Port 8 4-bit input port	ANI8
P81				ANI9
P82				ANI10
P83				ANI11
P90	I/O	No	Port 9 7-bit I/O port Input/output mode can be specified in 1-bit units.	$\overline{\text{LBEN}}$/$\overline{\text{WRL}}$
P91				$\overline{\text{UBEN}}$
P92				R/$\overline{\text{W}}$/$\overline{\text{WRH}}$
P93				$\overline{\text{DSTB}}$/$\overline{\text{RD}}$
P94				ASTB
P95				$\overline{\text{HLDAK}}$
P96				$\overline{\text{HLDRQ}}$
P100	I/O	Yes	Port 10 8-bit I/O port Input/output mode can be specified in 1-bit units.	RTP0/A5
P101				RTP1/A6
P102				RTP2/A7
P103				RTP3/A8
P104				RTP4/A9
P105				RTP5/A10
P106				RTP6/A11
P107				RTP7/A12
P110	I/O	Yes	Port 11 5-bit I/O port Input/output mode can be specified in 1-bit units. P114 is fixed to input mode.	A1
P111				A2
P112				A3
P113				A4
P114	Input	No		XT1
P120	I/O	No	Port 12 1-bit I/O port	$\overline{\text{WAIT}}$

Remark PULL: On-chip pull-up resistor

(2) Non-port pins

(1/2)

Pin Name	I/O	PULL	Function	Alternate Function
A1 to A4	Output	Yes	Lower address bus used for external memory expansion	P110 to P113
A5 to A12				P100/RTP0 to P107/RTP7
A13				P34/TO0
A14				P35/TO1
A15				P36/TI4/TO4
A16 to A21	Output	No	Higher address bus used for external memory expansion	P60 to P65
AD0 to AD7	I/O	No	16-bit multiplexed address/data bus used for external memory expansion	P40 to P47
AD8 to AD15				P50 to P57
ADTRG	Input	Yes	A/D converter external trigger input	P05/INTP4
ANI0 to ANI7	Input	No	Analog input to A/D converter	P70 to P77
ANI8 to ANI11	Input	No		P80 to P83
ASCK0	Input	Yes	Serial baud rate clock input for UART0 and UART1	P15/$\overline{\text{SCK1}}$
ASCK1				P25
ASTB	Output	No	External address strobe signal output	P94
AV_{DD}	—	—	Positive power supply for A/D converter	—
AV_{REF}	Input	—	Reference voltage input for A/D converter	—
AV_{SS}	—	—	Ground potential for A/D converter	—
BV_{DD}	—	—	Positive power supply for bus interface	—
BV_{SS}	—	—	Ground potential for bus interface	—
CLKOUT	Output	—	Internal system clock output	—
$\overline{\text{DSTB}}$	Output	No	External data strobe signal output	P93/$\overline{\text{RD}}$
$\overline{\text{HLDAK}}$	Output	No	Bus hold acknowledge output	P95
$\overline{\text{HLDRQ}}$	Input	No	Bus hold request input	P96
INTP0 to INTP3	Input	Yes	External interrupt request input (analog noise elimination)	P01 to P04
INTP4			External interrupt request input (digital noise elimination)	P05/ADTRG
INTP5				P06/RTPTRG
INTP6				P07
$\overline{\text{LBEN}}$	Output	No	External data bus's lower byte enable signal output	P90/$\overline{\text{WRL}}$
NMI	Input	Yes	Non-maskable interrupt request input (analog noise elimination)	P00
$\overline{\text{RD}}$	Output	No	Read strobe signal output	P93/$\overline{\text{DSTB}}$
$\overline{\text{RESET}}$	Input	–	System reset input	—
RTP0 to RTP7	Output	Yes	Real-time output port	P100/A5 to P107/A12
RTPTRG	Input	Yes	RTP external trigger input	P06/INTP5
R/$\overline{\text{W}}$	Output	No	External read/write status output	P92/$\overline{\text{WRH}}$
RXD0	Input	Yes	Serial receive data input for UART0 and UART1	P13/SI1
RXD1				P23
$\overline{\text{SCK0}}$	I/O	Yes	Serial clock I/O (3-wire type) for CSI0 to CSI2	P12/SCL[Note]

★ **Note** μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only

Remark PULL: On-chip pull-up resistor

(2/2)

Pin Name	I/O	PULL	Function	Alternate Function
$\overline{SCK1}$	I/O	Yes	Serial clock I/O (3-wire type) for CSI0 to CSI2	P15/ASCK0
$\overline{SCK2}$				P22
SCL[Note 1]	I/O	Yes	Serial clock I/O for I^2C	P12/$\overline{SCK0}$
SDA[Note 1]	I/O	Yes	Serial transmit/receive data I/O for I^2C	P10/SI0
SI0	Input	Yes	Serial receive data input (3-wire type) for CSI0 to CSI2	P10/SDA[Note 1]
SI1				P13/RXD0
SI2				P20
SO0	Output	Yes	Serial transmit data output (3-wire type) for CSI0 to CSI2	P11
SO1				P14/TXD0
SO2				P21
TI00	Input	Yes	External count clock input for TM0/external capture trigger input	P30
TI01			External capture trigger input for TM0	P31
TI10			External count clock input for TM1/external capture trigger input	P32
TI11			External capture trigger input for TM1	P33
TI2			External count clock input for TM2	P26/TO2
TI3			External count clock input for TM3	P27/TO3
TI4			External count clock input for TM4	P36/TO4/A15
TI5			External count clock input for TM5	P37/TO5
TO0, TO1	Output	Yes	Pulse signal output for TM0, TM1	P34/A13, P35/A14
TO2			Pulse signal output for TM2	P26/TI2
TO3			Pulse signal output for TM3	P27/TI3
TO4			Pulse signal output for TM4	P36/TI4/A15
TO5			Pulse signal output for TM5	P37/TI5
TXD0	Output	Yes	Serial transmit data output for UART0 and UART1	P14/SO1
TXD1				P24
$\overline{UBEN}$	Output	No	Higher byte enable signal output for external data bus	P91
V_{DD}	—	—	Positive power supply pin	—
V_{PP}[Note 2]	—	—	High-voltage application pin for program write/verify	—
V_{SS}	—	—	Ground potential	—
$\overline{WAIT}$	Input	Yes	Control signal input for inserting wait in bus cycle	P120
$\overline{WRH}$	Output	No	Higher byte write strobe signal output for external data bus	P92/R/$\overline{W}$
$\overline{WRL}$			Lower byte write strobe signal output for external data bus	P90/$\overline{LBEN}$
X1	Input	No	Resonator connection for main clock	—
X2	—			—
XT1	Input	No	Resonator connection for subclock	P114
XT2	—			—
IC[Note 3]	—	—	Internally connected	—

★ **Notes 1.** μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only

★ **2.** μPD70F3015B, 70F3015BY, 70F3017A, and 70F3017AY only

★ **3.** μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, and 703017AY only

Remark PULL: On-chip pull-up resistor

2.2 Pin States

The operating states of various pins are described below with reference to their operating modes.

★ **Table 2-2. Operating States of Pins in Each Operating Mode**

Pin \ Operating State	Reset[Note 1]	HALT Mode/ Idle State	IDLE Mode/ Software STOP Mode	Bus Hold	Bus Cycle Inactive[Note 2]
AD0 to AD15	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
A1 to A15	Hi-Z	Held	Held	Held	Held[Note 3]
A16 to A21	Hi-Z	Held	Hi-Z	Hi-Z	Held[Note 3]
$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	Hi-Z	Held	Hi-Z	Hi-Z	Held[Note 3]
R/$\overline{\text{W}}$	Hi-Z	H	Hi-Z	Hi-Z	H
$\overline{\text{DSTB}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$	Hi-Z	H	Hi-Z	Hi-Z	H
ASTB	Hi-Z	H	Hi-Z	Hi-Z	H
$\overline{\text{HLDRQ}}$	—	Operating	—	Operating	Operating
$\overline{\text{HLDAK}}$	Hi-Z	Operating	Hi-Z	L	Operating
$\overline{\text{WAIT}}$	—	—	—	—	—
CLKOUT	Hi-Z	Operating[Note 4]	L	Operating[Note 4]	Operating[Note 4]

Notes 1. Pins (except the CLKOUT pin) are used as port pins (input mode) after reset.

2. The bus cycle inactivation timing occurs when the internal memory area is specified by the program counter (PC) in the external expansion mode.

3.
- When the external memory area has not been accessed even once after reset is released and the external expansion mode is set: Undefined
- When the bus cycle is inactivated after access to the external memory area, or when the external memory area has not been accessed even once after the external expansion mode is released and set again: The state of the external bus cycle when the external memory area accessed last is held.

4. Low level (L) when in clock output inhibit mode

Remark Hi-Z: High impedance
Held: State is held during previously set external bus cycle
L: Low-level output
H: High-level output
–: Input without sampling sampled (not acknowledged)

2.3 Description of Pin Functions

(1) P00 to P07 (Port 0) ··· 3-state I/O

P00 to P07 constitute an 8-bit I/O port that can be set to input or output in 1-bit units.

P00 to P07 can also function as an NMI input, external interrupt request inputs, external trigger for the A/D converter, and external trigger for the real-time output port. The valid edges of the NMI and INTP0 to INTP6 pins are specified by the EGP0 and EGN0 registers.

(a) Port function

P00 to P07 can be set to input or output in 1-bit units using the port 0 mode register (PM0).

(b) Alternate functions

(i) NMI (Non-maskable interrupt request) ··· input

This is a non-maskable interrupt request signal input pin.

(ii) INTP0 to INTP6 (Interrupt request from peripherals) ··· input

These are external interrupt request input pins.

(iii) ADTRG (AD trigger input) ··· input

This is the A/D converter's external trigger input pin. This pin is controlled by the A/D converter mode register (ADM).

(iv) RTPTRG (Real-time output port trigger input) ··· input

This is the real-time output port's external trigger input pin. This pin is controlled by the real-time output port control register (RTPC).

(2) P10 to P15 (Port 1) ··· 3-state I/O

P10 to P15 constitute a 6-bit I/O port that can be set to input or output in 1-bit units.

P10 to P15 can also function as input or output pins for the serial interface.

P10 to P12, P14, and P15 can be selected as normal output or N-ch open-drain output.

(a) Port function

P10 to P15 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(b) Alternate functions

(i) SI0, SI1 (Serial input 0, 1) ··· input

These are the serial receive data input pins of CSI0 and CSI1.

(ii) SO0, SO1 (Serial output 0, 1) ··· output

These are the serial transmit data output pins of CSI0 and CSI1.

(iii) $\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$ (Serial clock 0, 1) ··· 3-state I/O

These are the serial clock I/O pins for CSI0 and CSI1.

★ **(iv) SDA (Serial data) ··· I/O**

This is the serial transmit/receive data I/O pin for I^2C (μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only).

★ **(v) SCL (Serial clock) ··· I/O**

This is the serial clock I/O pin for I^2C (μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only).

(vi) RXD0 (Receive data 0) ··· input

This is the serial receive data input pin of UART0.

(vii) TXD0 (Transmit data 0) ··· output

This is the serial transmit data output pin of UART0.

(viii) ASCK0 (Asynchronous serial clock 0) ··· input

This is the serial baud rate clock input pin of UART0.

(3) P20 to P27 (Port 2) ··· 3-state I/O

P20 to P27 constitute an 8-bit I/O port that can be set to input or output in 1-bit units.

P20 to P27 can also function as input or output pins for the serial interface, and input or output pins for the timer/counter.

P21 and P22 can be selected as normal output or N-ch open-drain output.

(a) Port function

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

(b) Alternate functions

(i) SI2 (Serial input 2) ··· input

This is the serial receive data input pin of CSI2.

(ii) SO2 (Serial output 2) ··· output

This is the serial transmit data output pin of CSI2.

(iii) $\overline{\text{SCK2}}$ (Serial clock 2) ··· 3-state I/O

This is the serial clock I/O pin of CSI2.

(iv) RXD1 (Receive data 1) ... input

This is the serial receive data input pin of UART1.

(v) TXD1 (Transmit data 1) ... output

This is the serial transmit data output pin of UART1.

(vi) ASCK1 (Asynchronous serial clock 1) ... input

This is the serial baud rate clock input pin of UART1.

(vii) TI2 and TI3 (Timer input 2, 3) ... input

These are the external count clock input pins for timer 2 and timer 3.

(viii) TO2 and TO3 (Timer output 2, 3) ... output

These are the pulse signal output pins for timer 2 and timer 3.

(4) P30 to P37 (Port 3) ··· 3-state I/O

P30 to P37 constitute an 8-bit I/O port that can be set to input or output in 1-bit units.

P30 to P37 can also function as input or output pins for the timer/counter, and an address bus (A13 to A15) when memory is expanded externally.

(a) Port function

P30 to P37 can be set to input or output in 1-bit units using the port 3 mode register (PM3).

(b) Alternate functions

(i) TI00, TI01, TI10, TI11, TI4, TI5 (Timer input 00, 01, 10, 11, 4, 5) ··· input

These are the external count clock input pins of timer 0, timer 1, timer 4, and timer 5.

(ii) TO0, TO1, TO4, TO5 (Timer output 0, 1, 4, 5) ··· output

These are the pulse signal output pins of timer 0, timer 1, timer 4, and timer 5.

(iii) A13 to A15 (Address 13 to 15) ··· output

These comprise an address bus that is used for external access. These pins operate as the A13 to A15 bit address output pins within a 22-bit address. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle to inactive, the previous bus cycle's address is retained.

(5) P40 to P47 (Port 4) ··· 3-state I/O

P40 to P47 constitute an 8-bit I/O port that can be set to input or output pins in 1-bit units.

P40 to P47 can also function as a time division address/data bus (AD0 to AD7) when memory is expanded externally.

The I/O signal level uses the bus interface power supply pins BV_{DD} and BV_{SS} as a reference.

(a) Port function

P40 to P47 can be set to input or output in 1-bit units using the port 4 mode register (PM4).

(b) Alternate functions (External expansion function)

P40 to P47 can be set as AD0 to AD7 using the memory expansion mode register (MM).

(i) AD0 to AD7 (Address/data 0 to 7) ··· 3-state I/O

These comprise a multiplexed address/data bus that is used for external access. At the address timing (T1 state), these pins operate as AD0 to AD7 (22-bit address) output pins. At the data timing (T2, TW, T3), they operate as the lower 8-bit I/O bus pins for 16-bit data. The output changes in synchronization with the rising edge of the clock in each state within the bus cycle. When the timing sets the bus cycle to inactive, these pins go into a high-impedance state.

(6) P50 to P57 (Port 5) ··· 3-state I/O

P50 to P57 constitute an 8-bit I/O port that can be set to input or output in 1-bit units.

P50 to P57 can also function as I/O port pins and as a time division address/data buses (AD8 to AD15) when memory is expanded externally.

The I/O signal level uses the bus interface power supply pins BV_{DD} and BV_{SS} as reference.

(a) Port function

P50 to P57 can be set to input or output in 1-bit units using the port 5 mode register (PM5).

(b) Alternate functions (External expansion function)

P50 to P57 can be set as AD8 to AD15 using the memory expansion mode register (MM).

(i) AD8 to AD15 (Address/data 8 to 15) ··· 3-state I/O

These comprise a multiplexed address/data bus that is used for external access. At the address timing (T1 state), these pins operate as AD8 to AD15 (22-bit address) output pins. At the data timing (T2, TW, T3), they operate as the higher 8-bit I/O bus pins for 16-bit data. The output changes in synchronization with the rising edge of the clock in each state within the bus cycle. When the timing sets the bus cycle to inactive, these pins go into a high-impedance state.

(7) P60 to P65 (Port 6) ··· 3-state I/O

P60 to P65 constitute a 6-bit I/O port that can be set to input or output in 1-bit units.

★ P60 to P65 can also function as an address bus (A16 to A21) when memory is expanded externally. When the port 6 is accessed in 8-bit units, the higher 2 bits of port 6 are ignored when they are written to and 00 is read when they are read.

The I/O signal level uses the bus interface power supply pins BV_{DD} and BV_{SS} as reference.

(a) Port function

P60 to P65 can be set to input or output in 1-bit units using the port 6 mode register (PM6).

(b) Alternate functions (External expansion function)

P60 to P65 can be set as A16 to A21 using the memory expansion mode register (MM).

(i) A16 to A21 (Address 16 to 21) ··· output

These comprise an address bus that is used for external access. These pins operate as the higher 6-bit address output pins within a 22-bit address. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle to inactive, the previous bus cycle's address is retained.

(8) P70 to P77 (Port 7), P80 to P83 (Port 8) ··· input

P70 to P77 constitute an 8-bit input-only port in which all the pins are fixed to input mode. P80 to P83 constitute a 4-bit input-only port in which all the pins are fixed to input.

P70 to P77 and P80 to P83 can also function as analog input pins for the A/D converter.

(a) Port function

P70 to P77 and P80 to P83 are input-only pins.

(b) Alternate functions

P70 to P77 also function as ANI0 to ANI7 and P80 to P83 also function as ANI8 to ANI11, but these alternate functions are not switchable.

(i) ANI0 to ANI11 (Analog input 0 to 11) ··· input

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

Connect a capacitor between these pins and AV_{SS} to prevent noise-related operation faults. Also, do not apply voltage that is outside the range for AV_{SS} and AV_{REF} to pins that are being used as inputs for the A/D converter. If it is possible for noise above the AV_{REF} range or below the AV_{SS} to enter, clamp these pins using a diode that has a small V_F value.

(9) P90 to P96 (Port 9) ··· 3-state I/O

P90 to P96 constitute a 7-bit I/O port that can be set to input or output pins in 1-bit units.

P90 to P96 can also function as control signal output pins and bus hold control signal output pins when memory is expanded externally.

During 8-bit access of port 9, the highest bit is ignored during a write operation and is read as a "0" during a read operation.

The I/O signal level uses the bus interface power supply pins BV_{DD} and BV_{SS} as a reference.

(a) Port function

P90 to P96 can be set to input or output in 1-bit units using the port 9 mode register (PM9).

(b) Alternate functions (External expansion function)

P90 to P96 can be set to operate as control signal outputs for external memory expansion using the memory expansion mode register (MM).

(i) $\overline{\text{LBEN}}$ (Lower byte enable) ··· output

This is a lower byte enable signal output pin for the external 16-bit data bus. During byte access of odd-numbered addresses, these pins are set as inactive (high level). The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle as inactive, the previous bus cycle's address is retained.

(ii) $\overline{\text{UBEN}}$ (Upper byte enable) ··· output

This is an upper byte enable signal output pin for the external 16-bit data bus. During byte access of even-numbered addresses, these pins are set as inactive (high level). The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle as inactive, the previous bus cycle's address is retained.

Access		$\overline{\text{UBEN}}$	$\overline{\text{LBEN}}$	A0
Word access		0	0	0
Halfword access		0	0	0
Byte access	Even-numbered address	1	0	0
	Odd-numbered address	0	1	1

(iii) R/$\overline{\text{W}}$ (Read/write status) ··· output

This is an output pin for the status signal pin that indicates whether the bus cycle is a read cycle or write cycle during external access. High level is set during a read cycle and low level is set during a write cycle. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. High level is set when the timing sets the bus cycle as inactive.

(iv) $\overline{\text{DSTB}}$ (Data strobe) ··· output

This is an output pin for the external data bus's access strobe signal. Output becomes active (low level) during the T2 and TW states of the bus cycle. Output becomes inactive (high level) when the timing sets the bus cycle as inactive.

(v) ASTB (Address strobe) ··· output

This is an output pin for the external address bus's latch strobe signal. Output becomes active (low level) in synchronization with the falling edge of the clock during the T1 state of the bus cycle, and becomes inactive (high level) in synchronization with the falling edge of the clock during the T3 state of the bus cycle. Output becomes inactive when the timing sets the bus cycle as inactive.

(vi) $\overline{\text{HLDAK}}$ (Hold acknowledge) ··· output

This is an output pin for the acknowledge signal that indicates high impedance status for the address bus, data bus, and control bus when the V850/SA1 receives a bus hold request.

The address bus, data bus, and control bus are set to high impedance status when this signal is active.

(vii) $\overline{\text{HLDRQ}}$ (Hold request) ··· input

This is an input pin by which an external device requests the V850/SA1 to release the address bus, data bus, and control bus. This pin accepts asynchronous input for CLKOUT. When this pin is active, the address bus, data bus, and control bus are set to high impedance status. This occurs either when the V850/SA1 completes execution of the current bus cycle or immediately if no bus cycle is being executed, then the $\overline{\text{HLDAK}}$ signal is set as active and the bus is released.

(viii) $\overline{\text{WRL}}$ (Write strobe low level data) ··· output

This is a write strobe signal output pin for the lower data in the external 16-bit data bus. Output occurs during the write cycle, similar to $\overline{\text{DSTB}}$.

(ix) $\overline{\text{WRH}}$ (Write strobe high level data) ··· output

This is a write strobe signal output pin for the higher data in the external 16-bit data bus. Output occurs during the write cycle, similar to $\overline{\text{DSTB}}$.

(x) $\overline{\text{RD}}$ (Read strobe) ··· output

This is a read strobe signal output pin for the external 16-bit data bus. Output occurs during the read cycle, similar to $\overline{\text{DSTB}}$.

(10) P100 to P107 (Port 10) ··· 3-state I/O

P100 to P107 constitute an 8-bit I/O port that can be set to input or output in 1-bit units.

P100 to P107 can also function as a real-time output port and an address bus (A5 to A12) when memory is expanded externally.

P100 to P107 can be selected as normal output or N-ch open-drain output.

(a) Port function

P100 to P107 can be set to input or output in 1-bit units using the port 10 mode register (PM10).

(b) Alternate functions

(i) RTP0 to RTP7 (Real-time output port 0 to 7) ··· output

These pins comprise a real-time output port.

(ii) A5 to A12 (Address 5 to 12) ··· output

These comprise the address bus that is used for external access. These pins operate as the A5 to A12 bit address output pins within a 22-bit address. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle as inactive, the previous bus cycle's address is retained.

(11) P110 to P114 (Port 11) ··· 3-state I/O

P110 to P114 constitute a 5-bit I/O port that can be set to input or output in 1-bit units. However, P114 is fixed as the XT1 input pin.

P110 to P113 can also function as an address bus (A1 to A4) when memory is expanded externally.

(a) Port function

P110 to P114 can be set to input or output in 1-bit units using the port 11 mode register (PM11). However, P114 is fixed as an input pin.

(b) Alternate functions

(i) A1 to A4 (Address 1 to 4) ··· output

These comprise the address bus that is used for external access. These pins operate as the lower 4-bit address output pins within a 22-bit address. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle as inactive, the previous bus cycle's address is retained.

(ii) XT1 (Crystal for subclock) ··· input

This is the pin that connects a resonator for subclock generation.

The external clock can also be input to this pin. At this time, input a clock signal to the XT1 pin and its inverted signal to the XT2 pin.

(12) P120 (Port 12) ··· 3-state I/O

P120 is a 1-bit I/O port that can be set to input or output in 1-bit units. P120 can also function as a control signal ($\overline{WAIT}$) pin when a wait is inserted in the bus cycle.

(a) Port function

P120 can be set to input or output using the port 12 mode register (PM12).

(b) Alternate functions

(i) $\overline{WAIT}$ (Wait) ··· input

This is an input pin for the control signal used to insert waits into the bus cycle. This pin is sampled at the falling edge of the clock during the T2 or TW state of the bus cycle.

★ **(13) CLKOUT (Clock out) ··· output**

This is the pin used to output the bus clock generated internally.

(14) $\overline{RESET}$ (Reset) ··· input

The $\overline{RESET}$ pin is an asynchronous input and inputs a signal that has a constant low level width regardless of the status of the operating clock. When this signal is input, a system reset is executed as the first priority ahead of all other operations.

In addition to being used for ordinary initialization/start operations, this pin can also be used to release a standby mode (HALT, IDLE, or software STOP mode).

(15) X1 and X2 (Crystal)

These pins are used to connect the resonator that generates the main clock.

These pins can also be used to input an external clock. When inputting an external clock, connect the X1 pin and leave the X2 pin open.

(16) XT2 (Crystal for subclock)

This pin is used to connect the resonator that generates the subclock.

(17) AV_{DD} (Power supply for analog)

This is the analog positive power supply pin for the A/D converter. Be sure to keep the same potential as the V_{DD} pin.

(18) AV_{SS} (Ground for analog)

This is the ground pin for the A/D converter.

(19) AV_{REF} (Analog reference voltage) … input

This is the reference voltage supply pin for the A/D converter. Be sure to keep the same potential as the AV_{DD} pin.

(20) BV_{DD} (Power supply for bus interface)

This is the positive power supply pin for the bus interface and its alternate-function port. Be sure to keep the same potential as the V_{DD} pin.

(21) BV_{SS} (Ground for bus interface)

This is the ground pin for the bus interface and its alternate-function port.

(22) V_{DD} (Power supply)

This is the positive power supply pin. All V_{DD} pins should be connected to a positive power source.

(23) V_{SS} (Ground)

This is the ground pin. All V_{SS} pins should be grounded.

★ **(24) V_{PP} (Programming power supply)**

This is the positive power supply pin used for flash memory programming mode.

This pin is used in the μPD70F3015B, 70F3015BY, 70F3017A, and 70F3017AY. In normal operation mode, connect directly to V_{SS}.

★ **(25) IC (Internally connected)**

This is an internally connected pin used in the μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, and 703017AY. Be sure to connect directly to V_{SS}.

★ 2.4 Pin I/O Circuits and Recommended Connection of Unused Pins

(1/2)

Pin	Alternate Function	I/O Circuit Type	Recommended Connection Method
P00	NMI	8-A	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01 to P04	INTP0 to INTP3		
P05	INTP4/ADTRG		
P06	INTP5/RTPTRG		
P07	INTP6		
P10	SI0/SDA[Note]	10-A	
P11	SO0	26	
P12	$\overline{SCK0}$/SCL[Note]	10-A	
P13	SI1/RXD0	8-A	
P14	SO1/TXD0	26	
P15	$\overline{SCK1}$/ASCK0	10-A	
P20	SI2	8-A	
P21	SO2	26	
P22	$\overline{SCK2}$	10-A	
P23	RXD1	8-A	
P24	TXD1	5-A	
P25	ASCK1	8-A	
P26, P27	TI2/TO2, TI3/TO3		
P30, P31	TI00, TI01		
P32, P33	TI10, TI11		
P34, P35	TO0/A13, TO1/A14	5-A	
P36	TI4/TO4/A15	8-A	
P37	TI5/TO5		
P40 to P47	AD0 to AD7	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
P50 to P57	AD8 to AD15		
P60 to P65	A16 to A21		
P70 to P77	ANI0 to ANI7	9	Connect to AV_{DD} or AV_{SS}.
P80 to P83	ANI8 to ANI11		
P90	$\overline{LBEN}$/$\overline{WRL}$	5	Input: Independently connect to BV_{DD} or BV_{SS}. Output: Leave open.
P91	$\overline{UBEN}$		
P92	R/$\overline{W}$/$\overline{WRH}$		
P93	$\overline{DSTB}$/$\overline{RD}$		
P94	ASTB		
P95	$\overline{HLDAK}$		
P96	$\overline{HLDRQ}$		

(2/2)

Pin	Alternate Function	I/O Circuit Type	Recommended Connection Method
P100 to P107	RTP0/A5 to RTP7/A12	26	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P110 to P113	A1 to A4	5	
P114	XT1	16-A	Connect to V_{SS}.
P120	$\overline{WAIT}$	5	Input: Connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
AV_{REF}	—	—	Connect to AV_{SS}.
CLKOUT	—	4	Leave open.
IC[Note 1]	—	—	Directly connect to V_{SS}.
$\overline{RESET}$	—	2	—
V_{PP}[Note 2]	—	—	Connect to V_{SS}.
X2	—	—	Leave open (when external clock is input to X1 pin).
XT2	—	16-A	Leave open.

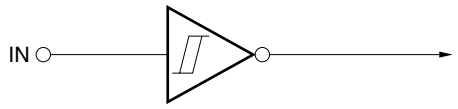
Notes 1. μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, and 703017AY only

2. μPD70F3015B, 70F3015BY, 70F3017A, and 70F3017AY only

2.5 Pin I/O Circuits

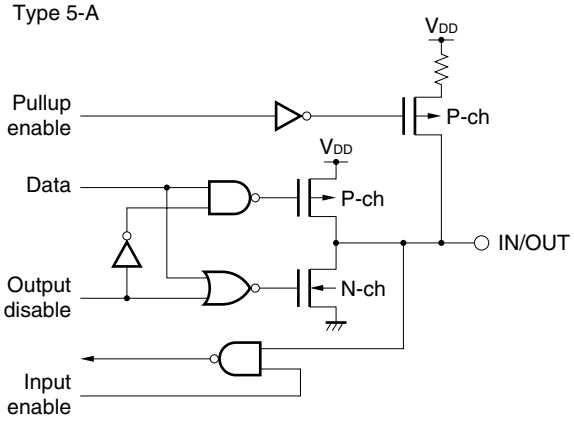
(1/2)

Type 2

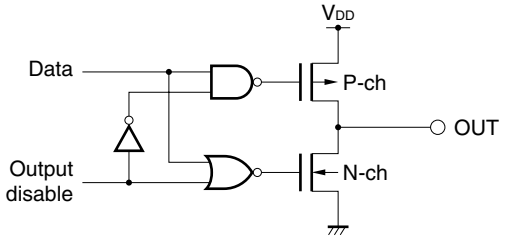


Schmitt-triggered input with hysteresis characteristics

Type 5-A

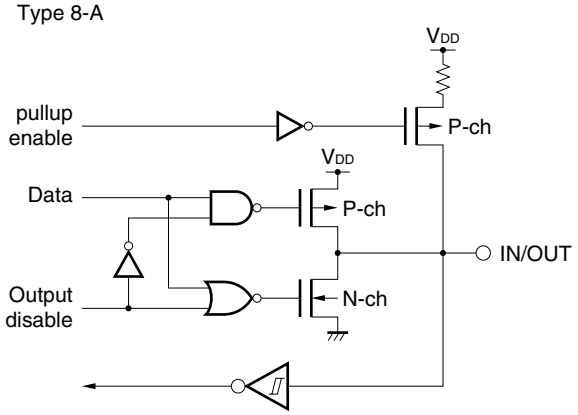


Type 4

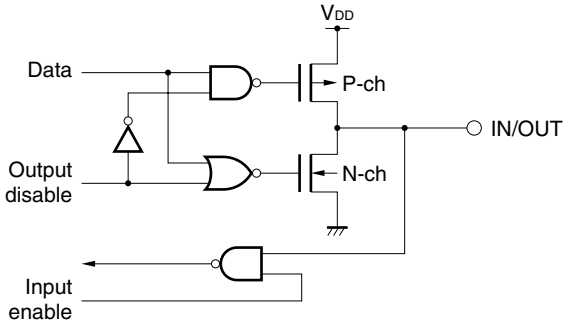


Push-pull output that can be set to high impedance output (both P-ch and N-ch off).

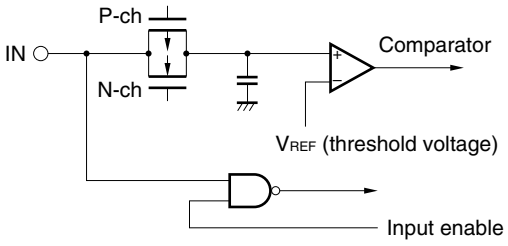
Type 8-A



Type 5

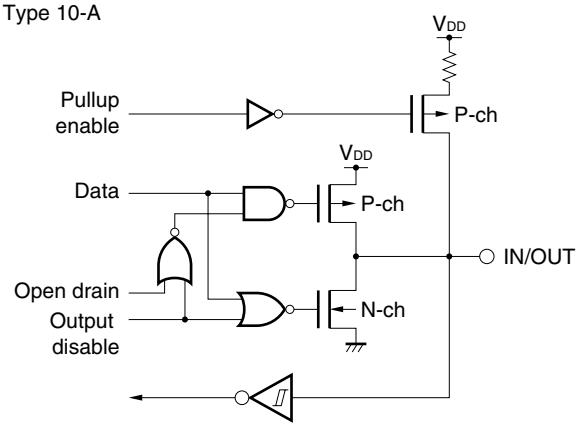


Type 9

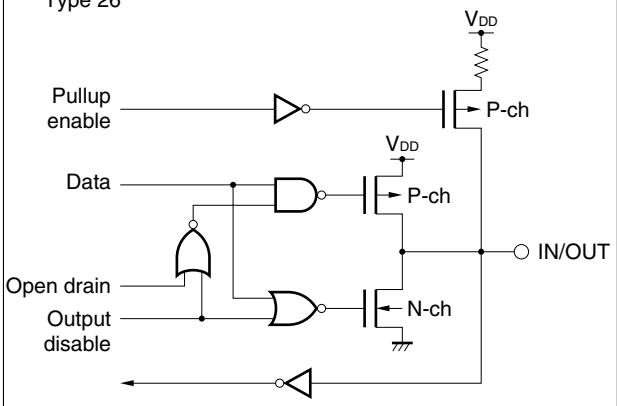


(2/2)

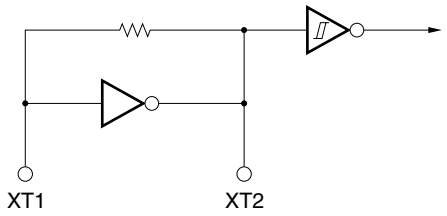
Type 10-A



Type 26



★ Type 16-A



CHAPTER 3 CPU FUNCTIONS

The CPU of the V850/SA1 is based on RISC architecture and executes most instructions in one clock cycle by using a 5-stage pipeline.

3.1 Features

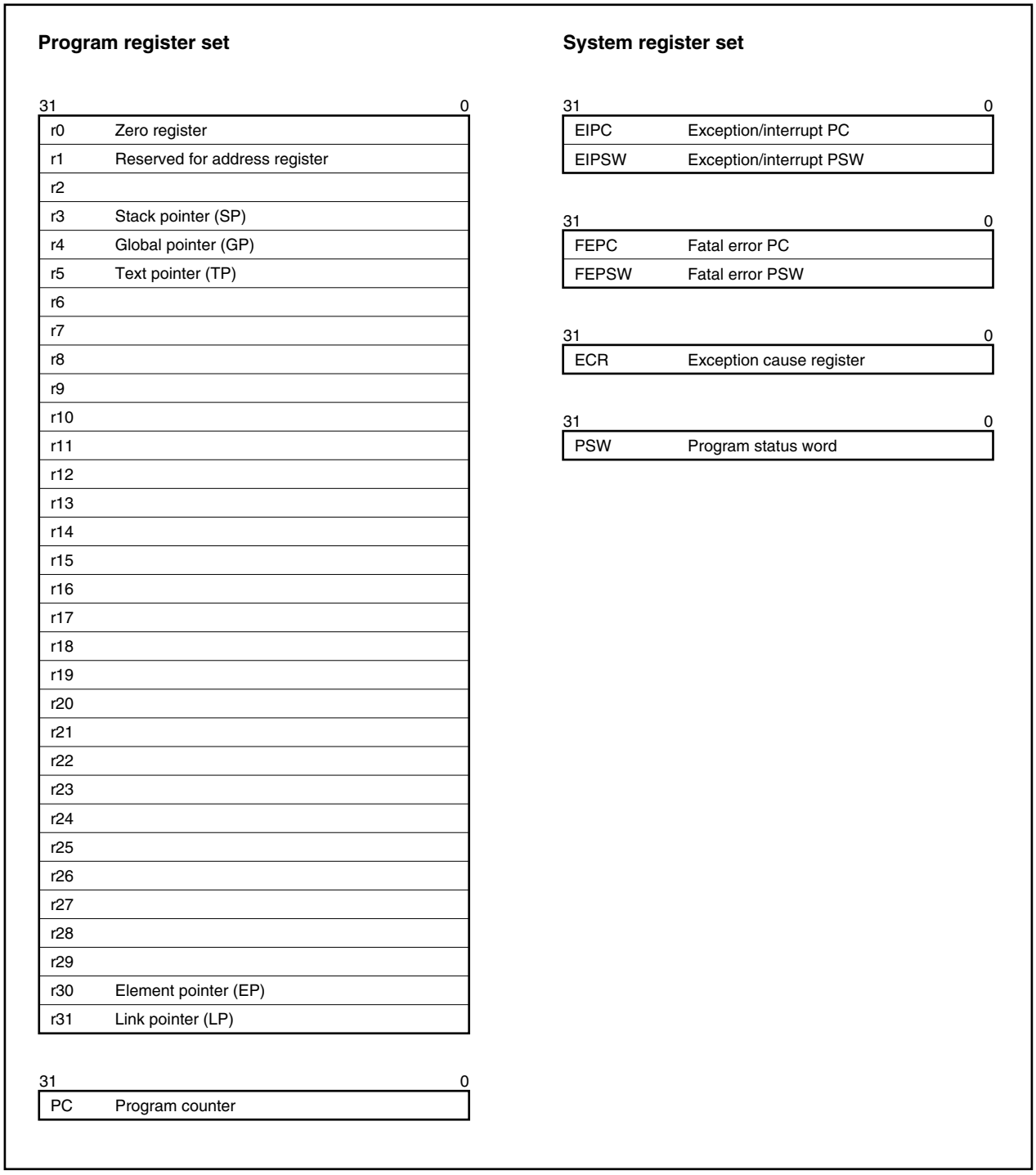
★

- Minimum instruction execution time: 50 ns (@ internal 20 MHz operation)
  58.8 ns (@ internal 17 MHz operation)
  30.5 μs (@ internal 32.768 kHz operation)
- Address space: 16 MB linear (Physical address space: 4 MB)
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiply/divide instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instruction with long/short format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The CPU registers of the V850/SA1 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32 bits width. For details, refer to **V850 Series Architecture User's Manual**.

Figure 3-1. CPU Register Set



3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

(1) General-purpose registers

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

However, r0 and r30 are implicitly used by instructions, so care must be exercised when using these registers. Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used. r2 is sometimes used by the real-time OS. When the real-time OS to be used is not using r2, r2 can be used as a variable register.

Table 3-1. Program Registers

Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate
r2	Address/data variable register (when the real-time OS to be used is not using r2)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area[Note]
r6 to r29	Address/data variable registers	
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function
PC	Program counter	Holds instruction address during program execution

Note Area in which program code is mapped.

(2) Program counter (PC)

This register holds the address of the instruction under execution. The lower 24 bits of this register are valid, and bits 31 to 24 are fixed to 0. If a carry occurs from bit 23 to bit 24, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.

After reset: 00000000H

Symbol	31 – 24	23 – 1	0
PC	Fixed to 0	Instruction address under execution	0

3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.

Table 3-2. System Register Numbers

No.	System Register Name	Usage	Operation
0	EIPC	Interrupt status saving registers	These registers save the PC and PSW when an exception or interrupt occurs. Because only one set of these registers is available, their contents must be saved when multiple interrupts are enabled.
1	EIPSW		
2	FEPC	NMI status saving registers	These registers save the PC and PSW when an NMI occurs.
3	FEPSW		
4	ECR	Interrupt source register	If an exception, maskable interrupt, or NMI occurs, this register will contain information referencing the interrupt source. The higher 16 bits of this register are called FECC, to which the exception code of the NMI is set. The lower 16 bits are called EICC, to which the exception code of the exception/interrupt is set.
5	PSW	Program status word	The program status word is a collection of flags that indicate the program status (instruction execution result) and CPU status.
6 to 31	Reserved		

To read/write these system registers, specify the system register number indicated by the system register load/store instruction (LDSR or STSR instruction).

(1) Interrupt source register (ECR)

After reset: 00000000H

Symbol	31 16	15 0
ECR	FECC	EICC

FECC	Exception code of NMI (For the exception code, refer to **Table 5-1**.)
EICC	Exception code of exception/interrupt

★ **(2) Program status word (PSW)**

(1/2)

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	RFU		NP	EP	ID	SAT	CY	OV	S	Z

RFU	Reserved field (fixed to 0).

NP	Non-maskable interrupt (NMI) servicing status
0	NMI servicing not under execution.
1	NMI servicing under execution. This flag is set (1) when an NMI is acknowledged, and disables multiple interrupts. For details, refer to **5.2.3 NP flag.**

EP	Exception processing status
0	Exception processing not under execution.
1	Exception processing under execution. This flag is set (1) when an exception is generated. Interrupt requests can be acknowledged when this bit is set. For details, refer to **5.4.3 EP flag.**

ID	Maskable interrupt servicing specification
0	Maskable interrupt acknowledgement enabled.
1	Maskable interrupt acknowledgement disabled. This flag is set (1) when a maskable interrupt request is acknowledged. For details, refer to **5.3.6 ID flag.**

(2/2)

SAT[Note]	Saturation detection of operation result of saturation operation instruction
0	Not saturated. This flag is not cleared (0) if the result of saturated operation instruction execution is not saturated while this flag is set (1). To clear (0) this flag, write the PSW directly.
1	Saturated.

CY	Detection of carry or borrow of operation result
0	Overflow has not occurred.
1	Overflow occurred.

OV[Note]	Detection of overflow during operation
0	Overflow has not occurred.
1	Overflow occurred.

S[Note]	Detection of operation result positive/negative
0	The operation result was positive or 0.
1	The operation result was negative.

Z	Detection of operation result zero
0	The operation result was not 0.
1	The operation result was 0.

Note The result of a saturation-processed operation is determined by the contents of the OV and S bits in the saturation operation. Simply setting (1) the OV bit will set (1) the SAT bit in a saturation operation.

Status of operation result	Flag status			Saturation-processed operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFFFFFFH
Maximum negative value exceeded	1	1	1	80000000H
Positive (not exceeding the maximum)	Retains the value before operation	0	0	Operation result itself
Negative (not exceeding the maximum)			1	

3.3 Operation Modes

The V850/SA1 has the following operation modes.

(1) Normal operation mode (single-chip mode)

After the system has been released from the reset status, the pins related to the bus interface are set to port mode, execution branches to the reset entry address of the internal ROM, and instruction processing written in the internal ROM is started. The external expansion mode in which an external device can be connected to external memory area is enabled by setting the memory expansion mode register (MM) by an instruction.

★ **(2) Flash memory programming mode**

This mode is provided only in the μPD70F3015B, 70F3015BY, 70F3017A, and 70F3017AY. The internal flash memory is programmable or erasable when the V_{PP} voltage is applied to the V_{PP} pin.

V_{PP}	Operation Mode
0	Normal operation mode
7.8 V	Flash memory programming mode
V_{DD}	Setting prohibited

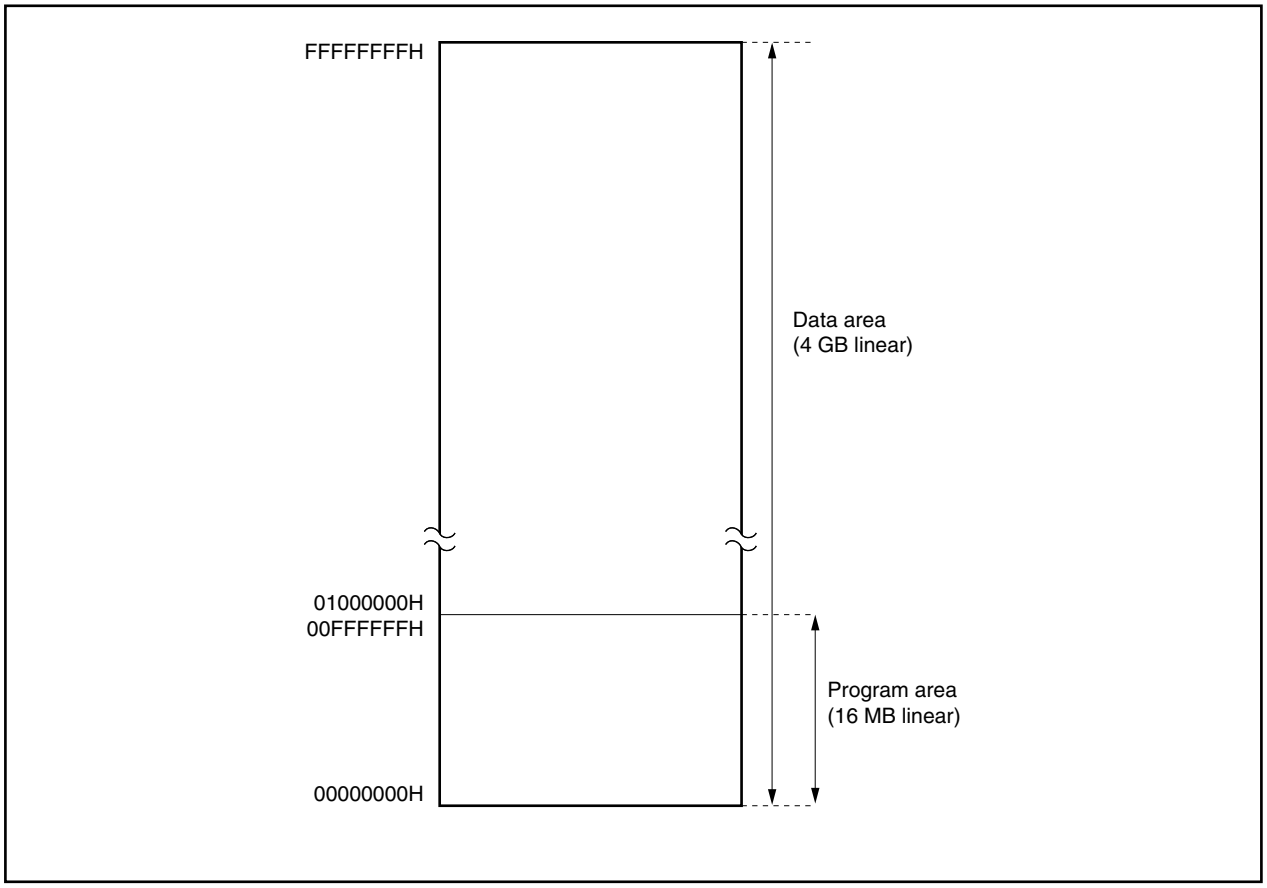
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850/SA1 is of 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). When referencing instruction addresses, linear address space (program space) of up to 16 MB is supported (physical address space: 4 MB).

The CPU address space is shown below.

Figure 3-2. CPU Address Space

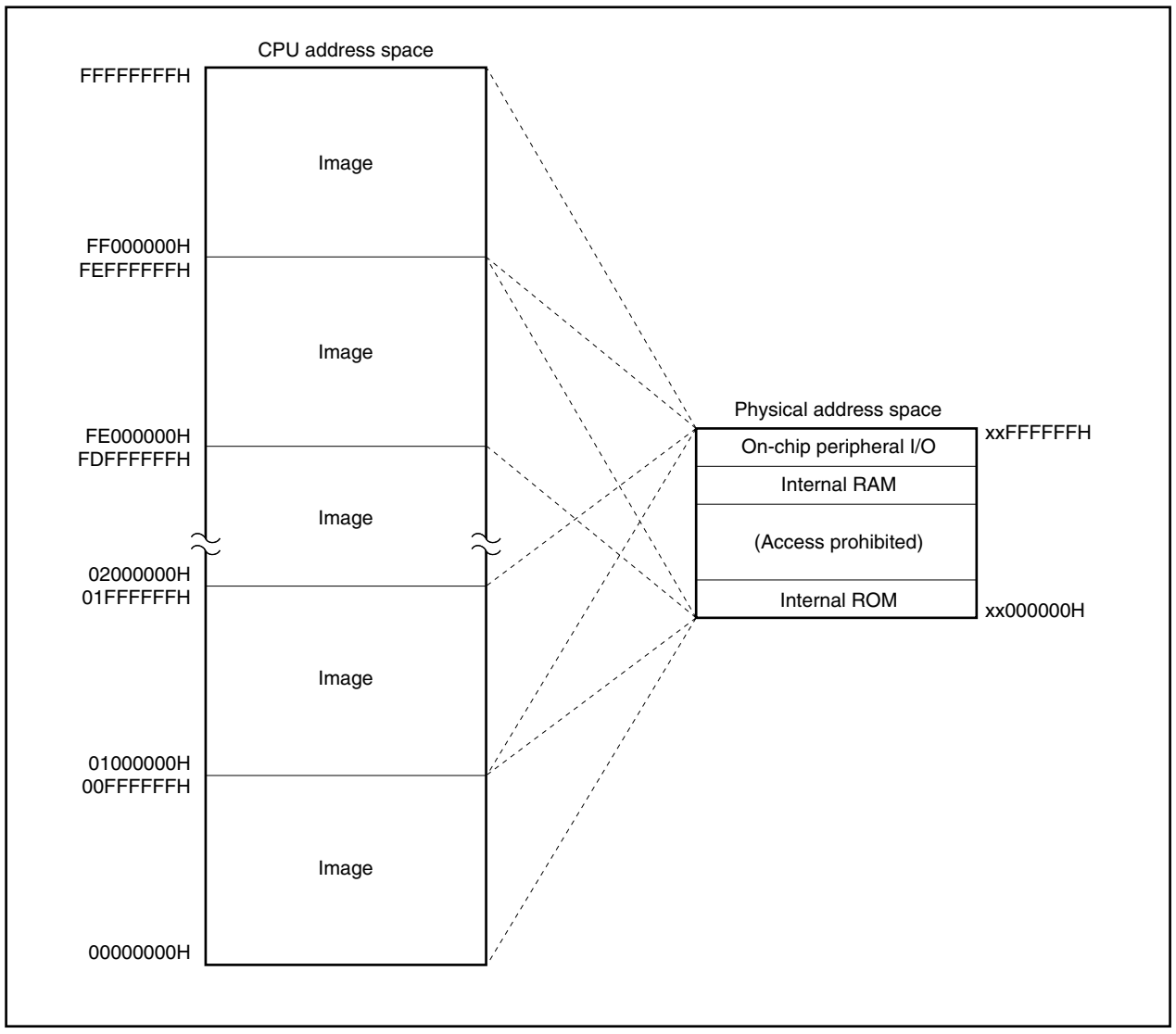


3.4.2 Image

The CPU supports 4 GB of “virtual” addressing space, or 256 memory blocks, each containing 16 MB memory locations. In actuality, the same 16 MB block is accessed regardless of the values of bits 31 to 24 of the CPU address. Figure 3-3 shows the image of the virtual addressing space.

Because the higher 8 bits of a 32-bit CPU address are ignored and the CPU address is only seen as a 24-bit external physical address, the physical location xx000000H is equally referenced by multiple address values 00000000H, 01000000H, 02000000H, ... FE000000H, FF000000H.

Figure 3-3. Image on Address Space



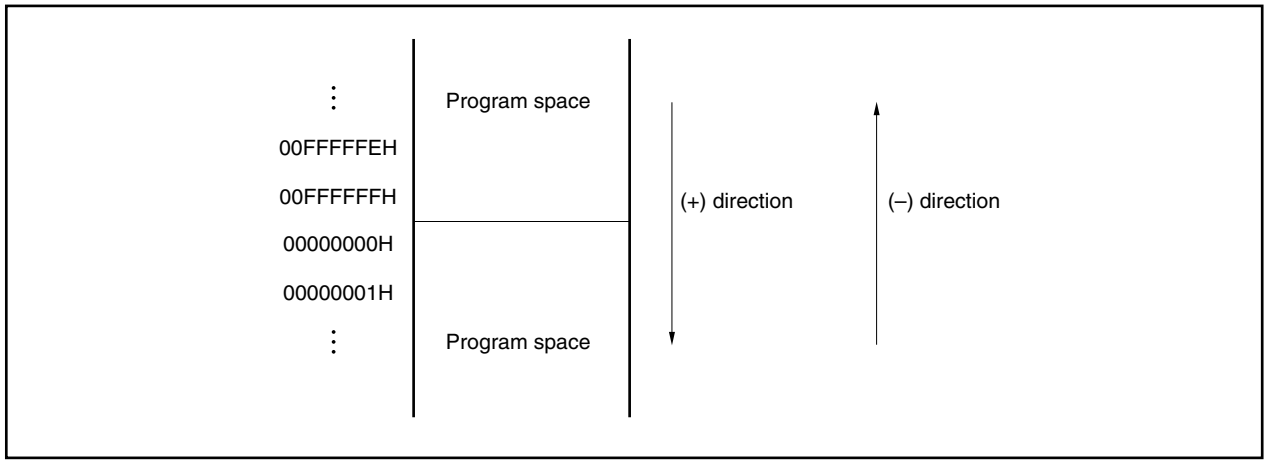
3.4.3 Wraparound of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 8 bits are fixed to 0, and only the lower 24 bits are valid. Even if a carry or borrow occurs from bit 23 to bit 24 as a result of branch address calculation, the higher 8 bits ignore the carry or borrow and remain 0. Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address 00FFFFFFH are contiguous addresses, and the program space is wrapped around at the boundary of these addresses.

Caution No instruction can be fetched from the 4 KB area of 00FFF000H to 00FFFFFFH because this area is defined as peripheral I/O area. Therefore, do not execute any branch operation instructions in which the destination address will reside in any part of this area.

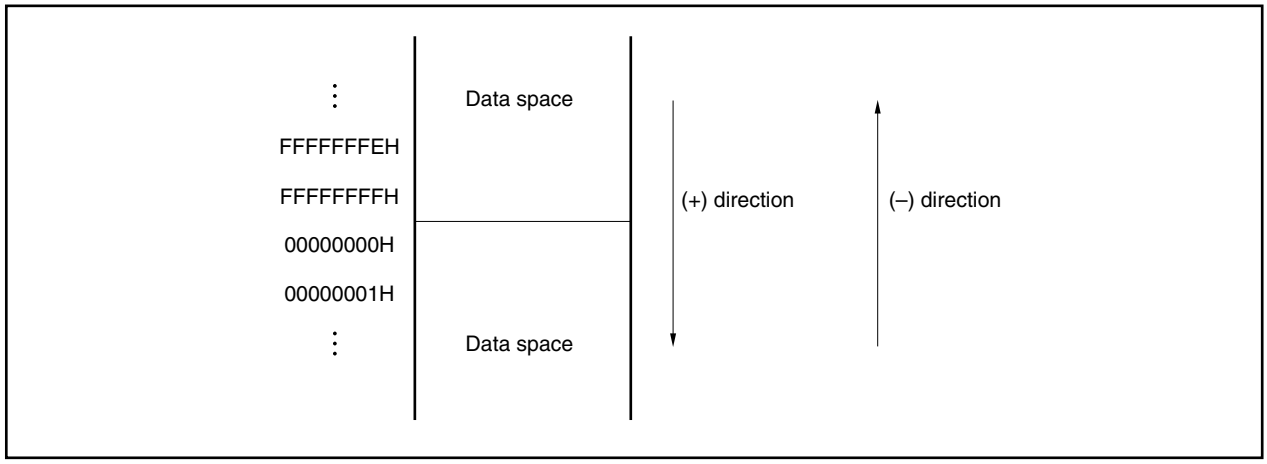
Figure 3-4. Program Space



(2) Data space

The result of operand address calculation that exceeds 32 bits is ignored. Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address FFFFFFFFH are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.

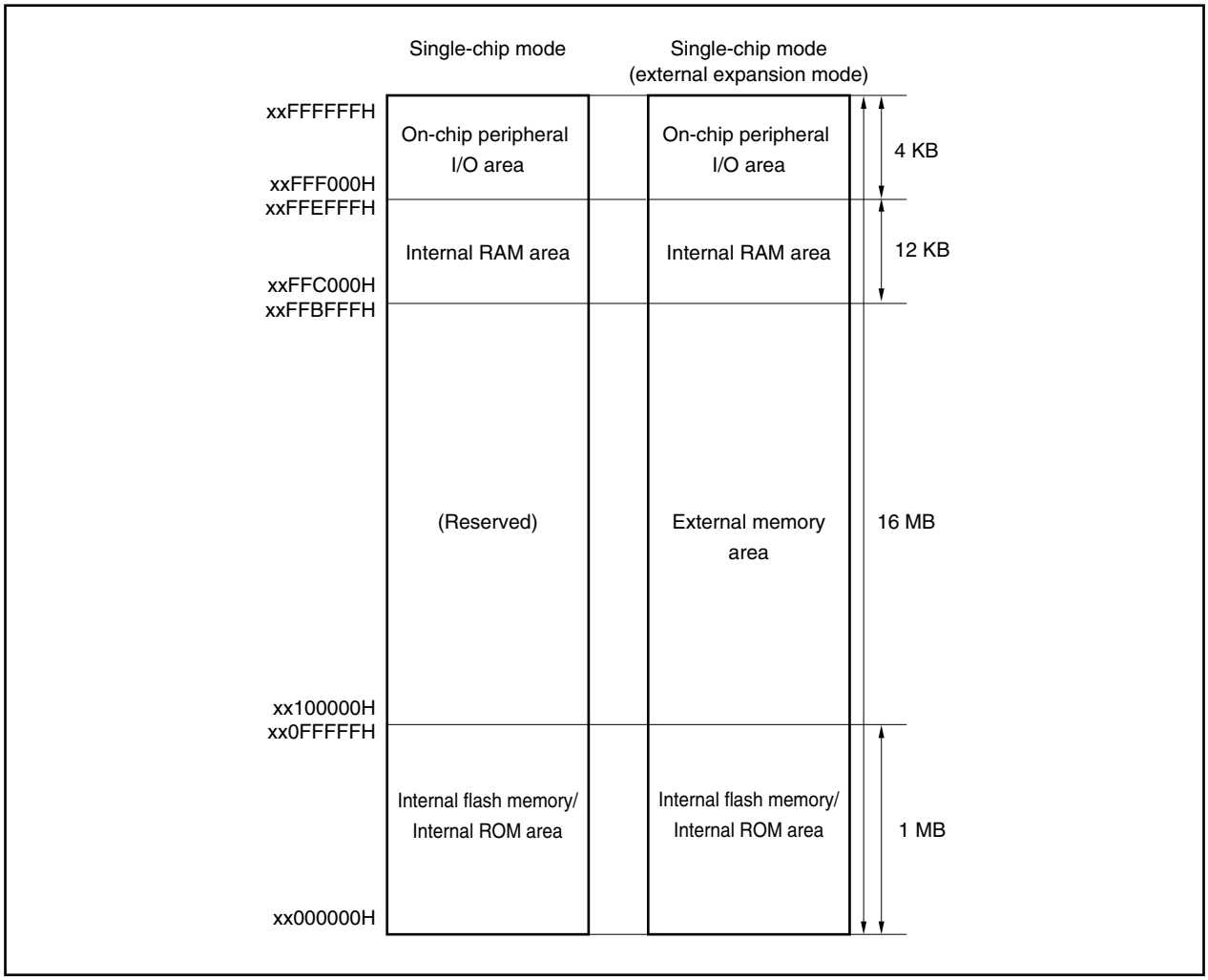
Figure 3-5. Data Space



3.4.4 Memory map

The V850/SA1 reserves areas as shown below.

Figure 3-6. Memory Map



3.4.5 Area

★ **(1) Internal ROM/internal flash memory area**

An area of 1 MB maximum is reserved for the internal ROM/internal flash memory area.

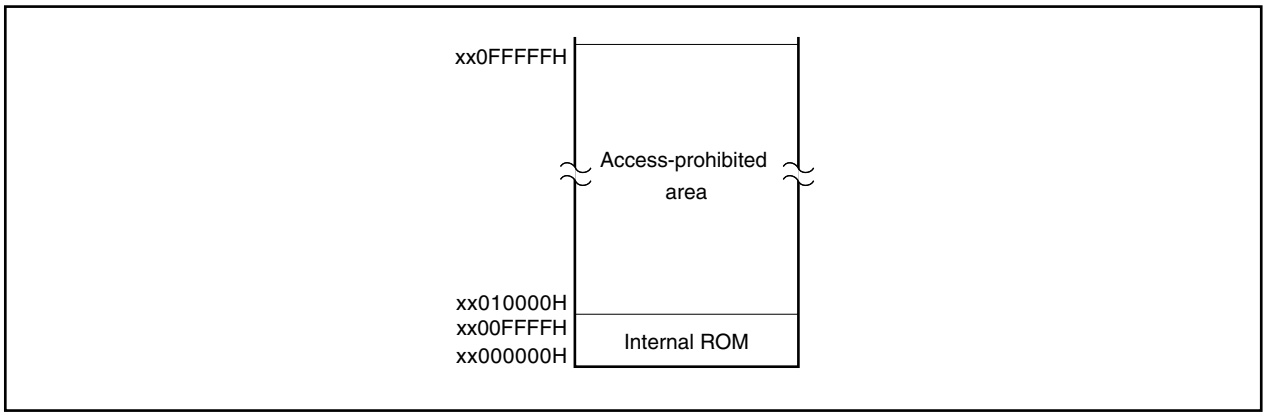
(a) Memory map

<1> μPD703014A, 703014AY, 703014B, 703014BY

64 KB is provided at addresses xx000000H to xx00FFFFH.

Addresses xx010000H to xx0FFFFFH are access-prohibited area.

Figure 3-7. Internal ROM Area (64 KB)

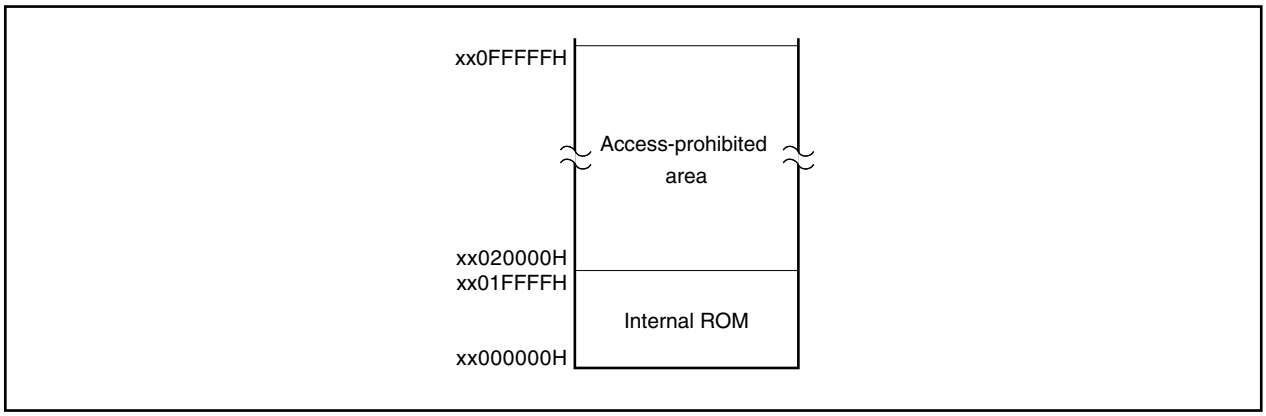


<2> μPD703015A, 703015AY, 703015B, 703015BY, 70F3015B, 70F3015BY

128 KB is provided at addresses xx000000H to xx01FFFFH.

Addresses xx020000H to xx0FFFFFH are access-prohibited area.

Figure 3-8. Internal ROM Area (128 KB)

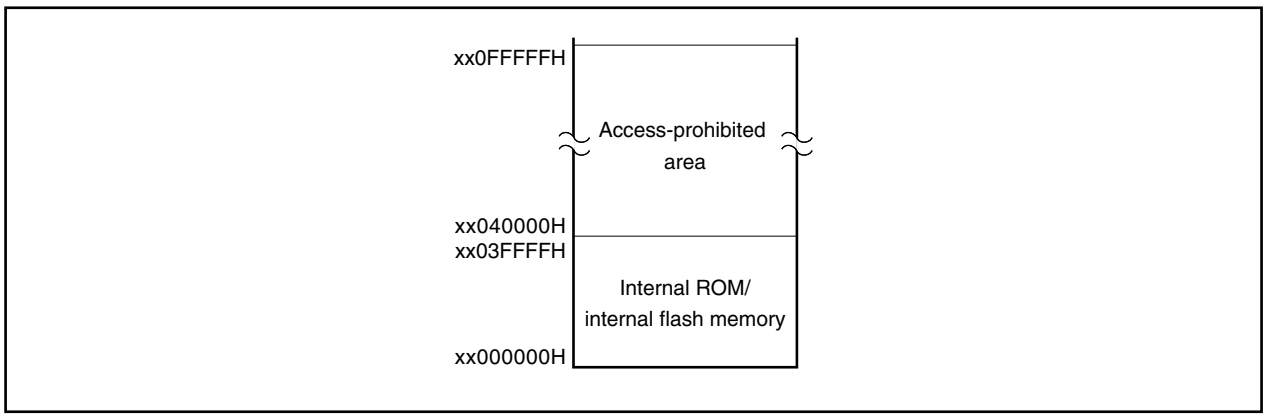


<3> μPD703017A, 703017AY, 70F3017A, 70F3017AY

256 KB is provided at addresses xx000000H to xx03FFFFH.

Addresses xx040000H to xx0FFFFFH are access-prohibited area.

Figure 3-9. Internal ROM/Internal Flash Memory Area (256 KB)



(b) Interrupt/exception table

The V850/SA1 increases the interrupt response speed by assigning handler addresses corresponding to interrupts/exceptions.

The collection of these handler addresses is called an interrupt/exception table, which is located in the internal ROM/on-chip flash memory area. When an interrupt/exception request is granted, execution jumps to the handler address, and the program written at that memory address is executed. The sources of interrupts/exceptions, and the corresponding addresses are shown below.

Table 3-3. Interrupt/Exception Table

Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000000H	RESET
00000010H	NMI
00000020H	INTWDT
00000040H	TRAP0n (n = 0 to F)
00000050H	TRAP1n (n = 0 to F)
00000060H	ILGOP
00000080H	INTWDTM
00000090H	INTP0
000000A0H	INTP1
000000B0H	INTP2
000000C0H	INTP3
000000D0H	INTP4
000000E0H	INTP5
000000F0H	INTP6
00000100H	INTWTI
00000110H	INTTM00
00000120H	INTTM01
00000130H	INTTM10
00000140H	INTTM11
00000150H	INTTM2
00000160H	INTTM3
00000170H	INTTM4
00000180H	INTTM5
00000190H	INTIIC0[Note]/INTCSI0
000001A0H	INTSER0
000001B0H	INTSR0/INTCSI1
000001C0H	INTST0
000001D0H	INTCSI2
000001E0H	INTSER1
000001F0H	INTSR1
00000200H	INTST1
00000210H	INTAD
00000220H	INTDMA0
00000230H	INTDMA1
00000240H	INTDMA2
00000250H	INTWT

★ **Note** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

★ **(2) Internal RAM area**

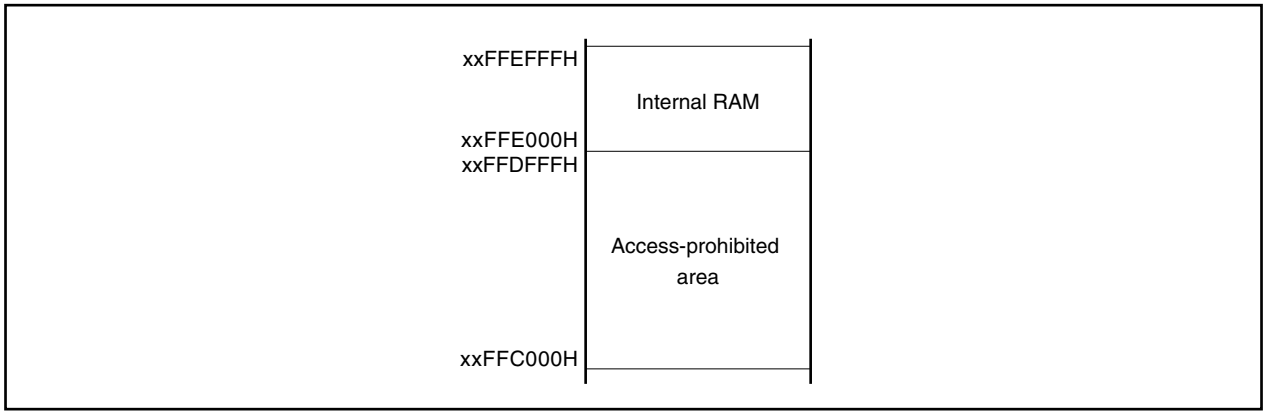
Up to 12 KB is reserved for the internal RAM area.

(a) μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 70F3015B, 70F3015BY

4 KB is provided at addresses xxFFE000H to xxFFEFFFH.

Addresses xxFFC000H to xxFFDFFFH are access-prohibited area.

Figure 3-10. Internal RAM Area (4 KB)

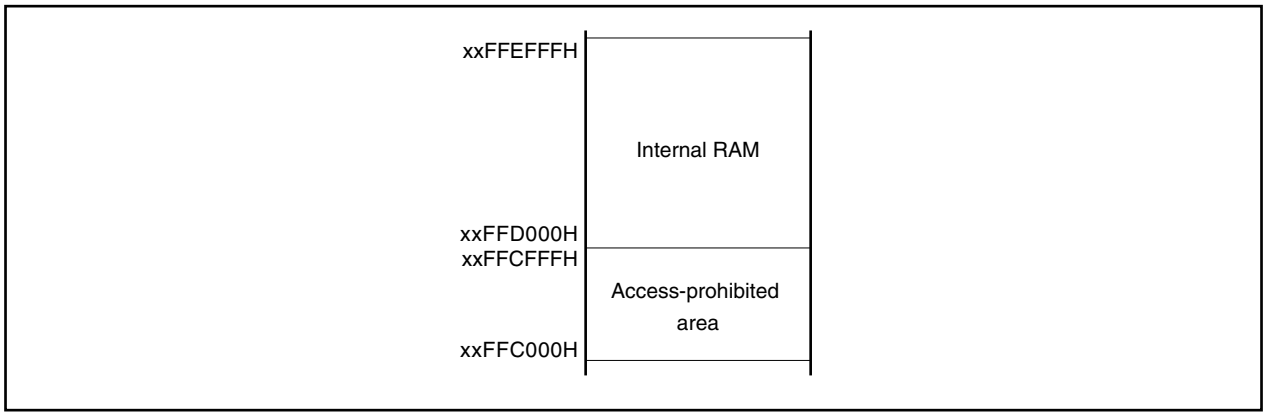


(b) μPD703017A, 703017AY, 70F3017A, 70F3017AY

8 KB is provided at addresses xxFFD000H to xxFFEFFFH.

Addresses xxFFC000H to xxFFCFFFH are access-prohibited area.

Figure 3-11. Internal RAM Area (8 KB)



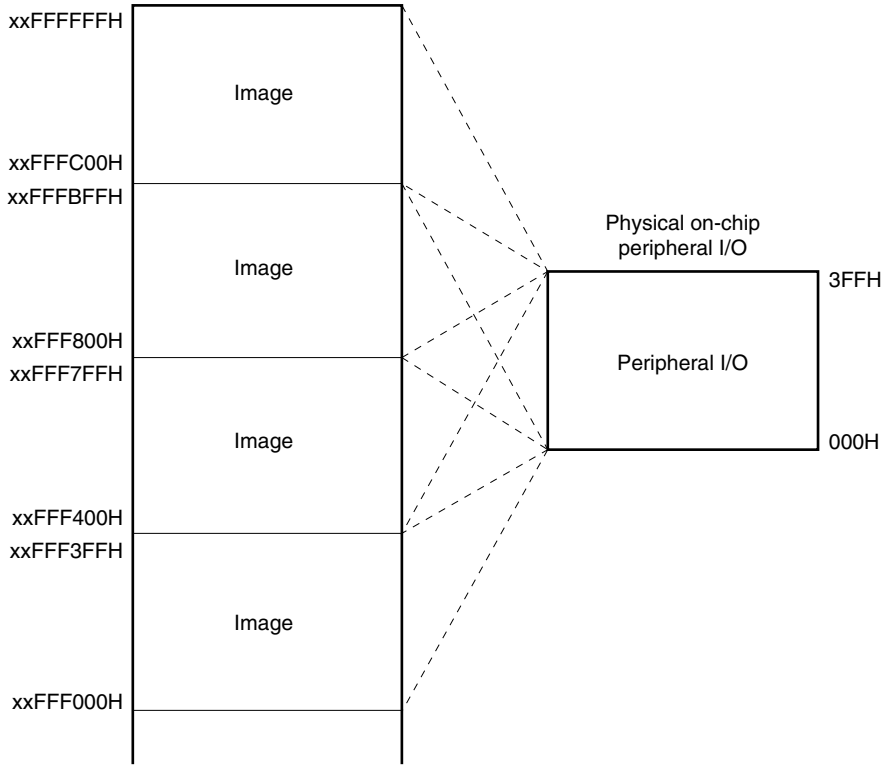
(3) On-chip peripheral I/O area

A 4 KB area of addresses FFF000H to FFFFFFH is reserved as an on-chip peripheral I/O area.

The V850/SA1 is provided with a 1 KB area of addresses FFF000H to FFF3FFH as a physical on-chip peripheral I/O area, and its image can be seen on the rest of the area (FFF400H to FFFFFFH).

Peripheral I/O registers associated with operation mode specification and state monitoring for the on-chip peripherals are all memory-mapped to the on-chip peripheral I/O area. Program fetches are not allowed in this area.

Figure 3-12. On-Chip Peripheral I/O Area



Cautions 1. The least significant bit of an address is not decoded since all registers reside at an even address. If an odd address (2n + 1) in the peripheral I/O area is referenced (accessed in byte units), the register at the next lowest even address (2n) will be accessed.

2. If a register that can be accessed in byte units is accessed in halfword units, the higher 8 bits become undefined, if the access is a read operation. If a write access is made, only the data in the lower 8 bits is written to the register.

3. If a register at address n that can be accessed only in halfword units is accessed in word units, the operation is replaced with two halfword operations. The first operation (lower 16 bits) accesses the register at address n and the second operation (higher 16 bits) accesses the register at address n + 2.

4. If a register at address n that can be accessed in word units is accessed with a word operation, the operation is replaced with two halfword operations. The first operation (lower 16 bits) accesses the register at address n and the second operation (higher 16 bits) accesses the register at address n + 2.

5. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

(4) External memory area

The V850/SA1 can use an area of up to 16 MB (xx100000H to xxFFBFFFH) for external memory accesses (in single-chip mode: during external expansion).

64 KB, 256 KB, 1 MB, or 4 MB of physical external memory can be allocated when the external expansion mode is specified. In the area of other than the physical external memory, the image of the physical external memory can be seen.

The internal RAM area and on-chip peripheral I/O area are not subject to external memory access.

Figure 3-13. External Memory Area (When Expanded to 64 KB, 256 KB, or 1 MB)

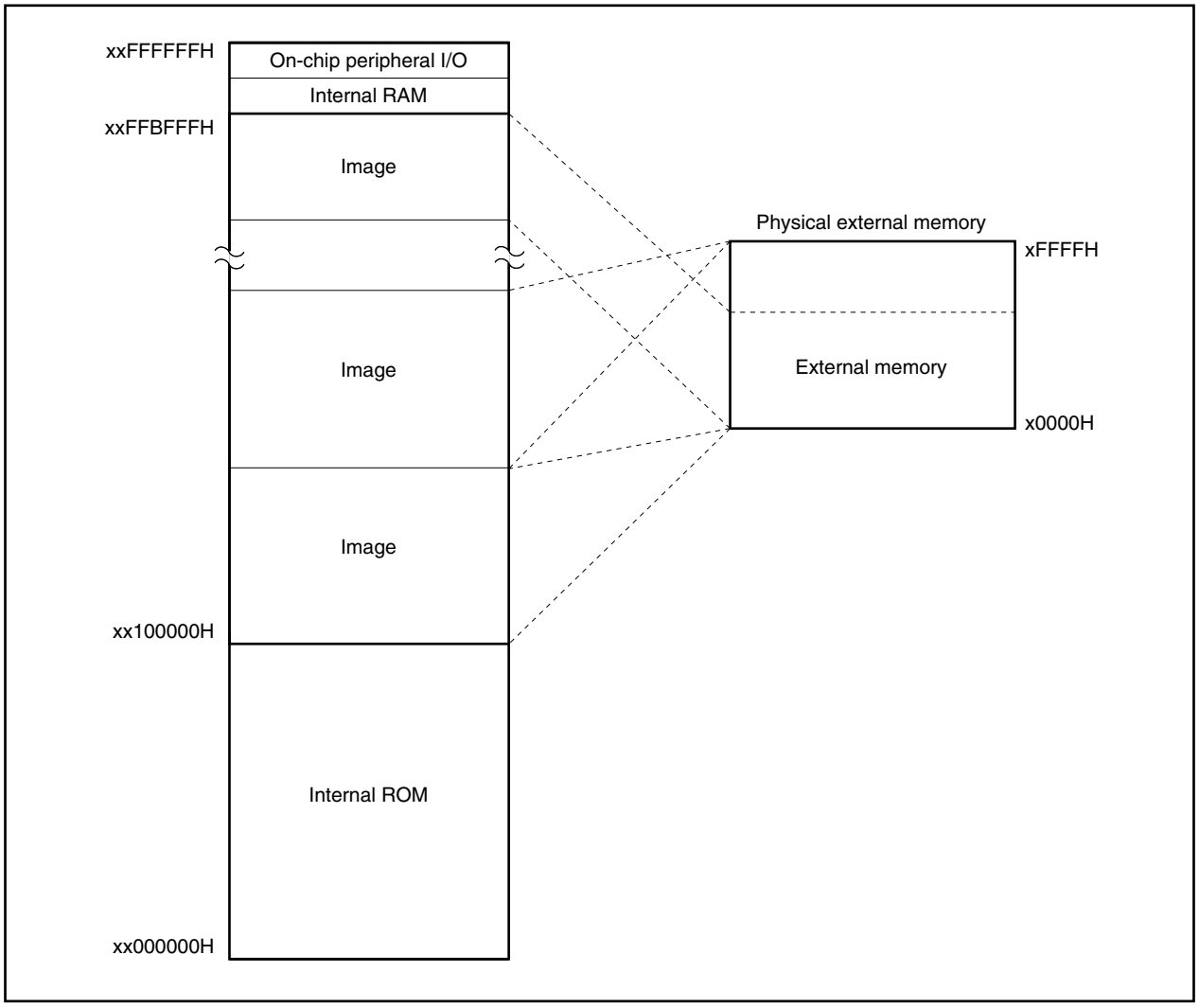
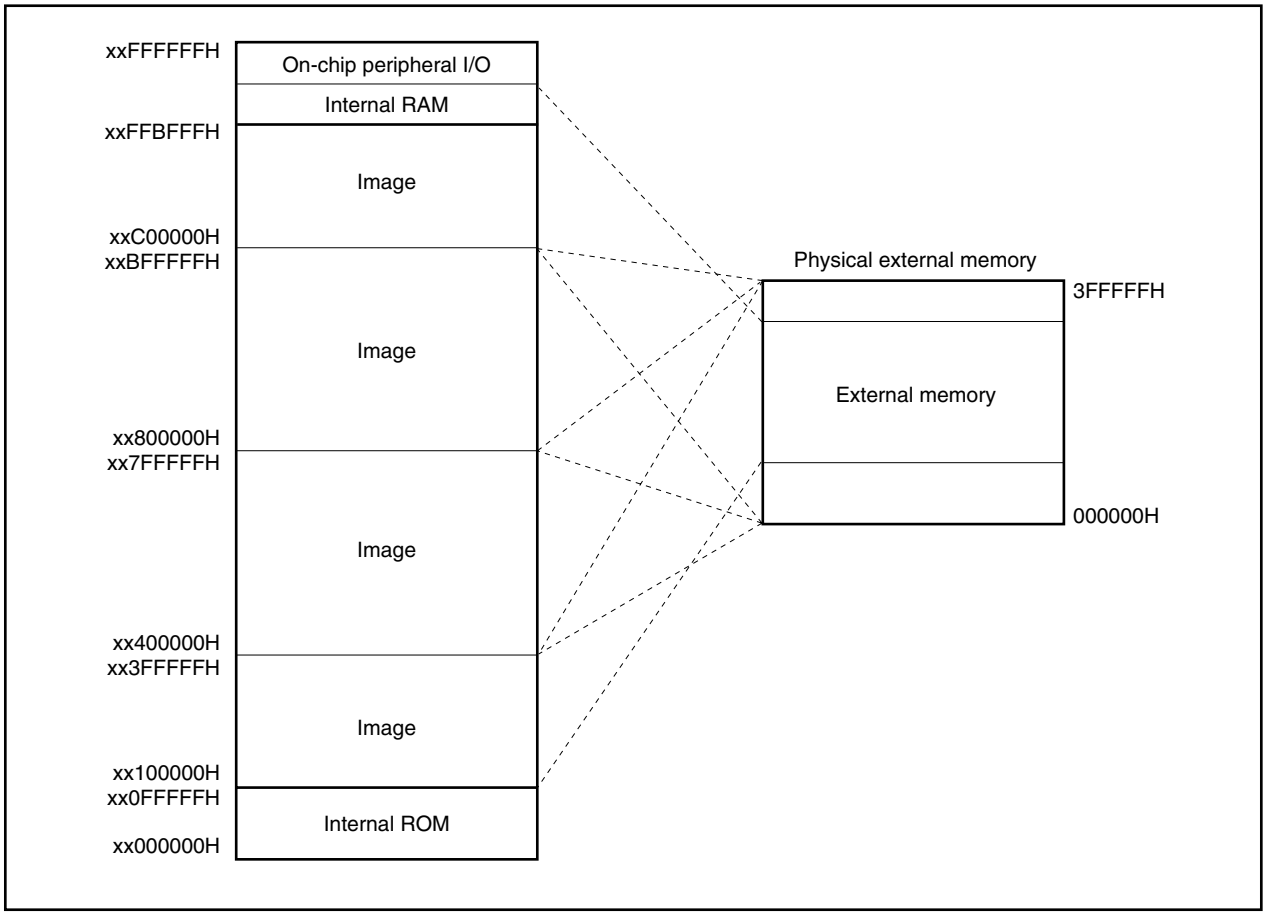


Figure 3-14. External Memory Area (When Expanded to 4 MB)



3.4.6 External expansion mode

The V850/SA1 allows external devices to be connected to the external memory space by using the pins of ports 4, 5, 6, and 9. To connect an external device, the port pins must be set to the external expansion mode by using the memory expansion mode register (MM).

The address bus (A1 to A15) is set to multiplexed bus output with the data bus (D1 to D15), though separate bus output is also possible by setting the memory address output mode register (MAM) (see **IE-703017-MC-EMI User's Manual** for debugging when using the separate bus).

Because the V850/SA1 is fixed to single-chip mode in the normal operation mode, the port/control mode alternate-function pins enter the port mode, and the external memory cannot be used. When the external memory is used (external expansion mode), specify the MM register or MAM register by the program (memory area is set by the MM register).

(1) Memory expansion mode register (MM)

This register sets the mode of each pin of ports 4, 5, 6, and 9. In the external expansion mode, an external device can be connected to an external memory area of up to 4 MB. However, an external device cannot be connected to the internal RAM area, on-chip peripheral I/O area, and internal ROM area in the single-chip mode (and even if the external device is connected physically, it cannot be accessed).

The MM register can be read/written in 8-bit or 1-bit units. However, bits 4 to 7 are fixed to 0.

After reset: 00H R/W Address: FFFFF04CH

Symbol	7	6	5	4	3	2	1	0
MM	0	0	0	0	MM3	MM2	MM1	MM0

MM3	P95 and P96 Operation Modes
0	Port mode
1	External expansion mode ($\overline{\text{HLDAK}}$: P95, $\overline{\text{HLDRQ}}$: P96)

MM2	MM1	MM0	Address Space	Port 4	Port 5	Port 6	Port 9
0	0	0	—	Port mode			
0	1	1	64 KB expansion mode	AD0 to AD7	AD8 to AD15		$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$, R/$\overline{\text{W}}$, $\overline{\text{DSTB}}$, ASTB, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$
1	0	0	256 KB expansion mode			A16, A17	
1	0	1	1 MB expansion mode			A18, A19	
1	1	×	4 MB expansion mode			A20, A21	
Other than above				RFU (reserved)			

Caution When switching to the external expansion mode, the P93 and P94 bits of port 9 (P9) must be set to 1 before switching.

Remark For details of the operation of each port pin, refer to **2.3 Description of Pin Functions**.

(2) Memory address output mode register (MAM)

This register sets the mode of each pin of ports 3, 10, and 11. Separate output can be set for the address bus (A1 to A15) in the external expansion mode. Separate bus output is output to P34 to P36, P100 to P107, and P110 to P113.

Set the separate bus output according to the following procedure.

(i) Set to output mode (PMn bit = 0) after setting port m, which will be used for separate output, to 0 output (Pn bit = 0).
(ii) Turn this function off if the ports to be used as the separate bus are being used as alternate-function pins other than those of the separate bus.
(iii) Set the memory address output register (MAM).
(iv) Set the memory expansion mode register (MM) (refer to **3.4.6 (1) Memory expansion mode register (MM)**).

The MAM register can be written in 8-bit units. If read, undefined values will be read. However, bits 3 to 7 are fixed to 0.

Remark When m = 3: n = 34 to 36
When m = 10: n = 100 to 107
When m = 11: n = 110 to 113

After reset: 00H W Address: FFFFF068H

Symbol	7	6	5	4	3	2	1	0
MAM	0	0	0	0	0	MAM2	MAM1	MAM0

MAM2	MAM1	MAM0	Address Space	Port 11	Port 10	Port 3
0	0	0	–	Port mode		
0	1	0	32 bytes	A1 to A4		
0	1	1	512 bytes		A5 to A8	
1	0	0	8 KB			A9 to A12
1	0	1	16 KB			A13
1	1	0	32 KB			A14
1	1	1	64 KB			A15

Caution An in-circuit emulator cannot be used to debug the memory address output mode register (MAM). Also, the separate bus cannot be switched to by setting the MAM register by software. For details, refer to IE-703017-MC-EMI User's Manual.

Remark For details of the operation of each port pin, refer to **2.3 Description of Pin Functions**.

3.4.7 Recommended use of address space

The architecture of the V850/SA1 requires that a register that serves as a pointer be secured for address generation in operand data accessing of the data space. The address in this pointer register ±32 KB can be accessed directly from an instruction. However, the general-purpose registers that can be used as a pointer register are limited. Therefore, by minimizing deterioration of the address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be minimized because instructions for calculating pointer addresses are not required.

To enhance the efficiency of using the pointer in connection with the memory map of the V850/SA1, the following points are recommended.

(1) Program space

Of the 32 bits of the PC (program counter), the higher 8 bits are fixed to 0, and only the lower 24 bits are valid. Therefore, a continuous 16 MB space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

(2) Data space

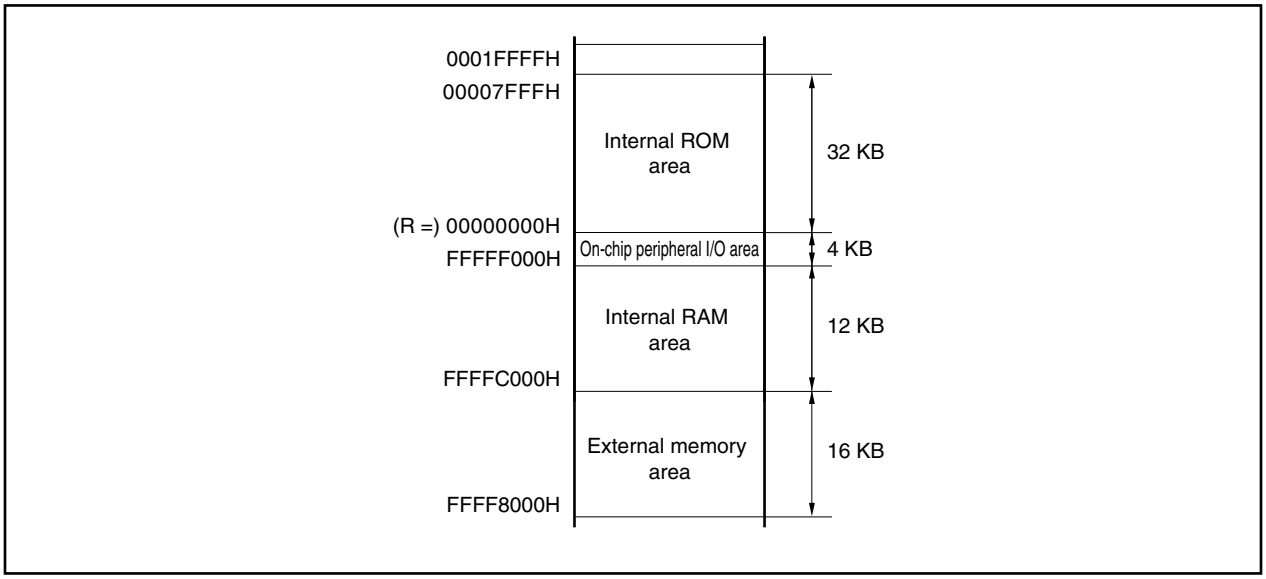
For the efficient use of resources to be performed using the wraparound feature of the data space, the continuous 8 MB address spaces 00000000H to 007FFFFFH and FF800000H to FFFFFFFFH of the 4 GB CPU are used as the data space. With the V850/SA1, the 16 MB physical address space is seen as 256 images in the 4 GB CPU address space. The highest bit (bit 23) of this 24-bit address is assigned as an address sign-extended to 32 bits.

Application example of wraparound

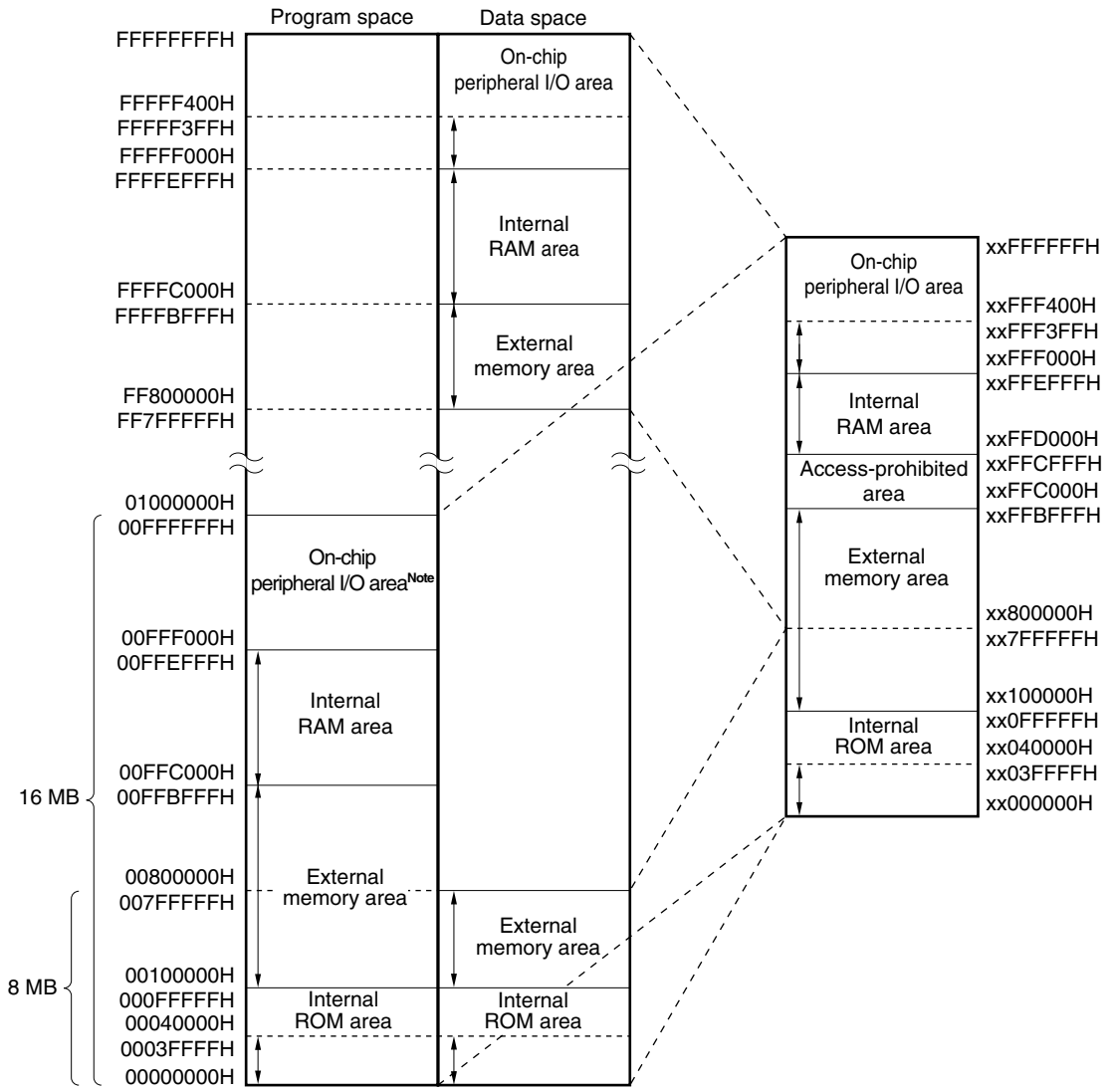
For example, when R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, an addressing range of 00000000H ±32 KB can be referenced with the sign-extended 16-bit displacement value. By mapping the external memory in the 16 KB area in Figure 3-15, all resources including on-chip hardware can be accessed with one pointer.

The zero register (r0) is a register set to 0 by the hardware, and eliminates the need for additional registers for the pointer.

Figure 3-15. Application Example of Wraparound



★ **Figure 3-16. Recommended Memory Map**



Note This area cannot be used as a program area.

Remarks 1. The arrows indicate the recommended area.

2. This is the recommended memory map for the μPD703017A and 703017AY.

3.4.8 Peripheral I/O registers

(1/5)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF000H	Port 0	P0	R/W	√	√		00H[Note]
FFFFF002H	Port 1	P1	R/W	√	√		00H[Note]
FFFFF004H	Port 2	P2	R/W	√	√		00H[Note]
FFFFF006H	Port 3	P3	R/W	√	√		00H[Note]
FFFFF008H	Port 4	P4	R/W	√	√		00H[Note]
FFFFF00AH	Port 5	P5	R/W	√	√		00H[Note]
FFFFF00CH	Port 6	P6	R/W	√	√		00H[Note]
FFFFF00EH	Port 7	P7	R	√	√		Undefined
FFFFF010H	Port 8	P8	R	√	√		Undefined
FFFFF012H	Port 9	P9	R/W	√	√		00H[Note]
FFFFF014H	Port 10	P10	R/W	√	√		00H[Note]
FFFFF016H	Port 11	P11	R/W	√	√		00H[Note]
FFFFF018H	Port 12	P12	R/W	√	√		00H[Note]
FFFFF020H	Port 0 mode register	PM0	R/W	√	√		FFH
FFFFF022H	Port 1 mode register	PM1	R/W	√	√		3FH
FFFFF024H	Port 2 mode register	PM2	R/W	√	√		FFH
FFFFF026H	Port 3 mode register	PM3	R/W	√	√		FFH
FFFFF028H	Port 4 mode register	PM4	R/W	√	√		FFH
FFFFF02AH	Port 5 mode register	PM5	R/W	√	√		FFH
FFFFF02CH	Port 6 mode register	PM6	R/W	√	√		3FH
FFFFF032H	Port 9 mode register	PM9	R/W	√	√		7FH
FFFFF034H	Port 10 mode register	PM10	R/W	√	√		FFH
FFFFF036H	Port 11 mode register	PM11	R/W	√	√		1FH
FFFFF038H	Port 12 mode register	PM12	R/W	√	√		01H
FFFFF04CH	Memory expansion mode register	MM	R/W	√	√		00H
FFFFF058H	Port 12 mode control register	PMC12	R/W	√	√		00H
FFFFF060H	Data wait control register	DWC	R/W			√	FFFFH
FFFFF062H	Bus cycle control register	BCC	R/W			√	AAAAH
FFFFF064H	System control register	SYC	R/W	√	√		00H
FFFFF068H	Memory address output mode register	MAM	W		√		00H
FFFFF070H	Power save control register	PSC	R/W	√	√		C0H
FFFFF074H	Processor clock control register	PCC	R/W	√	√		03H
FFFFF078H	System status register	SYS	R/W	√	√		00H
FFFFF080H	Pull-up resistor option register 0	PU0	R/W	√	√		00H
FFFFF082H	Pull-up resistor option register 1	PU1	R/W	√	√		00H

Note Resetting initializes registers to input mode and the pin level is read. Output latches are initialized to 00H.

(2/5)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF084H	Pull-up resistor option register 2	PU2	R/W	√	√		00H
FFFFF086H	Pull-up resistor option register 3	PU3	R/W	√	√		00H
FFFFF094H	Pull-up resistor option register 10	PU10	R/W	√	√		00H
FFFFF096H	Pull-up resistor option register 11	PU11	R/W	√	√		00H
FFFFF0A2H	Port 1 function register	PF1	R/W	√	√		00H
FFFFF0A4H	Port 2 function register	PF2	R/W	√	√		00H
FFFFF0B4H	Port 10 function register	PF10	R/W	√	√		00H
FFFFF0C0H	Rising edge specification register 0	EGP0	R/W	√	√		00H
FFFFF0C2H	Falling edge specification register 0	EGN0	R/W	√	√		00H
FFFFF100H	Interrupt control register	WDTIC	R/W	√	√		47H
FFFFF102H	Interrupt control register	PIC0	R/W	√	√		47H
FFFFF104H	Interrupt control register	PIC1	R/W	√	√		47H
FFFFF106H	Interrupt control register	PIC2	R/W	√	√		47H
FFFFF108H	Interrupt control register	PIC3	R/W	√	√		47H
FFFFF10AH	Interrupt control register	PIC4	R/W	√	√		47H
FFFFF10CH	Interrupt control register	PIC5	R/W	√	√		47H
FFFFF10EH	Interrupt control register	PIC6	R/W	√	√		47H
FFFFF110H	Interrupt control register	WTIIC	R/W	√	√		47H
FFFFF112H	Interrupt control register	TMIC00	R/W	√	√		47H
FFFFF114H	Interrupt control register	TMIC01	R/W	√	√		47H
FFFFF116H	Interrupt control register	TMIC10	R/W	√	√		47H
FFFFF118H	Interrupt control register	TMIC11	R/W	√	√		47H
FFFFF11AH	Interrupt control register	TMIC2	R/W	√	√		47H
FFFFF11CH	Interrupt control register	TMIC3	R/W	√	√		47H
FFFFF11EH	Interrupt control register	TMIC4	R/W	√	√		47H
FFFFF120H	Interrupt control register	TMIC5	R/W	√	√		47H
FFFFF122H	Interrupt control register	CSIC0	R/W	√	√		47H
FFFFF124H	Interrupt control register	SERIC0	R/W	√	√		47H
FFFFF126H	Interrupt control register	CSIC1	R/W	√	√		47H
FFFFF128H	Interrupt control register	STIC0	R/W	√	√		47H
FFFFF12AH	Interrupt control register	CSIC2	R/W	√	√		47H
FFFFF12CH	Interrupt control register	SERIC1	R/W	√	√		47H
FFFFF12EH	Interrupt control register	SRIC1	R/W	√	√		47H
FFFFF130H	Interrupt control register	STIC1	R/W	√	√		47H
FFFFF132H	Interrupt control register	ADIC	R/W	√	√		47H
FFFFF134H	Interrupt control register	DMAIC0	R/W	√	√		47H

(3/5)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF136H	Interrupt control register	DMAIC1	R/W	√	√		47H
FFFFF138H	Interrupt control register	DMAIC2	R/W	√	√		47H
FFFFF13AH	Interrupt control register	WTIC	R/W	√	√		47H
FFFFF166H	In-service priority register	ISPR	R	√	√		00H
FFFFF170H	Command register	PRCMD	W		√		Undefined
FFFFF180H	DMA peripheral I/O address register 0	DIOA0	R/W			√	Undefined
FFFFF182H	DMA internal RAM address register 0	DRA0	R/W			√	Undefined
FFFFF184H	DMA byte count register 0	DBC0	R/W		√		Undefined
FFFFF186H	DMA channel control register 0	DCHC0	R/W	√	√		00H
FFFFF190H	DMA peripheral I/O address register 1	DIOA1	R/W			√	Undefined
FFFFF192H	DMA internal RAM address register 1	DRA1	R/W			√	Undefined
FFFFF194H	DMA byte count register 1	DBC1	R/W		√		Undefined
FFFFF196H	DMA channel control register 1	DCHC1	R/W	√	√		00H
FFFFF1A0H	DMA peripheral I/O address register 2	DIOA2	R/W			√	Undefined
FFFFF1A2H	DMA internal RAM address register 2	DRA2	R/W			√	Undefined
FFFFF1A4H	DMA byte count register 2	DBC2	R/W		√		Undefined
FFFFF1A6H	DMA channel control register 2	DCHC2	R/W	√	√		00H
★ FFFFF1F4H	Flash programming mode control register[Note 1]	FLPMC	R/W	√	√		**Note 2**
FFFFF200H	16-bit timer register 0	TM0	R			√	0000H
FFFFF202H	16-bit capture/compare register 00	CR00	**Note 3**			√	0000H
FFFFF204H	16-bit capture/compare register 01	CR01	**Note 3**			√	0000H
FFFFF206H	Prescaler mode register 0	PRM0	R/W		√		00H
FFFFF208H	16-bit timer mode control register 0	TMC0	R/W	√	√		00H
FFFFF20AH	Capture/compare control register 0	CRC0	R/W	√	√		00H
FFFFF20CH	Timer output control register 0	TOC0	R/W	√	√		00H
FFFFF20EH	Prescaler mode register 01	PRM01	R/W		√		00H
FFFFF210H	16-bit timer register 1	TM1	R			√	0000H
FFFFF212H	16-bit capture/compare register 10	CR10	**Note 3**			√	0000H
FFFFF214H	16-bit capture/compare register 11	CR11	**Note 3**			√	0000H
FFFFF216H	Prescaler mode register 1	PRM1	R/W		√		00H
FFFFF218H	16-bit timer mode control register 1	TMC1	R/W	√	√		00H
FFFFF21AH	Capture/compare control register 1	CRC1	R/W	√	√		00H
FFFFF21CH	Timer output control register 1	TOC1	R/W	√	√		00H

★ **Notes 1.** Valid only for the μPD70F3015B, 70F3015BY, 70F3017A, and 70F3017AY.

2. In single-chip mode: 18H or 38H
In flash memory programming mode: 1CH or 3CH

3. In compare mode: R/W
In capture mode: R

(4/5)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF21EH	Prescaler mode register 11	PRM11	R/W		√		00H
FFFFF240H	8-bit counter 2	TM2	R		√		00H
FFFFF242H	8-bit compare register 2	CR20	R/W		√		00H
FFFFF244H	Timer clock select register 2	TCL2	R/W		√		00H
FFFFF246H	8-bit timer mode control register 2	TMC2	R/W	√	√		04H[Note]
FFFFF24AH	16-bit counter 23 (only when connected in cascade)	TM23	R			√	0000H
FFFFF24CH	16-bit compare register 23 (only when connected in cascade)	CR23	R/W			√	0000H
FFFFF24EH	Timer clock select register 21	TCL21	R/W		√		00H
FFFFF250H	8-bit counter 3	TM3	R		√		00H
FFFFF252H	8-bit compare register 3	CR30	R/W		√		00H
FFFFF254H	Timer clock select register 3	TCL3	R/W		√		00H
FFFFF256H	8-bit timer mode control register 3	TMC3	R/W	√	√		04H[Note]
FFFFF25EH	Timer clock select register 31	TCL31	R/W		√		00H
FFFFF260H	8-bit counter 4	TM4	R		√		00H
FFFFF262H	8-bit compare register 4	CR40	R/W		√		00H
FFFFF264H	Timer clock select register 4	TCL4	R/W		√		00H
FFFFF266H	8-bit timer mode control register 4	TMC4	R/W	√	√		04H[Note]
FFFFF26AH	16-bit counter 45 (only when connected in cascade)	TM45	R			√	0000H
FFFFF26CH	16-bit compare register 45 (only when connected in cascade)	CR45	R/W			√	0000H
FFFFF26EH	Timer clock select register 41	TCL41	R/W		√		00H
FFFFF270H	8-bit counter 5	TM5	R		√		00H
FFFFF272H	8-bit compare register 5	CR50	R/W		√		00H
FFFFF274H	Timer clock select register 5	TCL5	R/W		√		00H
FFFFF276H	8-bit timer mode control register 5	TMC5	R/W	√	√		04H[Note]
FFFFF27EH	Timer clock select register 51	TCL51	R/W		√		00H
FFFFF2A0H	Serial I/O shift register 0	SIO0	R/W		√		00H
FFFFF2A2H	Serial operation mode register 0	CSIM0	R/W	√	√		00H
FFFFF2A4H	Serial clock select register 0	CSIS0	R/W	√	√		00H
FFFFF2B0H	Serial I/O shift register 1	SIO1	R/W		√		00H
FFFFF2B2H	Serial operation mode register 1	CSIM1	R/W	√	√		00H
FFFFF2B4H	Serial clock select register 1	CSIS1	R/W	√	√		00H
FFFFF2C0H	Serial I/O shift register 2	SIO2	R/W		√		00H
FFFFF2C2H	Serial operation mode register 2	CSIM2	R/W	√	√		00H

Note Although the hardware status is initialized to 04H, 00H will be read out in a read operation.

(5/5)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF2C4H	Serial clock select register 2	CSIS2	R/W	√	√		00H
FFFFF300H	Asynchronous serial interface mode register 0	ASIM0	R/W	√	√		00H
FFFFF302H	Asynchronous serial interface status register 0	ASIS0	R	√	√		00H
FFFFF304H	Baud rate generator control register 0	BRGC0	R/W		√		00H
FFFFF306H	Transmission shift register 0	TXS0	W		√		FFH
FFFFF308H	Reception buffer register 0	RXB0	R		√		FFH
FFFFF30EH	Baud rate generator mode control register 00	BRGMC0	R/W		√		00H
FFFFF310H	Asynchronous serial interface mode register 1	ASIM1	R/W	√	√		00H
FFFFF312H	Asynchronous serial interface status register 1	ASIS1	R	√	√		00H
FFFFF314H	Baud rate generator control register 1	BRGC1	R/W		√		00H
FFFFF316H	Transmission shift register 1	TXS1	W		√		FFH
FFFFF318H	Reception buffer register 1	RXB1	R		√		FFH
FFFFF31EH	Baud rate generator mode control register 1	BRGMC1	R/W		√		00H
FFFFF320H	Baud rate generator mode control register 01	BRGMC01	R/W		√		00H
FFFFF340H	IIC control register 0[Note]	IICC0	R/W	√	√		00H
FFFFF342H	IIC status register 0[Note]	IICS0	R	√	√		00H
FFFFF344H	IIC clock select register 0[Note]	IICCL0	R/W	√	√		00H
FFFFF346H	Slave address register 0[Note]	SVA0	R/W		√		00H
FFFFF348H	IIC shift register 0[Note]	IIC0	R/W		√		00H
FFFFF34AH	IIC function expansion register 0[Note]	IICX0	R/W	√	√		00H
FFFFF360H	Watch timer mode register	WTM	R/W	√	√		00H
FFFFF380H	Oscillation stabilization time select register	OSTS	R/W		√		04H
FFFFF382H	Watchdog timer clock select register	WDCS	R/W		√		00H
FFFFF384H	Watchdog timer mode register	WDTM	R/W	√	√		00H
FFFFF3A0H	Real-time output buffer register L	RTBL	R/W		√		00H
FFFFF3A2H	Real-time output buffer register H	RTBH	R/W		√		00H
FFFFF3A4H	Real-time output port mode register	RTPM	R/W	√	√		00H
FFFFF3A6H	Real-time output port control register	RTPC	R/W	√	√		00H
FFFFF3C0H	A/D converter mode register	ADM	R/W	√	√		00H
FFFFF3C2H	Analog input channel specification register	ADS	R/W	√	√		00H
FFFFF3C4H	A/D conversion result register	ADCR	R			√	0000H
FFFFF3C6H	A/D conversion result register H (higher 8 bits)	ADCRH	R		√		00H

★ **Note** Valid only for the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

⋆ 3.4.9 Specific registers

Specific registers are registers that are protected from being written with illegal data due to erroneous program execution, etc. The write access of these specific registers is executed in a specific sequence, and if abnormal write operations occur, it is checked by the PRERR bit of the system status register (SYS). The V850/SA1 has three specific registers, the power save control register (PSC), processor clock control register (PCC), and flash programming mode control register (FLPMC). For details of the PSC register, refer to **6.3.1 (2) Power save control register (PSC),** for details of the PCC register, refer to **6.3.1 (1) Processor clock control register (PCC),** and for details of the FLPMC register, refer to **16.7.12 Flash programming mode control register (FLPMC).**

The following sequence shows the data setting of the specific registers.

<1> Disable DMA operation.
<2> Set the PSW NP bit to 1 (interrupt disabled).
<3> Write any 8-bit data in the command register (PRCMD).
<4> Write the set data in the specific registers (using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<5> Return the PSW NP bit to 0 (interrupt disable canceled).
<6> If necessary, enable DMA operation.

No special sequence is required when reading the specific registers.

Cautions 1. If an interrupt request or a DMA request is acknowledged between the time PRCMD is generated (<3>) and the specific register write operation (<4>) that follows immediately after, the write operation to the specific register is not performed and a protection error (PRERR bit of SYS register is 1) may occur. Therefore, set the NP bit of PSW to 1 (<2>) to disable the acknowledgement of INT/NMI or to disable DMA transfer.
The above also applies when a bit manipulation instruction is used to set a specific register.

A description example is given below.

[Description example]: In case of PCC register

```
LDSR rX.5              ; NP bit = 1
ST.B  r0, PRCMD [r0]   ; Write to PRCMD
ST.B  rD, PCC [r0]     ; PCC register setting
LDSR rY, 5             ; NP bit = 0
        .
        .
        .
```

rX: Value to be written to PSW
rY: Value to be written back to PSW
rD: Value to be set to PCC

When saving the value of PSW, the value of PSW prior to setting the NP bit must be transferred to the rY register.

Cautions 2. Always stop the DMA prior to accessing specific registers.

3. When data is set to the PSC register in order to set the IDLE mode or software STOP mode, a dummy instruction must be inserted so that the routine after releasing the IDLE/software STOP mode is executed correctly. For details, refer to 6.6 Cautions on Power Save Function.

4. When the FLSPM bit of the FLPMC register is manipulated to switch between the normal mode and the flash memory self-programming mode, a dummy instruction must be inserted. For details, refer to 16.7.12 Flash programming mode control register (FLPMC).

(1) Command register (PRCMD)

The command register (PRCMD) is a register used when write-accessing a specific register to prevent incorrect writing to the specific register due to the erroneous program execution.

This register can be written in 8-bit units. It becomes undefined in a read cycle.

Occurrence of illegal write operations can be checked by the PRERR bit of the SYS register.

After reset: Undefined W Address: FFFFF170H

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

REGn	Registration Code
0/1	Any 8-bit data

Remark n = 0 to 7

(2) System status register (SYS)

This register is allocated with status flags showing the operating state of the entire system. This register can be read/written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF078H

Symbol	7	6	5	4	3	2	1	0
SYS	0	0	0	PRERR	0	0	0	0

PRERR	Detection of Protection Error
0	Protection error did not occur
1	Protection error occurred

The operating conditions of PRERR flag are shown below.

(a) Set conditions (PRERR = 1)

(1) When a write operation to the specific register took place in a state where the store instruction operation for the recent peripheral I/O was not a write operation to the PRCMD register

(2) When the first store instruction operation following a write operation to the PRCMD register is to any peripheral I/O register (including the PRCMD register and SYS register) apart from specific registers

(b) Reset conditions (PRERR = 0)

(1) When 0 is written to the PRERR flag of the SYS register[Note]

(2) At system reset

Note If 0 is written to the PRERR flag immediately after writing to the PRCMD register, the PRERR flag is set to 1 (because the SYS register is not a specific register).

CHAPTER 4 BUS CONTROL FUNCTION

The V850/SA1 is provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

4.1 Features

- Address bus (capable of separate output)
- 16-bit data bus
- Able to be connected to external devices via the pins that have alternate functions as ports
- Wait function
  - Programmable wait function, capable of inserting up to 3 wait states per 2 blocks
  - External wait control through $\overline{\text{WAIT}}$ input pin
- Idle state insertion function
- Bus hold function

4.2 Bus Control Pins and Control Register

4.2.1 Bus control pins

The following pins are used for interfacing with external devices.

Table 4-1. Bus Control Pins

External Bus Interface Function	Corresponding Port (Pins)
Address/data bus (AD0 to AD7)	Port 4 (P40 to P47)
Address/data bus (AD8 to AD15)	Port 5 (P50 to P57)
Address bus (A1 to A4)	Port 11 (P110 to P113)
Address bus (A5 to A12)	Port 10 (P100 to P107)
Address bus (A13 to A15)	Port 3 (P34 to P36)
Address bus (A16 to A21)	Port 6 (P60 to P65)
Read/write control ($\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$, R/$\overline{\text{W}}$, $\overline{\text{DSTB}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$)	Port 9 (P90 to P93)
Address strobe (ASTB)	Port 9 (P94)
Bus hold control ($\overline{\text{HLDRQ}}$, $\overline{\text{HLDAK}}$)	Port 9 (P95, P96)
External wait control ($\overline{\text{WAIT}}$)	Port 12 (P120)

The bus interface function of each pin is enabled by specifying the memory expansion mode register (MM) or the memory address output mode register (MAM). For the details of specifying an operation mode of the external bus interface, refer to **3.4.6 (1) Memory expansion mode register (MM)** and for **(2) Memory address output mode register (MAM).**

Caution For debugging using the separate bus, refer to IE-703017-MC-EM1 User's Manual.

4.2.2 Control register

(1) System control register (SYC)

This register switches the control signals for bus interface.

The system control register can be read/written in 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF064H

Symbol	7	6	5	4	3	2	1	0
SYC	0	0	0	0	0	0	0	BIC

BIC	Bus Interface Control
0	$\overline{DSTB}$, R/$\overline{W}$, $\overline{LBEN}$, $\overline{UBEN}$ signal output
1	$\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{UBEN}$[Note] signal output

Note The $\overline{UBEN}$ signal is output regardless of the BIC bit setting in the external expansion mode (set by the memory expansion mode register (MM)).

Caution When using port 9 as an I/O port, be sure to set the BIC bit to 0.

4.3 Bus Access

4.3.1 Number of access clocks

The number of basic clocks necessary for accessing each resource is as follows.

Table 4-2. Number of Access Clocks

Bus Cycle Type	Peripheral I/O (Bus Width)			
	Internal ROM (32 Bits)	Internal RAM (32 Bits)	Peripheral I/O (16 Bits)	External Memory (16 Bits)
Instruction fetch	1	3	Disabled	3 + n
Operand data access	3	1	3	3 + n

Remarks 1. Unit: Clock/access

2. n: Number of waits inserted

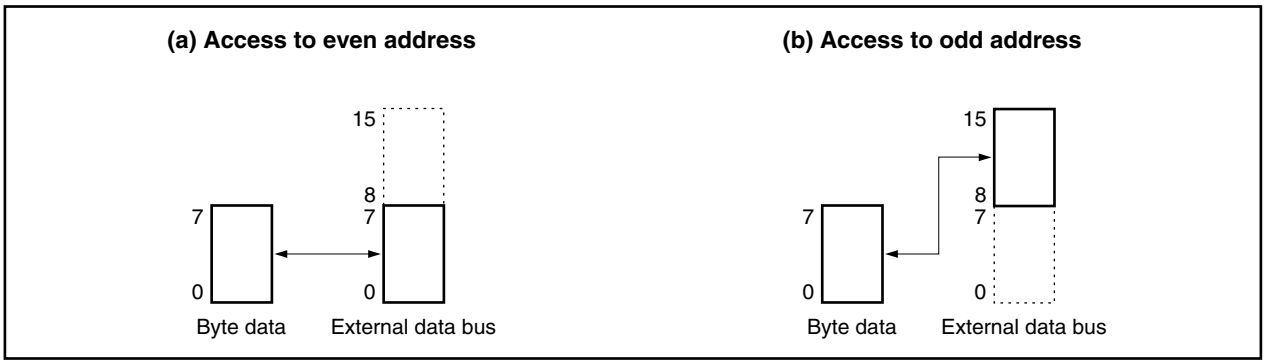
4.3.2 Bus width

The CPU carries out peripheral I/O access and external memory access in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each access.

(1) Byte access (8 bits)

Byte access is divided into two types, access to even addresses and access to odd addresses.

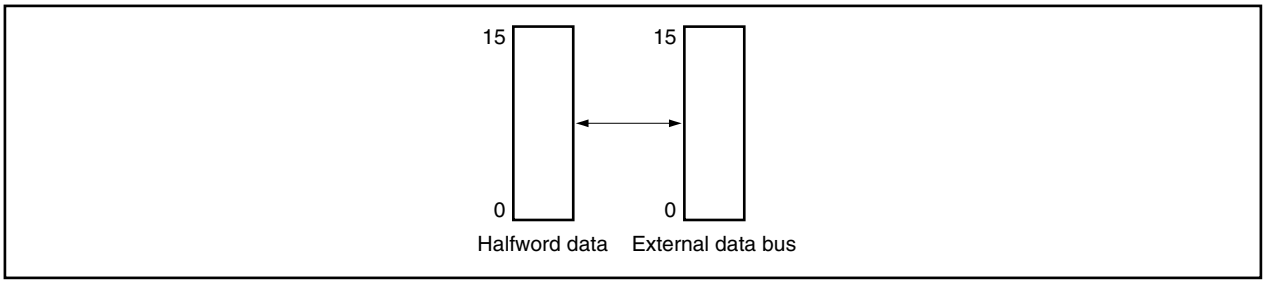
Figure 4-1. Byte Access (8 Bits)



(2) Halfword access (16 bits)

In halfword access to external memory, data is dealt with as is because the data bus is fixed to 16 bits.

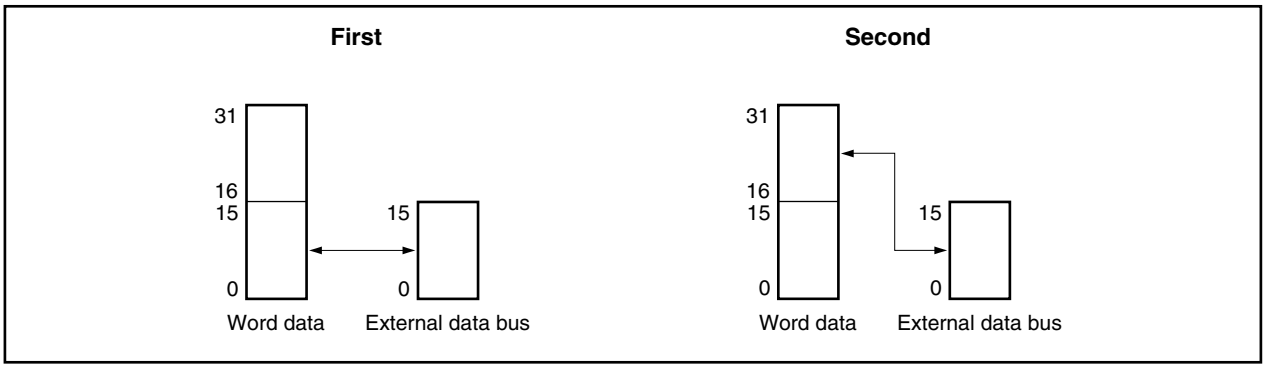
Figure 4-2. Halfword Access (16 Bits)



(3) Word access (32 bits)

In word access to external memory, the lower halfword is accessed first and then the higher halfword is accessed.

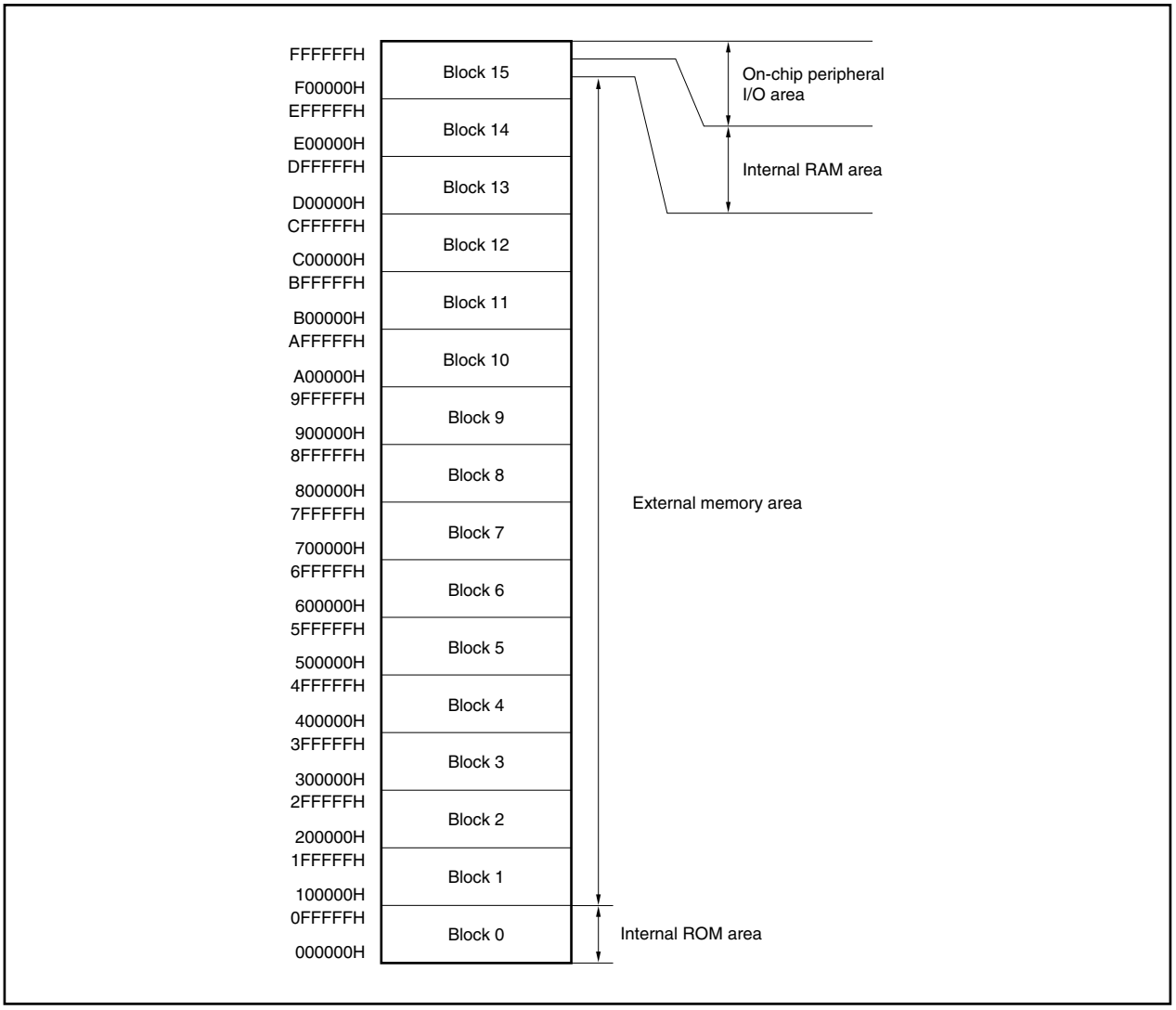
Figure 4-3. Word Access (32 Bits)



4.4 Memory Block Function

The 16 MB memory space is divided into memory blocks of 1 MB units. The programmable wait function and bus cycle operation mode can be independently controlled for every two memory blocks.

Figure 4-4. Memory Space



4.5 Wait Function

4.5.1 Programmable wait function

To facilitate interfacing with low-speed memories and I/O devices, up to 3 data waits can be inserted in a bus cycle that starts every two memory blocks.

The number of waits can be programmed by using the data wait control register (DWC). Immediately after the system has been reset, a state in which three data waits are inserted is automatically programmed for all memory blocks.

(1) Data wait control register (DWC)

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

After reset: FFFFH R/W Address: FFFFF060H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWC	DW71	DW70	DW61	DW60	DW51	DW50	DW41	DW40	DW31	DW30	DW21	DW20	DW11	DW10	DW01	DW00

DWn1	DWn0	Number of Wait States to Be Inserted
0	0	0
0	1	1
1	0	2
1	1	3

n	Blocks into Which Wait States Are Inserted
0	Blocks 0/1
1	Blocks 2/3
2	Blocks 4/5
3	Blocks 6/7
4	Blocks 8/9
5	Blocks 10/11
6	Blocks 12/13
7	Blocks 14/15

Block 0 is reserved for the internal ROM area. It is not subject to programmable wait control, regardless of the setting of DWC, and is always accessed without wait states.

The internal RAM area of block 15 is not subject to programmable wait control and is always accessed without wait states. The on-chip peripheral I/O area of this block is not subject to programmable wait control either; only wait control from each peripheral function is performed.

4.5.2 External wait function

When an extremely slow device, I/O, or asynchronous system is connected, any number of wait states can be inserted in a bus cycle by sampling the external wait pin ($\overline{\text{WAIT}}$) to synchronize with the external device.

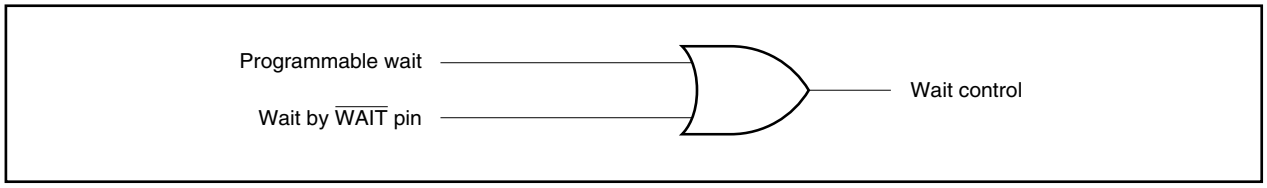
The external wait signal is data wait only, and does not affect the access times of the internal ROM, internal RAM, and on-chip peripheral I/O areas, similar to programmable wait.

Input of the external $\overline{\text{WAIT}}$ signal can be done asynchronously to CLKOUT and is sampled at the falling edge of the clock in the T2 and TW states of the bus cycle. If the setup/hold time at sampling timing is not satisfied, the wait state may or may not be inserted in the next state.

4.5.3 Relationship between programmable wait and external wait

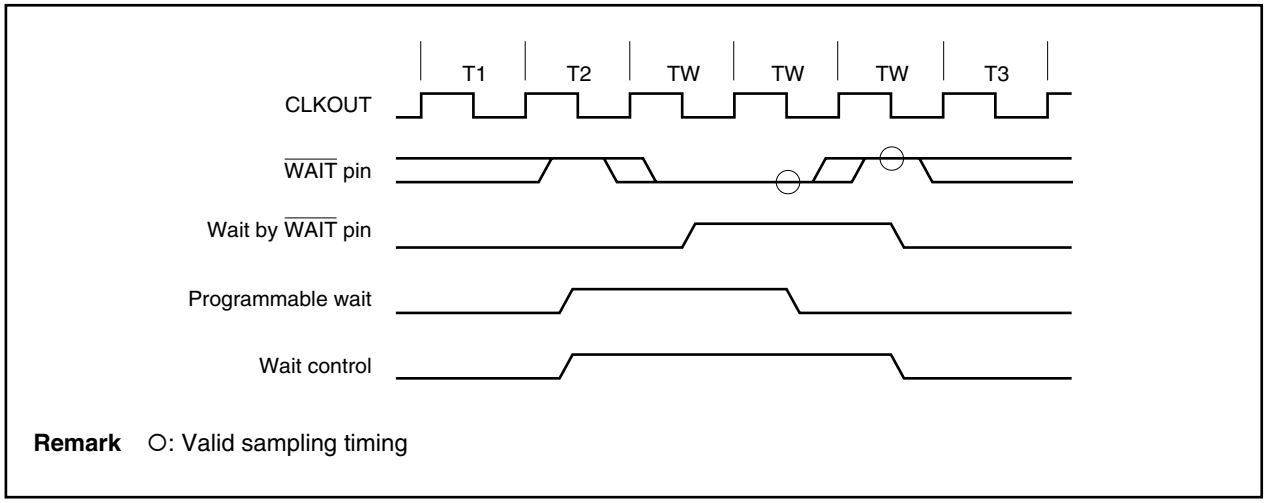
A wait cycle is inserted as a result of an OR operation between the wait cycle specified by the set value of programmable wait and the wait cycle controlled by the $\overline{\text{WAIT}}$ pin. In other words, the number of wait cycles is determined by whichever side has the greatest number.

Figure 4-5. Wait Control



For example, if the number of programmable waits and the timing of the $\overline{\text{WAIT}}$ pin input signal are as illustrated below, three wait states will be inserted in the bus cycle.

Figure 4-6. Example of Inserting Wait States



Remark ○: Valid sampling timing

4.6 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices and meeting the data output float delay time on memory read accesses every two blocks, one idle state (TI) can be inserted into the current bus cycle after the T3 state. The following bus cycle starts after one idle state.

Specifying insertion of the idle state is programmable by using the bus cycle control register (BCC).

Immediately after the system has been reset, idle state insertion is automatically programmed for all memory blocks.

(1) Bus cycle control register (BCC)

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

After reset: AAAAH R/W Address: FFFFF062H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCC	BC71	0	BC61	0	BC51	0	BC41	0	BC31	0	BC21	0	BC11	0	BC01	0

BCn1	Idle State Insertion Specification
0	Not inserted
1	Inserted

n	Blocks into Which Idle State Is Inserted
0	Blocks 0/1
1	Blocks 2/3
2	Blocks 4/5
3	Blocks 6/7
4	Blocks 8/9
5	Blocks 10/11
6	Blocks 12/13
7	Blocks 14/15

Block 0 is reserved for the internal ROM area, so no idle state can be specified.

The internal RAM area and on-chip peripheral I/O area of block 15 are not subject to insertion of an idle state.

Be sure to set bits 0, 2, 4, 6, 8, 10, 12, and 14 to 0. If these bits are set to 1, the operation is not guaranteed.

4.7 Bus Hold Function

4.7.1 Outline of function

When the MM3 bit of the memory expansion mode register (MM) is set (1), the $\overline{\text{HLDRQ}}$ and $\overline{\text{HLDAK}}$ pin functions of P95 and P96 become valid.

When the $\overline{\text{HLDRQ}}$ pin becomes active (low) indicating that another bus master is requesting acquisition of the bus, the external address/data bus and strobe pins go into a high-impedance state[Note], and the bus is released (bus hold status). When the $\overline{\text{HLDRQ}}$ pin becomes inactive (high) indicating that the request for the bus is cleared, these pins are driven again. During the bus hold period, the internal operation continues until the next external memory access.

The bus hold status can be recognized by the $\overline{\text{HLDAK}}$ pin becoming active (low).

This feature can be used to design a system where two or more bus masters exist, such as when a multi-processor configuration is used and when a DMA controller is connected.

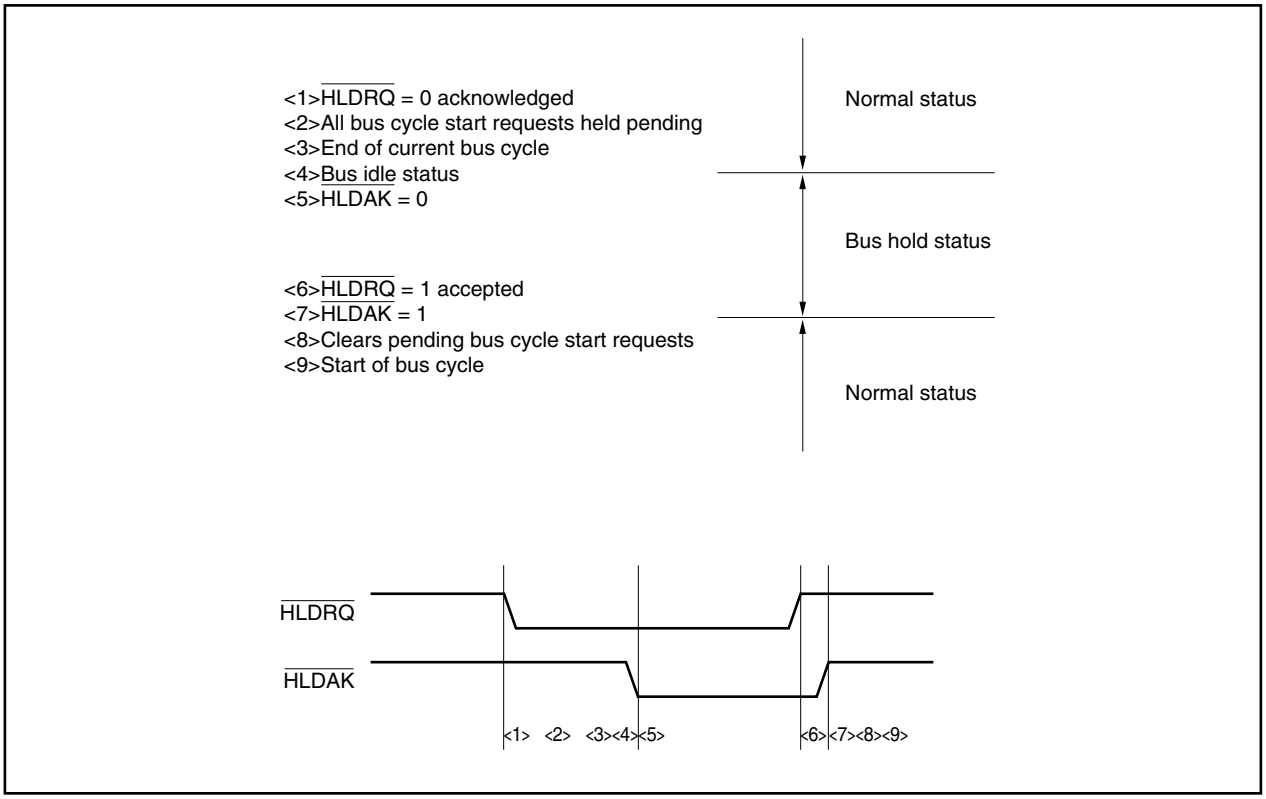
A bus hold request is not acknowledged between the first and the second word access, and between the read access and write access in a read-modify-write access executed using a bit manipulation instruction.

Note The A1 to A15 pins are set to the hold state when a separate bus is used.

4.7.2 Bus hold procedure

The procedure of the bus hold function is illustrated below.

Figure 4-7. Bus Hold Procedure



4.7.3 Operation in power save mode

In the IDLE or software STOP mode, the system clock is stopped. Consequently, the bus hold status is not set even if the $\overline{\text{HLDRQ}}$ pin becomes active.

In the HALT mode, the $\overline{\text{HLDAK}}$ pin immediately becomes active when the $\overline{\text{HLDRQ}}$ pin becomes active, and the bus hold status is set. When the $\overline{\text{HLDRQ}}$ pin becomes inactive, the $\overline{\text{HLDAK}}$ pin becomes inactive. As a result, the bus hold status is cleared, and the HALT mode is set again.

4.8 Bus Timing

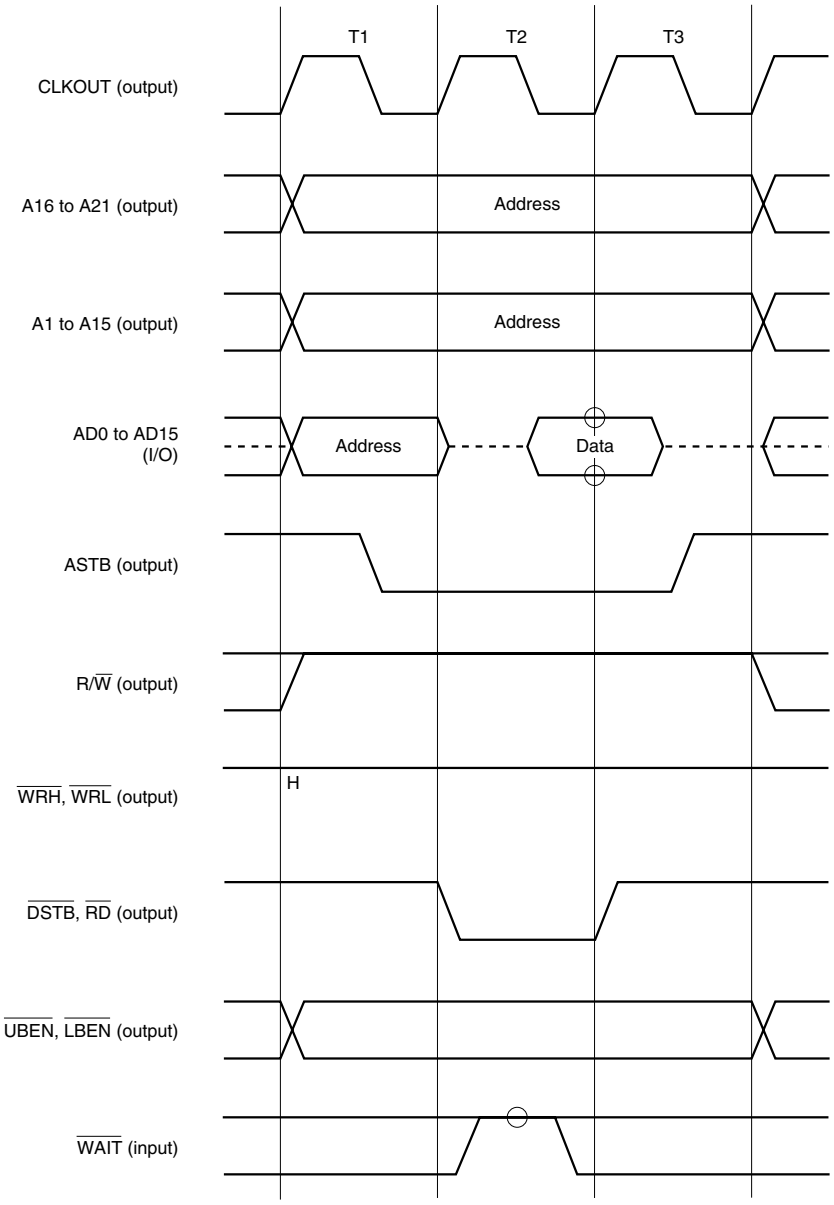
The V850/SA1 can execute read/write control for an external device using the following two modes.

- Mode using $\overline{\text{DSTB}}$, R/$\overline{\text{W}}$, $\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$, and ASTB signals
- Mode using $\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, and ASTB signals

Set these modes by using the BIC bit of the system control register (SYC) (refer to **4.2.2 (1) System control register (SYC)**).

Figure 4-8. Memory Read (1/4)

(a) 0 waits

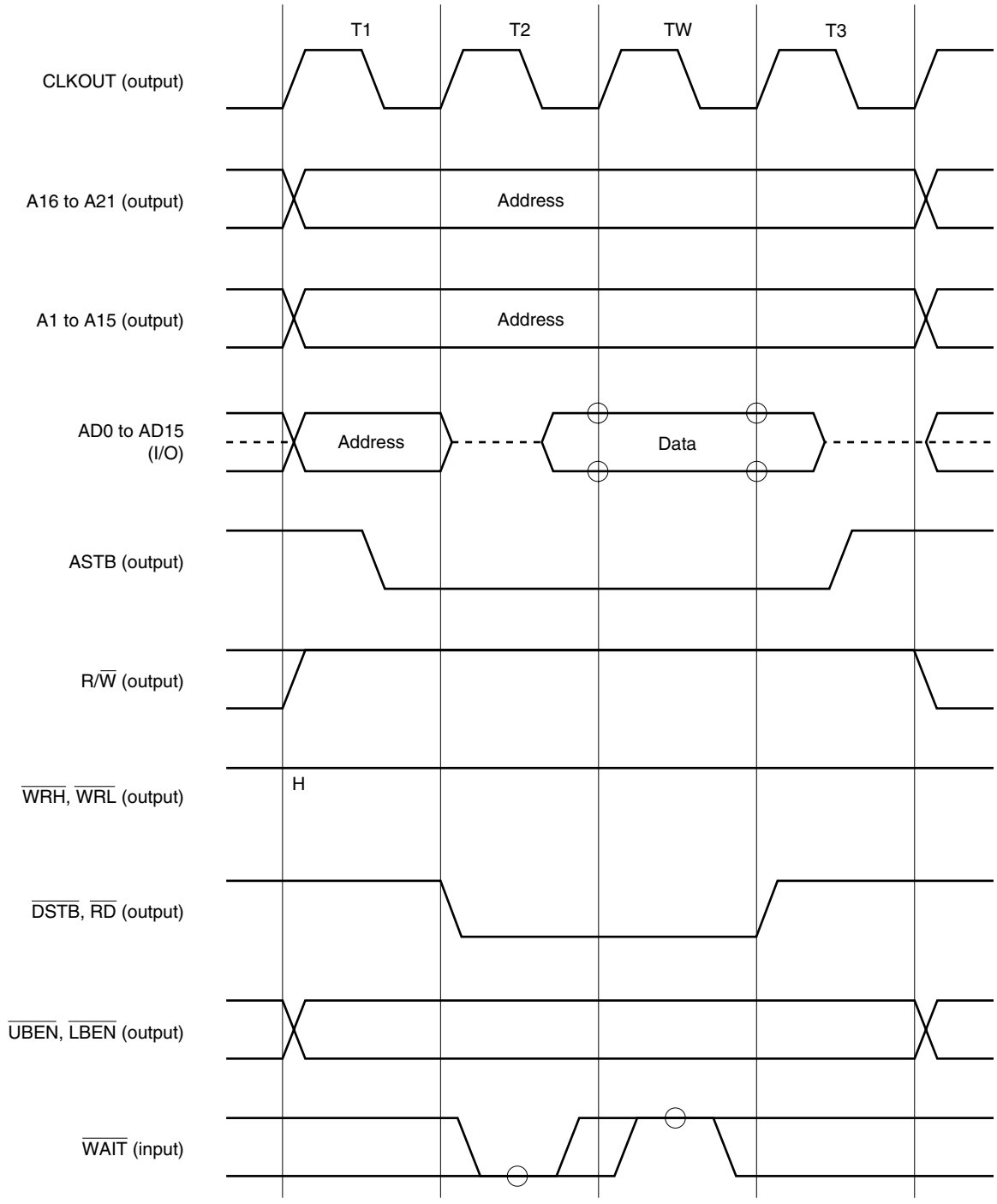


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 0.

2. The broken line indicates the high-impedance state.

Figure 4-8. Memory Read (2/4)

(b) 1 wait

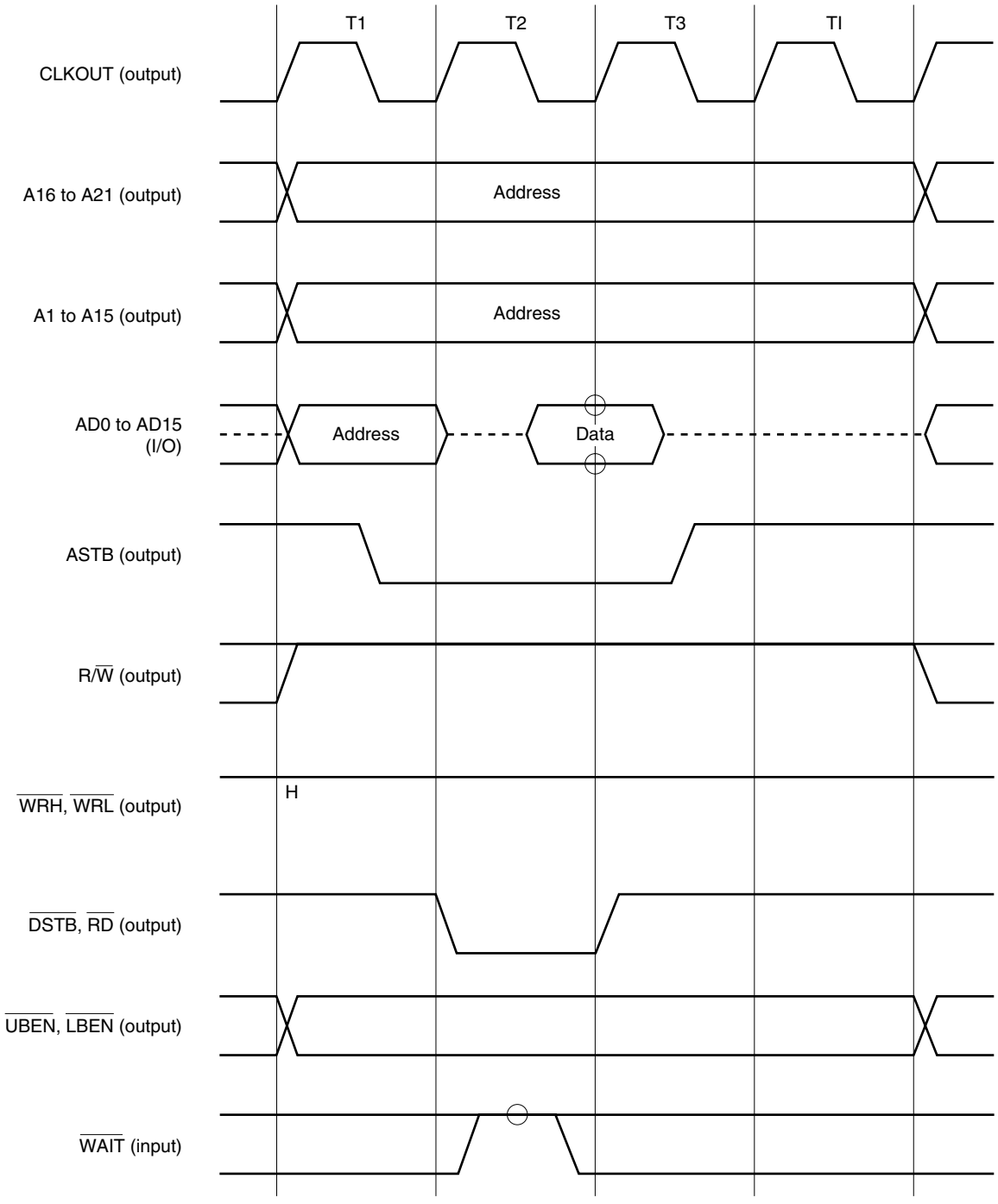


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 1.

2. The broken line indicates the high-impedance state.

Figure 4-8. Memory Read (3/4)

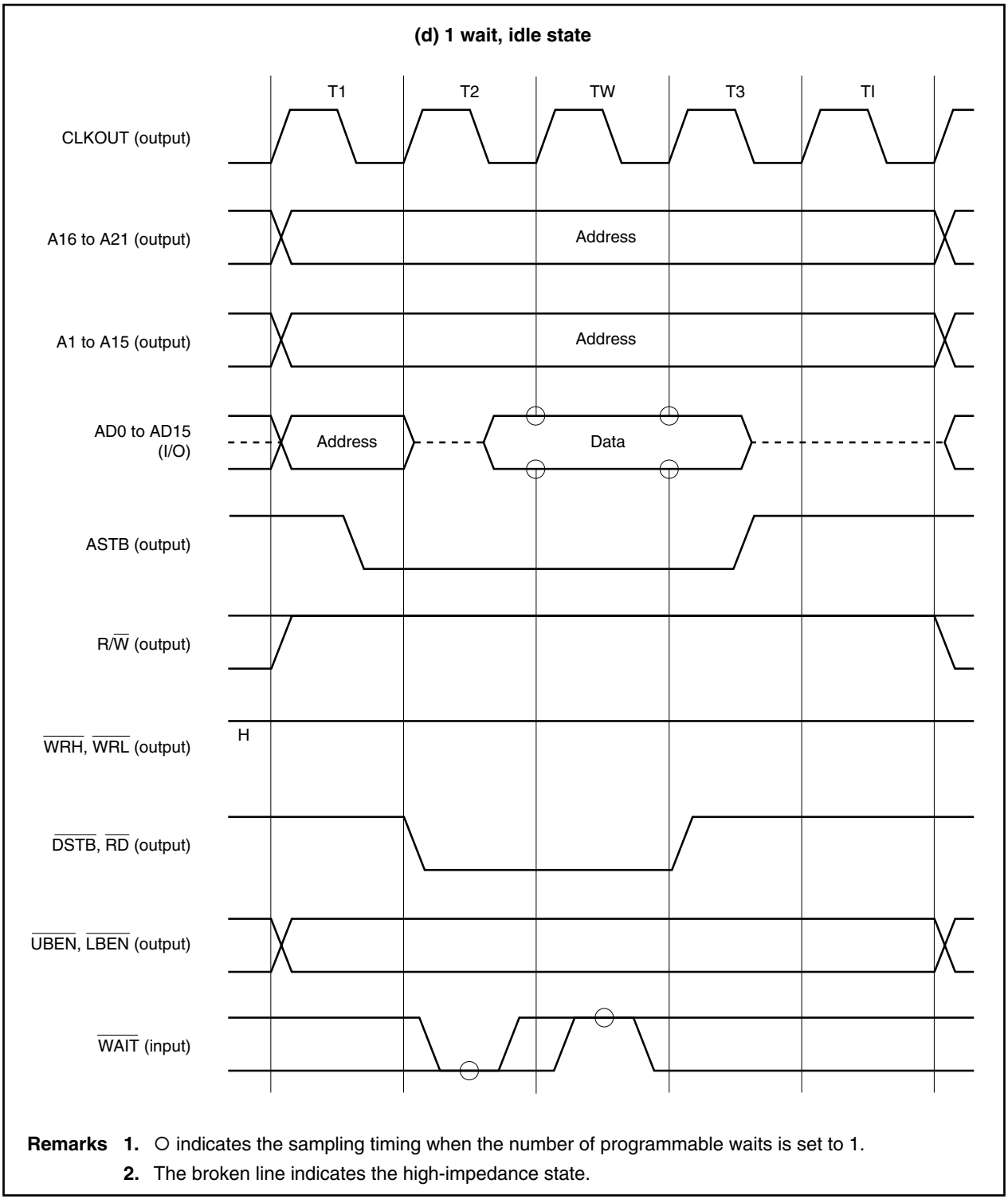
(c) 0 waits, idle state



Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 0.

2. The broken line indicates the high-impedance state.

Figure 4-8. Memory Read (4/4)

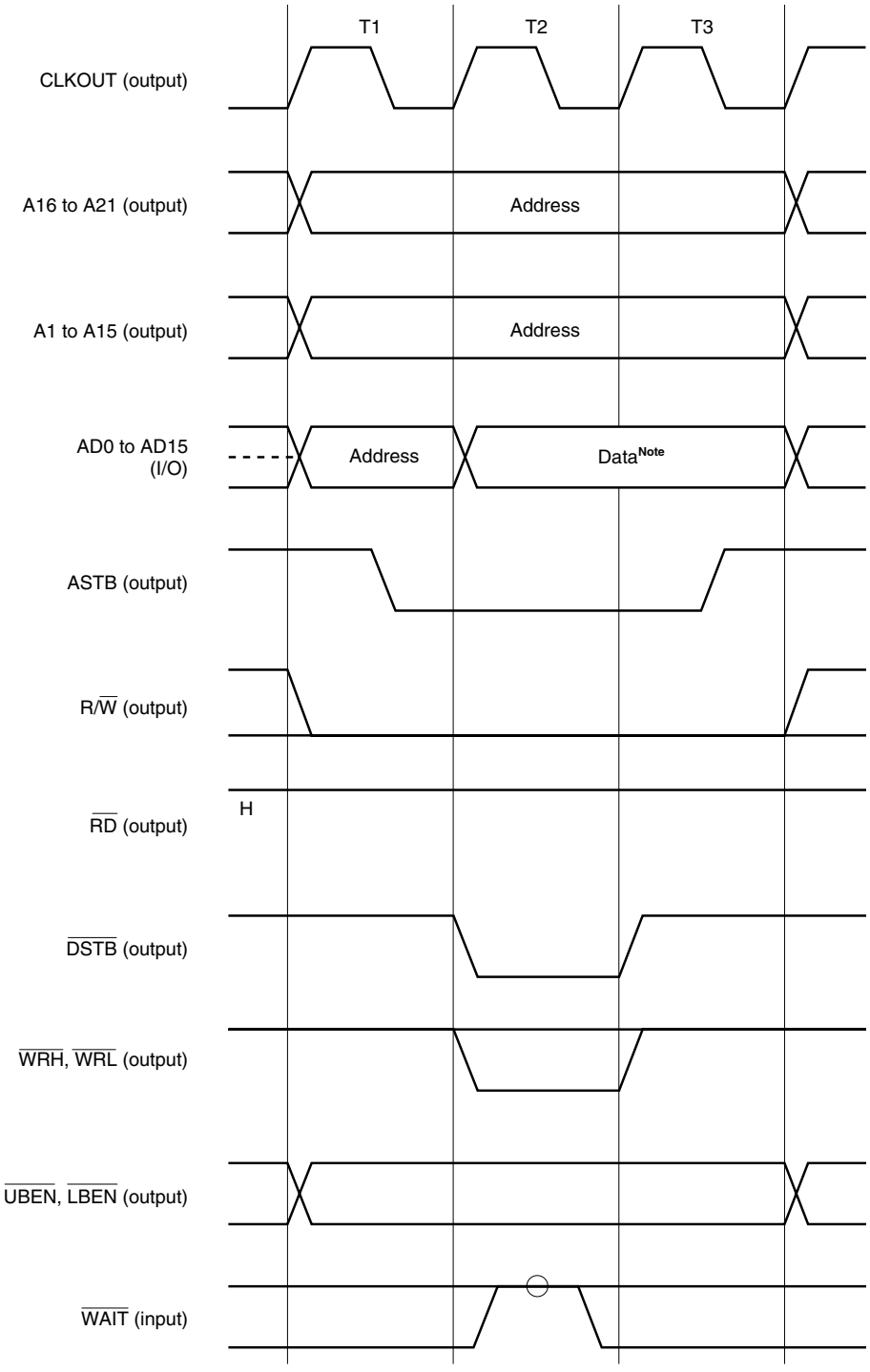


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 1.

2. The broken line indicates the high-impedance state.

Figure 4-9. Memory Write (1/2)

(a) 0 waits

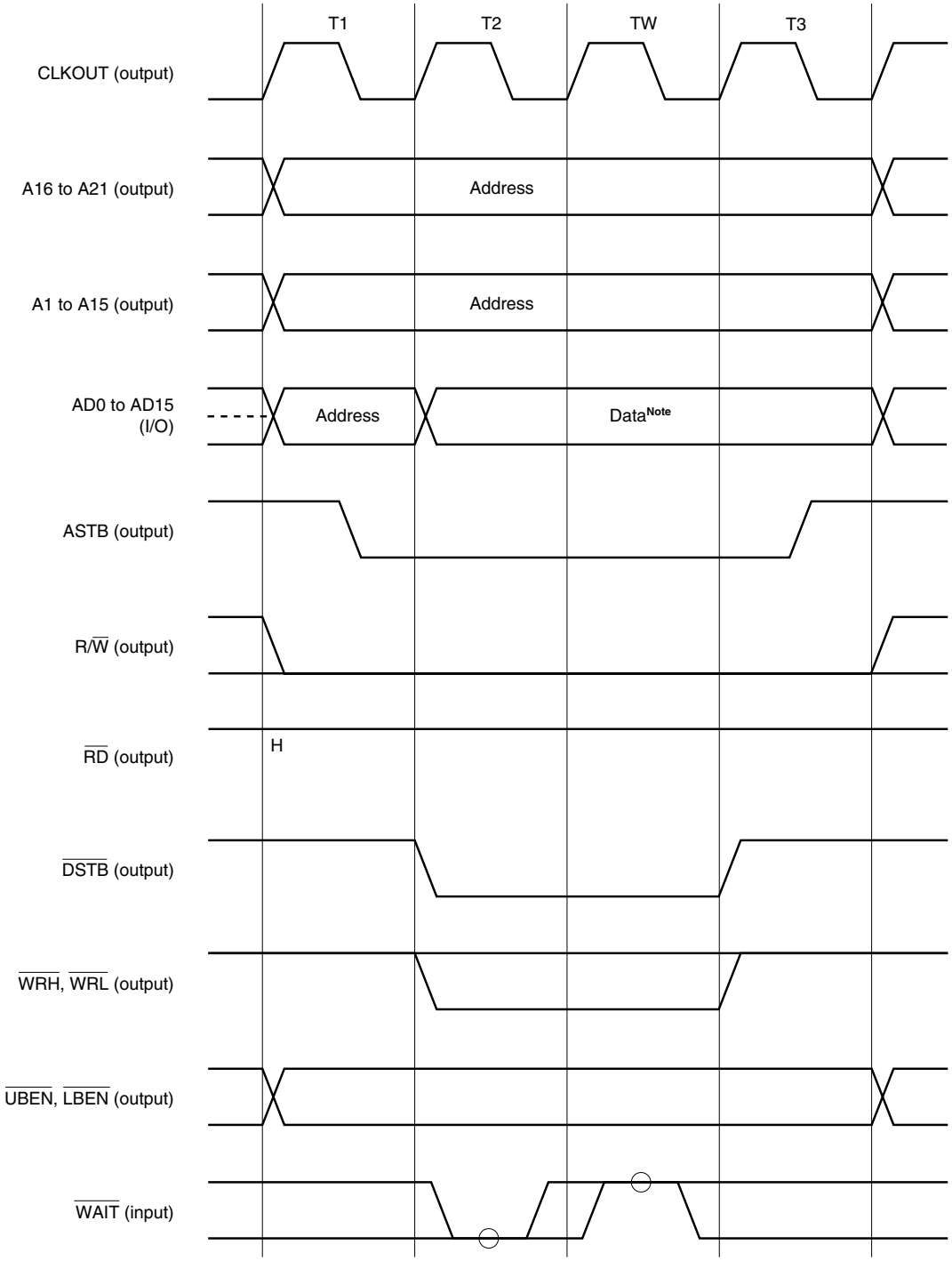


Note AD0 to AD7 output invalid data when odd-address byte data is accessed.
AD8 to AD15 output invalid data when even-address byte data is accessed.

Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The broken line indicates the high-impedance state.

Figure 4-9. Memory Write (2/2)

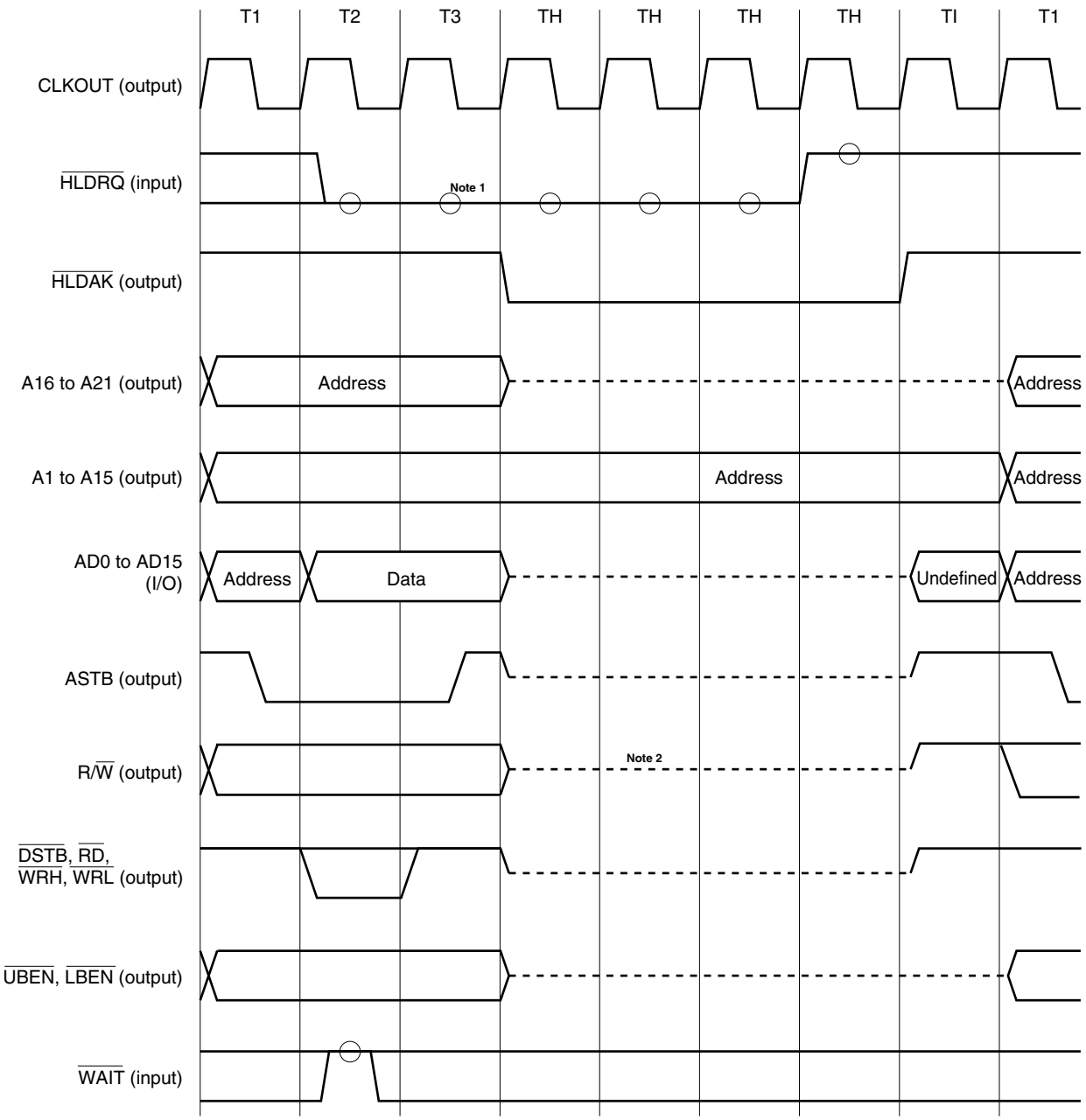
(b) 1 wait



Note AD0 to AD7 output invalid data when odd-address byte data is accessed.
AD8 to AD15 output invalid data when even-address byte data is accessed.

Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 1.
2. The broken line indicates the high-impedance state.

Figure 4-10. Bus Hold Timing



Notes 1. If the HLDRQ signal is inactive (high level) at this sampling timing, the bus hold state is not entered.

2. If the bus hold status is entered after a write cycle, a high level may be output momentarily from the R/W pin immediately before the HLDAK signal changes from high level to low level.

Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 0.

2. The broken line indicates the high-impedance state.

4.9 Bus Priority

There are four external bus cycles: bus hold, memory access, instruction fetch (branch), and instruction fetch (continuous). The bus hold cycle is given the highest priority, followed by memory access, instruction fetch (branch), and instruction fetch (continuous) in that order.

The instruction fetch cycle may be inserted between the read access and write access in a read-modify-write access.

No instruction fetch cycle is inserted between the lower halfword access and higher halfword access of word access operations.

Table 4-3. Bus Priority

External Bus Cycle	Priority
Bus hold	1
Memory access	2
Instruction fetch (branch)	3
Instruction fetch (continuous)	4

4.10 Memory Boundary Operation Conditions

4.10.1 Program space

(1) Do not execute a branch to the on-chip peripheral I/O area or continuous fetch from the internal RAM area to peripheral I/O area. If a branch or instruction fetch is executed, the NOP instruction code is continuously fetched and no data is fetched from external memory.

(2) A prefetch operation straddling over the on-chip peripheral I/O area (invalid fetch) does not take place if a branch instruction exists at the upper-limit address of the internal RAM area.

4.10.2 Data space

Only the address aligned at the halfword boundary (when the least significant bit of the address is “0”)/word boundary (when the lowest 2 bits of the address are “0”) is accessed by halfword (16 bits)/word (32 bits) data. Therefore, access that straddles over the memory or memory block boundary does not take place.

For details, refer to **V850 Series Architecture User’s Manual**.

CHAPTER 5 INTERRUPT/EXCEPTION PROCESSING FUNCTION

5.1 Outline

The V850/SA1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and realizes a high-powered interrupt function that can service interrupt requests from a total of 30 sources.

An interrupt is an event that occurs independently of program execution, and an exception is an event that is dependent on program execution.

The V850/SA1 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 (fetching of an illegal opcode) (exception trap).

5.1.1 Features

- Interrupts
  - External interrupts: 8 sources (5 sources[Note])
  - Internal interrupts: 24 sources
  - 8 levels of programmable priorities
  - Mask specification for interrupt requests according to priority
  - Masks can be specified for each maskable interrupt request.
  - Noise elimination, edge detection, and valid edge of external interrupt request signal can be specified.

  Note Number of external interrupts that can release the software STOP mode.

- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 1 source (illegal opcode exception)

The interrupt/exception sources are listed in Table 5-1.

Table 5-1. Interrupt Source List (1/2)

Type	Classification	Default Priority	Name	Trigger	Interrupt Source	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	Reset input	–	0000H	00000000H	Undefined	–
Non-maskable	Interrupt	–	NMI	NMI pin input	Pin	0010H	00000010H	nextPC	–
	Interrupt	–	INTWDT	WDTOVF non-maskable	WDT	0020H	00000020H	nextPC	–
Software exception	Exception	–	TRAP0n	TRAP instruction	–	004nH[Note 1]	00000040H	nextPC	–
	Exception	–	TRAP1n	TRAP instruction	–	005nH[Note 1]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP	Illegal opcode	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTWDTM	WDTOVF maskable	WDT	0080H	00000080H	nextPC	WDTIC
		1	INTP0	INTP0 pin	Pin	0090H	00000090H	nextPC	PIC0
		2	INTP1	INTP1 pin	Pin	00A0H	000000A0H	nextPC	PIC1
		3	INTP2	INTP2 pin	Pin	00B0H	000000B0H	nextPC	PIC2
		4	INTP3	INTP3 pin	Pin	00C0H	000000C0H	nextPC	PIC3
		5	INTP4	INTP4 pin	Pin	00D0H	000000D0H	nextPC	PIC4
		6	INTP5	INTP5 pin	Pin	00E0H	000000E0H	nextPC	PIC5
		7	INTP6	INTP6 pin	Pin	00F0H	000000F0H	nextPC	PIC6
		8	INTWTI	Watch timer prescaler	WT	0100H	00000100H	nextPC	WTIIC
		9	INTTM00	INTTM00	TM0	0110H	00000110H	nextPC	TMIC00
		10	INTTM01	INTTM01	TM0	0120H	00000120H	nextPC	TMIC01
		11	INTTM10	INTTM10	TM1	0130H	00000130H	nextPC	TMIC10
		12	INTTM11	INTTM11	TM1	0140H	00000140H	nextPC	TMIC11
		13	INTTM2	TM2 compare match/OVF	TM2	0150H	00000150H	nextPC	TMIC2
		14	INTTM3	TM3 compare match/OVF	TM3	0160H	00000160H	nextPC	TMIC3
		15	INTTM4	TM4 compare match/OVF	TM4	0170H	00000170H	nextPC	TMIC4
		16	INTTM5	TM5 compare match/OVF	TM5	0180H	00000180H	nextPC	TMIC5
		17	INTIIC0[Note 2]/ INTCSI0	I^2C interrupt/ CSI0 transmit end	I^2C/ CSI0	0190H	00000190H	nextPC	CSIC0
		18	INTSER0	UART0 serial error	UART0	01A0H	000001A0H	nextPC	SERIC0
		19	INTSR0/ INTCSI1	UART0 receive end/ CSI1 transmit end	UART0/ CSI1	01B0H	000001B0H	nextPC	CSIC1
		20	INTST0	UART0 transmit end	UART0	01C0H	000001C0H	nextPC	STIC0
		21	INTCSI2	CSI2 transmit end	CSI2	01D0H	000001D0H	nextPC	CSIC2

Notes 1. n: 0 to FH

★ **2.** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY and 70F3017AY.

Table 5-1. Interrupt Source List (2/2)

Type	Classification	Default Priority	Name	Trigger	Interrupt Source	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	22	INTSER1	UART1 serial error	UART1	01E0H	000001E0H	nextPC	SERIC1
		23	INTSR1	UART1 receive end	UART1	01F0H	000001F0H	nextPC	SRIC3
		24	INTST1	UART1 transmit end	UART1	0200H	00000200H	nextPC	STIC1
		25	INTAD	A/D conversion end	A/D	0210H	00000210H	nextPC	ADIC
		26	INTDMA0	DMA0 transfer end	DMA0	0220H	00000220H	nextPC	DMAIC0
		27	INTDMA1	DMA1 transfer end	DMA1	0230H	00000230H	nextPC	DMAIC1
		28	INTDMA2	DMA2 transfer end	DMA2	0240H	00000240H	nextPC	DMAIC2
		29	INTWT	Watch timer OVF	WT	0250H	00000250H	nextPC	WTIC

Remarks 1. Default Priority: Priority when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

Restored PC: The value of the PC saved to EIPC or FEPC when interrupt/exception processing is started. However, the value of the PC saved when an interrupt is acknowledged during DIVH (division) instruction execution is the value of the PC of the current instruction (DIVH).

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

3. The restored PC of an interrupt/exception other than RESET is the value of the PC (when an event occurred) + 1.

4. The non-maskable interrupt (INTWDT) and maskable interrupt (INTWDTM) are set by the WDTM4 bit of the watchdog timer mode register (WDTM).

5.2 Non-Maskable Interrupts

Non-maskable interrupt requests are acknowledged unconditionally, even in the interrupt disabled (DI) status. NMI requests are not subject to priority control and take precedence over all the other interrupts.

The V850/SA1 includes the following two non-maskable interrupt requests.

- NMI pin input (NMI)
- Non-maskable watchdog timer interrupt request (INTWDT)

When the valid edge specified by rising edge specification register 0 (EGP0) and falling edge specification register 0 (EGN0) is detected at the NMI pin, an interrupt occurs.

INTWDT functions as the non-maskable interrupt (INTWDT) only in the state in which the WDTM4 bit of the watchdog timer mode register (WDTM) is set to 1.

While the service routine of a non-maskable interrupt is being executed (PSW.NP = 1), the acknowledgement of another non-maskable interrupt request is held pending. The pending NMI is acknowledged when PSW.NP is cleared to 0 after the original service routine of the non-maskable interrupt under execution has been terminated (by the RETI instruction). Note that if two or more NMI requests are input during the execution of the service routine for an NMI, the number of NMIs that will be acknowledged after PSW.NP goes to ‘‘0’’, is only one.

Caution Do not clear PSW.NP to 0 by the LDSR instruction during non-maskable interrupt servicing. If PSW.NP is cleared to 0, the interrupts afterwards cannot be acknowledged correctly.

5.2.1 Operation

If a non-maskable interrupt request is generated, 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 codes (0010H, 0020H) to the higher halfword (FECC) of ECR.
<4> Sets the NP and ID bits of the PSW and clears the EP bit.
<5> Loads the handler address (00000010H, 00000020H) of the non-maskable interrupt routine to the PC, and transfers control.

Figure 5-1. Non-Maskable Interrupt Servicing

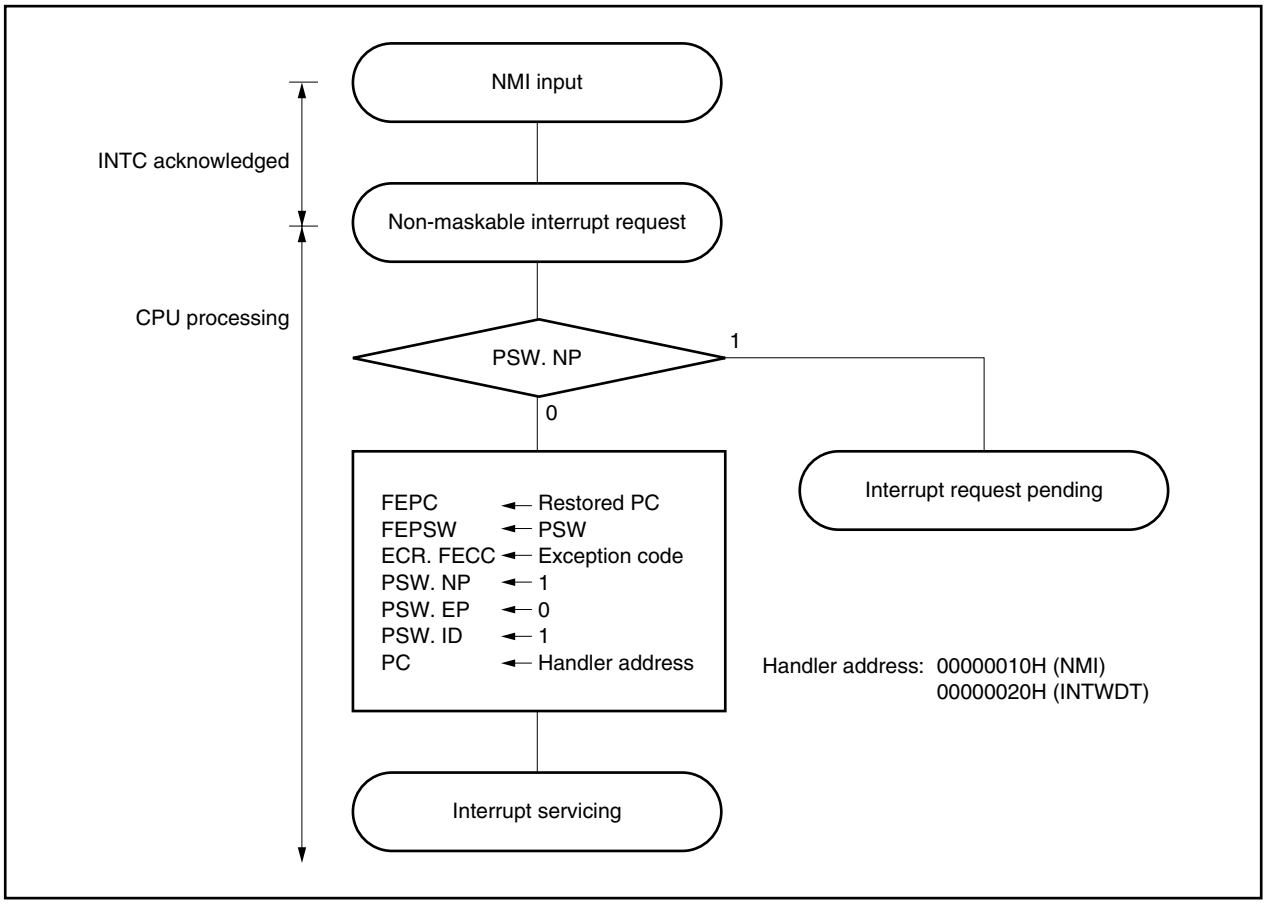
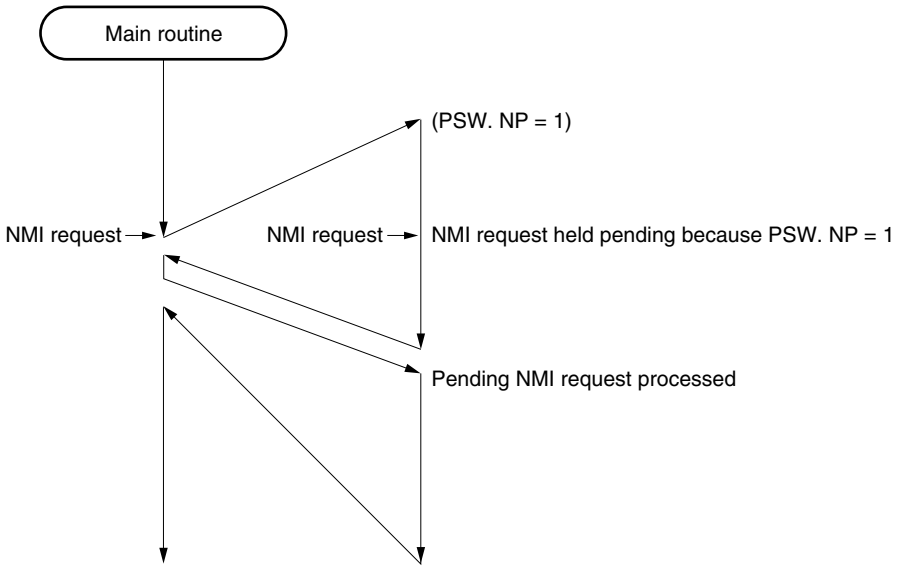
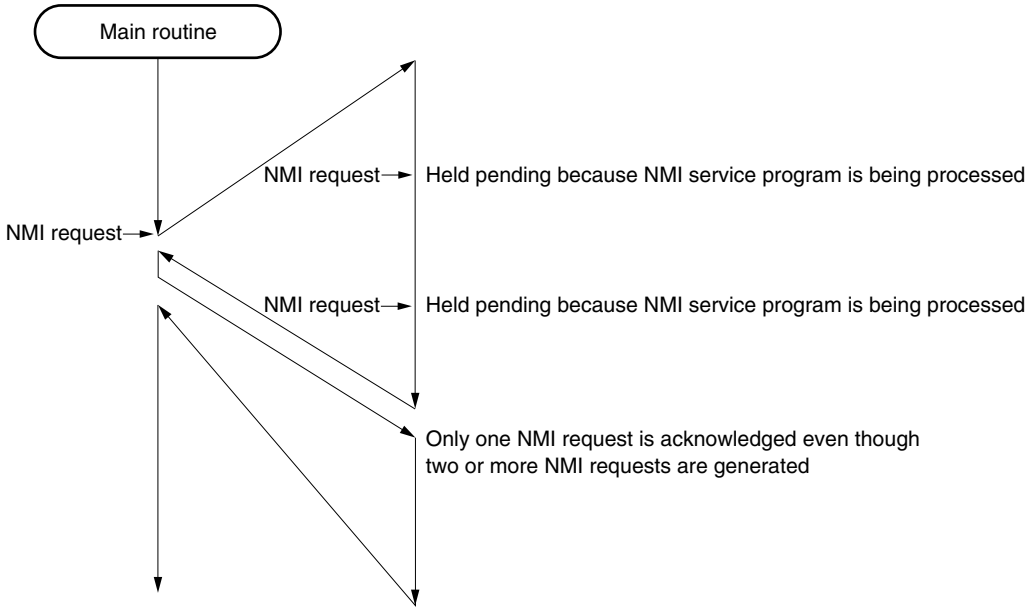


Figure 5-2. Acknowledging Non-Maskable Interrupt Request

(a) If a new NMI request is generated while an NMI service routine is being executed



(b) If a new NMI request is generated twice while an NMI service routine is being executed



5.2.2 Restore

Execution is restored from non-maskable interrupt servicing by the RETI instruction.

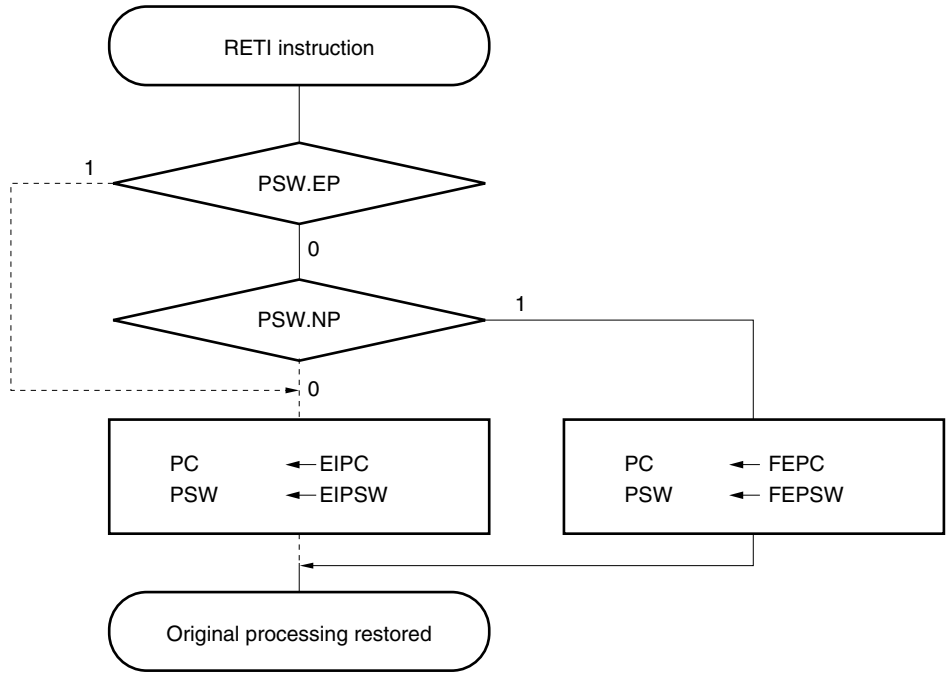
Operation of RETI instruction

When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

(1) Restores the values of the PC and PSW from FEPC and FEPSW, respectively, because the EP bit of PSW is 0 and the NP bit of 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 5-3. 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.

5.2.3 NP flag

The NP flag is a status flag that indicates that non-maskable interrupt (NMI) servicing is under execution. This flag is set when an NMI interrupt request has been acknowledged, and masks all interrupt requests to prohibit multiple interrupts from being acknowledged.

Figure 5-4. NP Flag (NP)

After reset: 00000020H

Symbol	31	8	7	6	5	4	3	2	1	0
PSW	0		**NP**	EP	ID	SAT	CY	OV	S	Z

NP	NMI Servicing State
0	No NMI interrupt servicing
1	NMI interrupt currently being serviced

★

5.2.4 Noise elimination of external interrupt request input pin

(1) Noise elimination of NMI and INTP0 to INTP3 pins

The noise of the NMI pin and INTP0 to INTP3 pins is eliminated by the noise eliminator using analog delay. Therefore, signals input to the NMI and INTP0 to INTP3 pins are not detected as an edge, unless they maintain their input level for a certain period. The edge is detected after a certain period has elapsed.

The NMI and INTP0 to INTP3 pins can be used for releasing the software STOP mode. In the software STOP mode, the system clock is not used for noise elimination because the internal system clock is stopped.

(2) Noise elimination of INTP4 to INTP6 pins

The INTP4 to INTP6 pins incorporate a digital noise eliminator. If the input level of the INTP pin is detected by the sampling clock (fxx) and the same level is not detected three successive times, the input pulse is eliminated as a noise. In the software STOP mode, the INTP4 to INTP6 pins cannot be used for releasing the software STOP mode because the internal system clock is stopped. Note the following.

- If the input pulse width is between 2 and 3 clocks, whether the input pulse is detected as a valid edge or eliminated as noise is undefined. To securely detect the level as a valid edge, the same level input of 3 clocks or more is required.
- When noise is generated in synchronization with the sampling clock, this may not be recognized as noise. In this case, eliminate the noise by adding a filter to the input pin.

★ 5.2.5 Edge detection function of external interrupt request input pin

The NMI pin valid edge can be selected from the following four types: falling edge, rising edge, both edges, or neither edge.

Rsing edge specification register 0 (EGP0) and falling edge specification register 0 (EGN0) specify the valid edge of the external interrupt. These two registers can be read/written in 1-bit or 8-bit units.

After reset, the valid edge of the external interrupt request input pin is set to the “detect neither rising nor falling edge” state. Therefore, the NMI pin functions as a normal port and an interrupt request cannot be acknowledged, unless a valid edge is specified by using the EGP0 and EGN0 registers.

When using the P00 pin as an output port, set the NMI pin valid edge to “detect neither rising nor falling edge”. When using the P01 to P07 pins as an output port, set the valid edges of the INTP0 to INTP6 pins to “detect neither rising nor falling edge” or mask the interrupt requests.

(1) Rising edge specification register 0 (EGP0)

After reset: 00H R/W Address: FFFFF0C0H

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP07	EGP06	EGP05	EGP04	EGP03	EGP02	EGP01	EGP00

EGP0n	Rising Edge Valid Control
0	No interrupt request signal occurred at the rising edge
1	Interrupt request signal occurred at the rising edge

n = 0: NMI pin control

n = 1 to 7: INTP0 to INTP6 pin control

(2) Falling edge specification register 0 (EGN0)

After reset: 00H R/W Address: FFFFF0C2H

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN07	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00

EGN0n	Falling Edge Valid Control
0	No interrupt request signal occurred at the falling edge
1	Interrupt request signal occurred at the falling edge

n = 0: NMI pin control

n = 1 to 7: INTP0 to INTP6 pin control

5.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850/SA1 has 30 maskable interrupt sources (refer to **5.1.1 Features**).

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, allowing programmable priority control.

When an interrupt request has been acknowledged, the acknowledgement of other maskable interrupts is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt servicing routine, the interrupt enabled (EI) status is set which enables interrupts having a higher priority to immediately interrupt the current service routine in progress. Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To use multiple interrupts, it is necessary to save EIPC and EIPSW to memory or a register before executing the EI instruction, and restore EIPC and EIPSW to the original values by executing the DI instruction before the RETI instruction.

When the WDTM4 bit of the watchdog timer mode register (WDTM) is set to 0, the watchdog timer overflow interrupt functions as a maskable interrupt (INTWDTM).

5.3.1 Operation

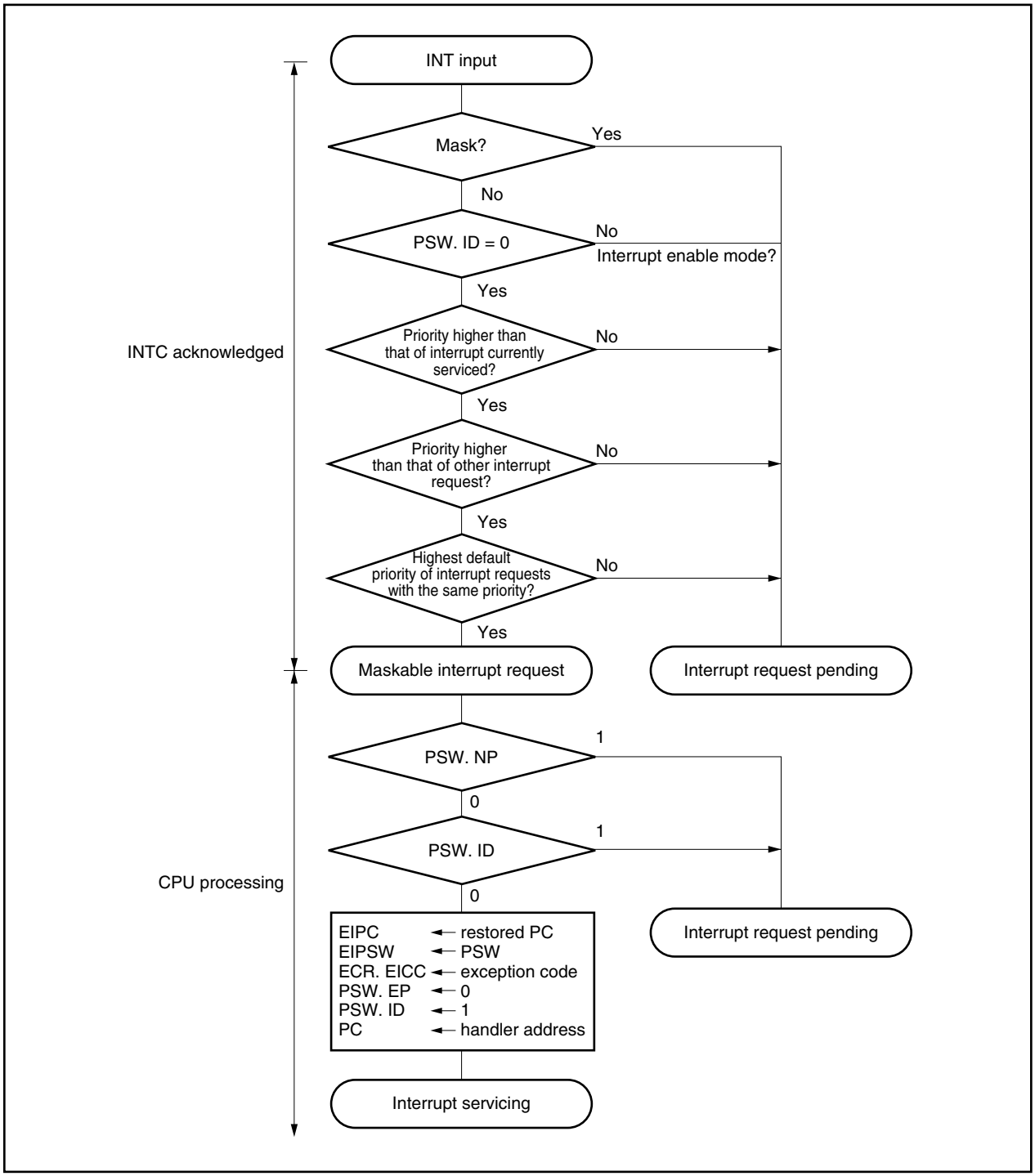
If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine.

<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> Loads the corresponding handler address to the PC, and transfers control.

The INT input masked by INTC and the INT input that occurs during the other interrupt servicing (when PSW.NP = 1 or PSW.ID = 1) are internally held pending. When the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 by using the RETI and LDSR instructions, the pending INT is input to start the new maskable interrupt servicing.

How the maskable interrupts are serviced is shown below.

Figure 5-5. Maskable Interrupt Servicing



5.3.2 Restore

To restore execution from maskable interrupt servicing, the RETI instruction is used.

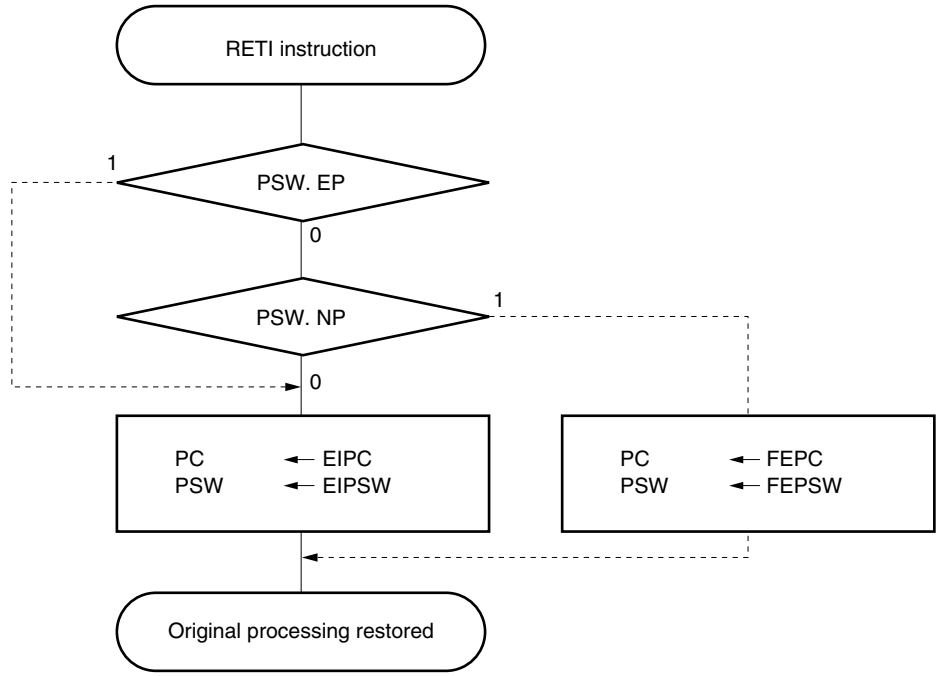
Operation of RETI instruction

When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

(1) Restores the values of the 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 processing of the RETI instruction is shown below.

Figure 5-6. RETI Instruction Processing



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.

5.3.3 Priorities of maskable interrupts

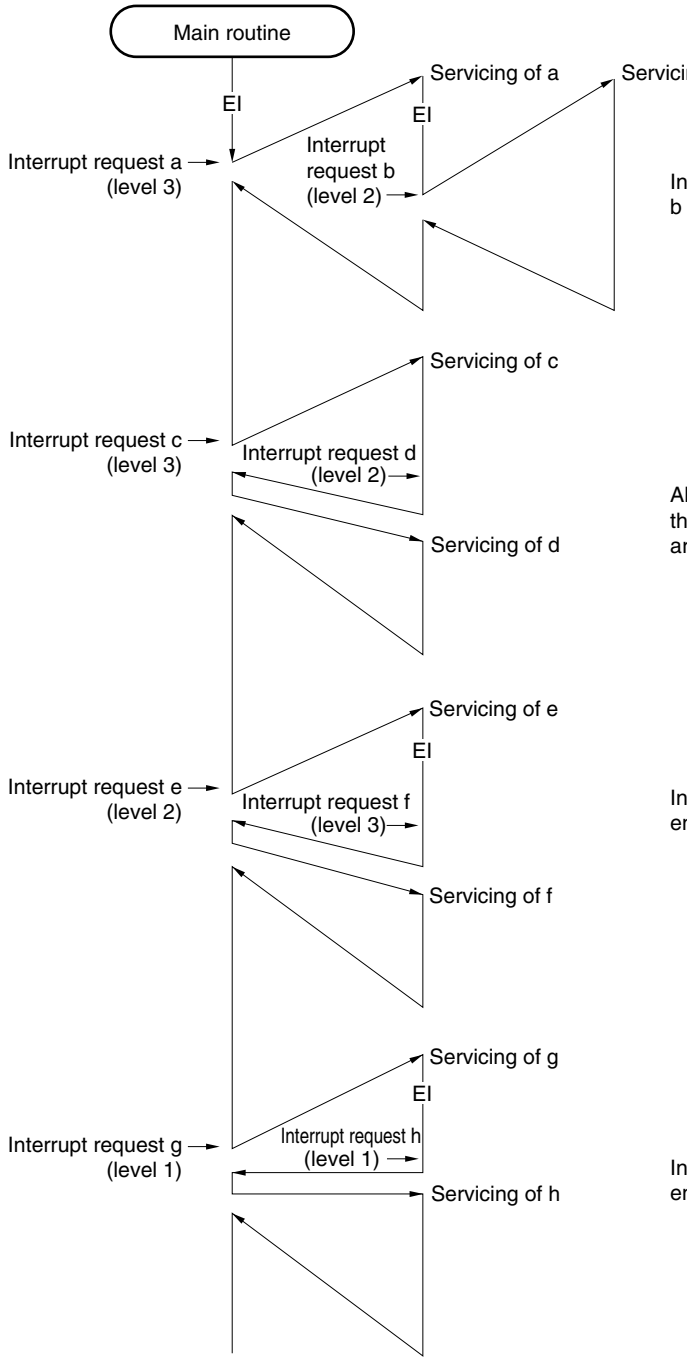
The V850/SA1 provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels which are specified by the interrupt priority level specification bit (xxPRn). When two or more interrupts having the same priority level specified by xxPRn 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 5-1 Interrupt Source List**. 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 the 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 into the interrupt servicing program) to set the interrupt enabled mode.

Remark xx: Identification name of each peripheral unit (refer to **Table 5-2**)
n: Number of each peripheral unit (refer to **Table 5-2**)

Figure 5-7. Example of Multiple Interrupt Servicing (1/2)

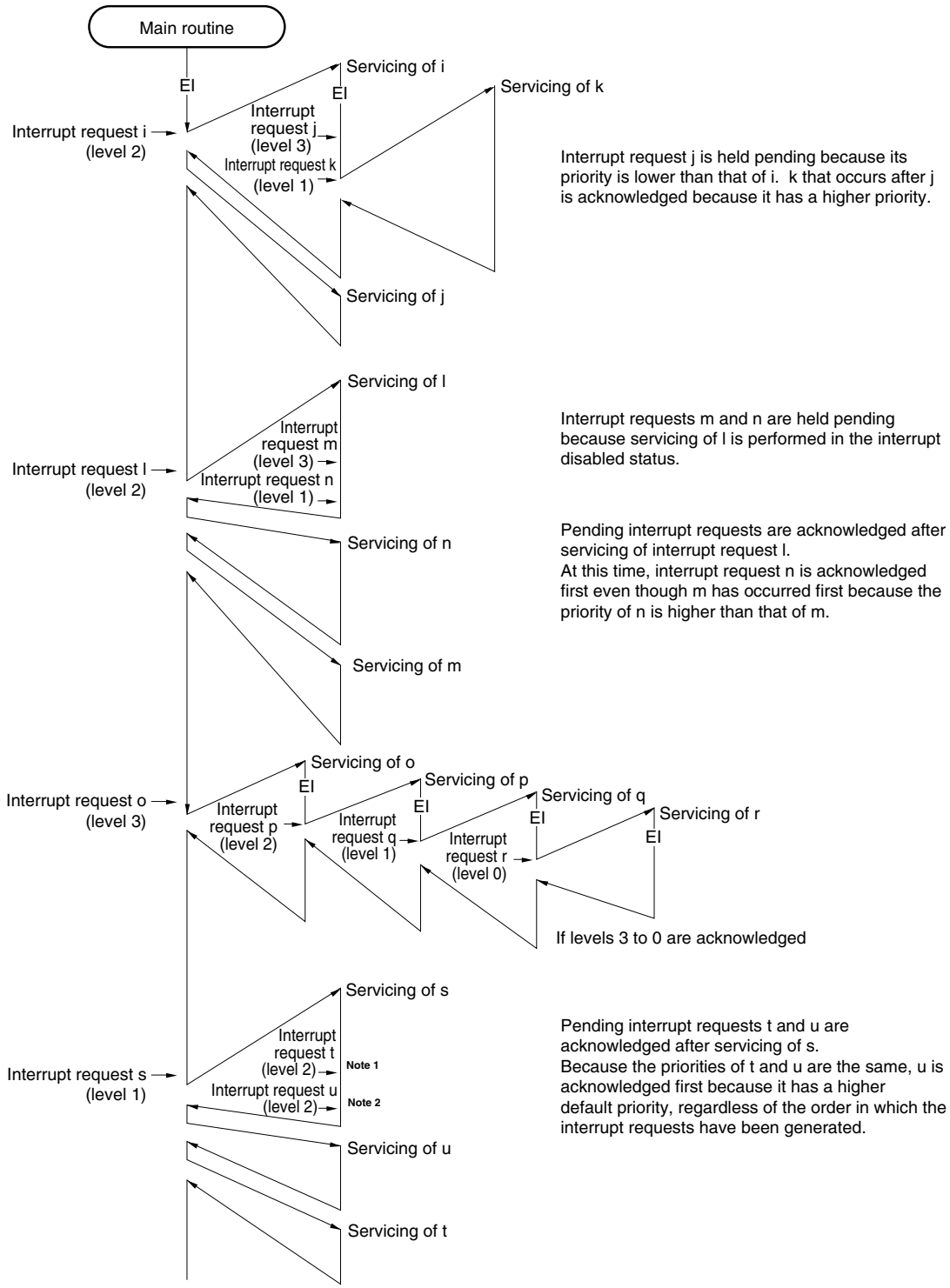


Caution The values of EIPC and EIPSW must be saved before executing multiple interrupts.

Remarks 1. a to u in the figure are the names of interrupt requests shown for the sake of explanation.

2. The default priority in the above figure indicates the relative priority between two interrupt requests.

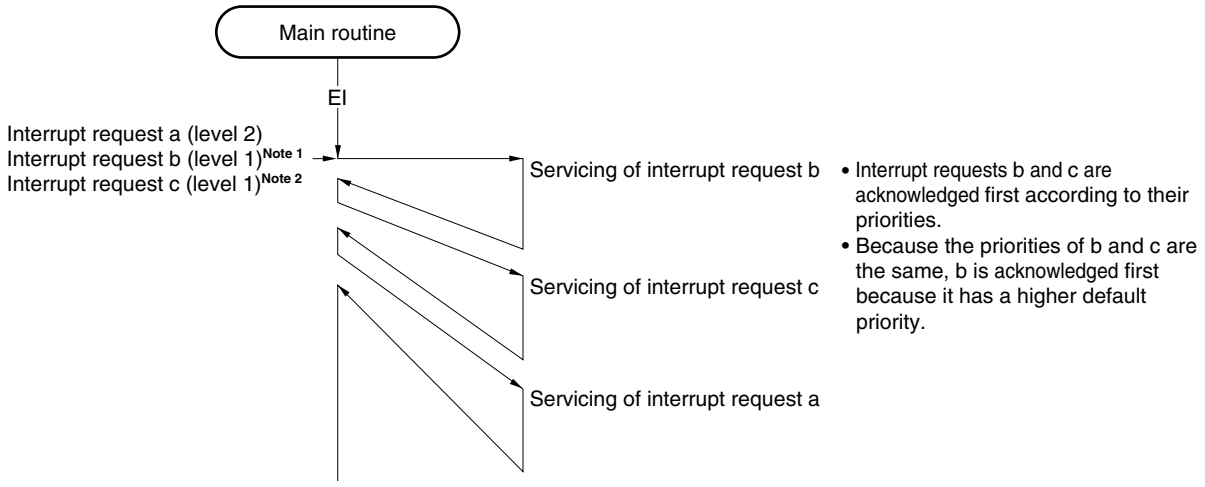
Figure 5-7. Example of Multiple Interrupt Servicing (2/2)



Notes 1. Lower default priority

2. Higher default priority

Figure 5-8. Example of Servicing Interrupt Requests Generated Simultaneously



Notes 1. Higher default priority

2. Lower default priority

Remarks 1. a, b, and c in the above figure are the names of interrupt requests shown for the sake of explanation.

2. The default priority in the above figure indicates the relative priority between two interrupt requests.

5.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each maskable interrupt and sets the control conditions for each maskable interrupt request. The interrupt control register can be read/written in 8-bit or 1-bit units.

Cautions 1. If the following three conditions conflict, interrupt servicing will be performed twice. However, interrupt servicing is not performed twice if DMA is not being used.

- **Execution of bit manipulation instruction for interrupt request flag (xxIFn)**
- **Interrupt request generated by the same hardware interrupt control register (xxICn) as the interrupt request flag (xxIFn)**
- **DMA activated during execution of bit manipulation instruction for interrupt request flag (xxIFn)**

Two software-based countermeasures are shown below.

- **Insert the DI and EI instructions before and after (respectively) the software bit manipulation instruction to avoid jumping to an interrupt immediately after execution of the bit manipulation instruction.**
- **Because interrupts are disabled (DI state) by hardware after an interrupt request has been acknowledged, clear the interrupt request flag (xxIFn) before executing the EI instruction in each interrupt servicing routine.**

★ **2. Read the xxIFn bit of the xxICn register with interrupts disabled. When the xxIFn bit is read with interrupts enabled, a normal value may not be read if the interrupt acknowledgement timing and the bit reading timing conflict.**

After reset: 47H R/W Address: FFFFF100H to FFFFF13AH

Symbol	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 generated
1	Interrupt request generated

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 Automatically reset by hardware when an interrupt request is acknowledged.

Remark xx: Identification name of each peripheral unit (refer to **Table 5-2**)
n: Number of each peripheral unit (refer to **Table 5-2**)

The address and bits of each interrupt control register are as follows.

Table 5-2. Interrupt Control Register (xxICn)

Address	Register	Bit							
		7	6	5	4	3	2	1	0
FFFFF100H	WDTIC	WDTIF	WDTMK	0	0	0	WDTPR2	WDTPR1	WDTPR0
FFFFF102H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF104H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF106H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF108H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF10AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF10CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF10EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF110H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF112H	TMIC00	TMIF00	TMMK00	0	0	0	TMPR002	TMPR001	TMPR000
FFFFF114H	TMIC01	TMIF01	TMMK01	0	0	0	TMPR012	TMPR011	TMPR010
FFFFF116H	TMIC10	TMIF10	TMMK10	0	0	0	TMPR102	TMPR101	TMPR100
FFFFF118H	TMIC11	TMIF11	TMMK11	0	0	0	TMPR112	TMPR111	TMPR110
FFFFF11AH	TMIC2	TMIF2	TMMK2	0	0	0	TMPR22	TMPR21	TMPR20
FFFFF11CH	TMIC3	TMIF3	TMMK3	0	0	0	TMPR32	TMPR31	TMPR30
FFFFF11EH	TMIC4	TMIF4	TMMK4	0	0	0	TMPR42	TMPR41	TMPR40
FFFFF120H	TMIC5	TMIF5	TMMK5	0	0	0	TMPR52	TMPR51	TMPR50
FFFFF122H	CSIC0	CSIF0	CSMK0	0	0	0	CSPR02	CSPR01	CSPR00
FFFFF124H	SERIC0	SERIF0	SERMK0	0	0	0	SERPR02	SERPR01	SERPR00
FFFFF126H	CSIC1	CSIF1	CSMK1	0	0	0	CSPR12	CSPR11	CSPR10
FFFFF128H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF12AH	CSIC2	CSIF2	CSMK2	0	0	0	CSPR22	CSPR21	CSPR20
FFFFF12CH	SERIC1	SERIF1	SERMK1	0	0	0	SERPR12	SERPR11	SERPR10
FFFFF12EH	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF130H	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF132H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF134H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF136H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF138H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF13AH	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0

5.3.5 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 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 reset to 0 by hardware. However, it is not reset when execution is returned from non-maskable interrupt processing or exception processing.

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

★ **Caution Read the ISPR register with interrupts disabled. When the ISPR register is read with interrupts enabled, a normal value may not be read if the interrupt acknowledgement timing and the bit reading timing conflict.**

After reset: 00H R Address: FFFFF166H

Symbol	7	6	5	4	3	2	1	0
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Indicates 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)

5.3.6 ID flag

The interrupt disable status flag (ID) of the PSW controls the enabling and disabling of maskable interrupt requests. As a status flag, it also displays the current maskable interrupt acknowledgment status.

Figure 5-9. Interrupt Disable Flag (ID)

After reset: 00000020H

Symbol	31	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 acknowledgement enabled
1	Maskable interrupt request acknowledgement disabled

Note Interrupt disable flag (ID) function

It 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 request is acknowledged, the ID flag is automatically set to 1 by hardware.

The interrupt request generated during the acknowledgement disabled period (ID = 1) can be acknowledged when the xxIFn bit of xxICn is set to 1, and the ID flag is reset to 0.

Remark xx: Identification name of each peripheral unit (refer to **Table 5-2**)

n: Number of each peripheral unit (refer to **Table 5-2**)

5.3.7 Watchdog timer mode register (WDTM)

This register can be read/written in 8-bit or 1-bit units (for details, refer to **CHAPTER 9 WATCHDOG TIMER**).

After reset: 00H R/W Address: FFFFF384H

Symbol	7	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	0	0	0	0

RUN	Watchdog Timer Operation Control
0	Count operation stopped
1	Count start after clearing

WDTM4	Timer Mode Selection/Interrupt Control by WDT
0	Interval timer mode
1	WDT mode

Caution If the RUN or WDTM4 bit is set to 1, it cannot be cleared other than by reset input.

5.4 Software Exceptions

A software exception is generated when the CPU executes the TRAP instruction, and can be always acknowledged.

- TRAP instruction format: TRAP vector (where vector is 0 to 1FH)

For details of the instruction function, refer to the **V850 Series Architecture User's Manual.**

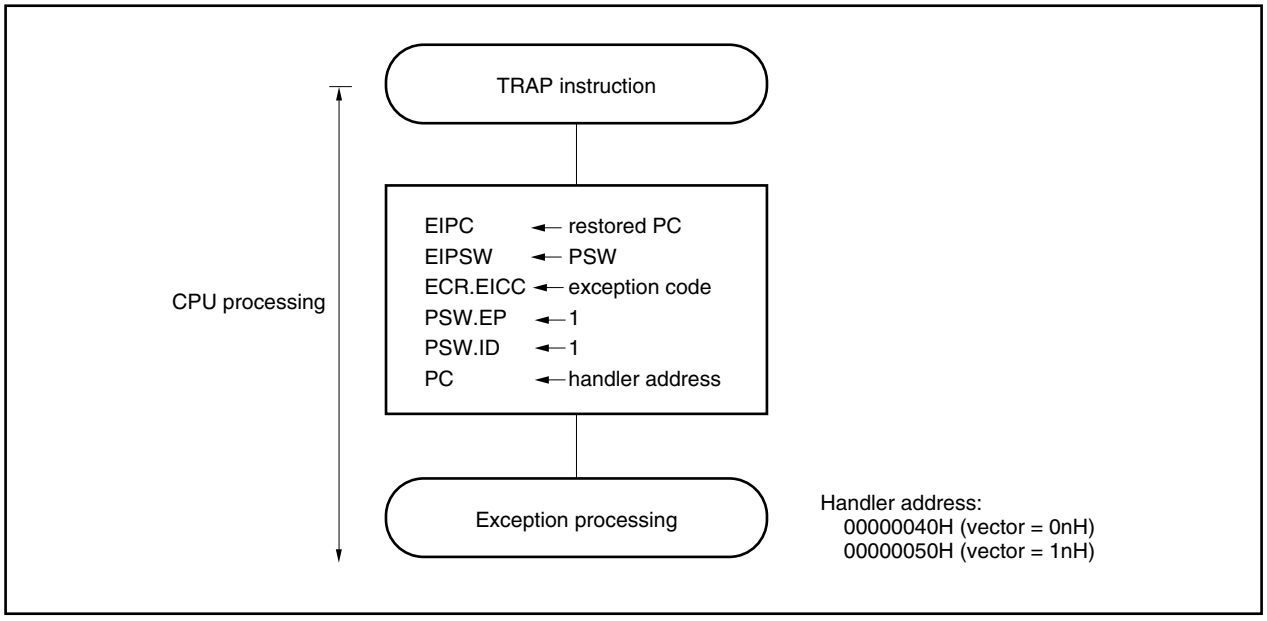
5.4.1 Operation

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 PSW.
(5) Loads the handler address (00000040H or 00000050H) of the software exception routine in the PC, and transfers control.

How a software exception is processed is shown below.

Figure 5-10. Software Exception Processing



5.4.2 Restore

To restore or return execution from the software exception service routine, the RETI instruction is used.

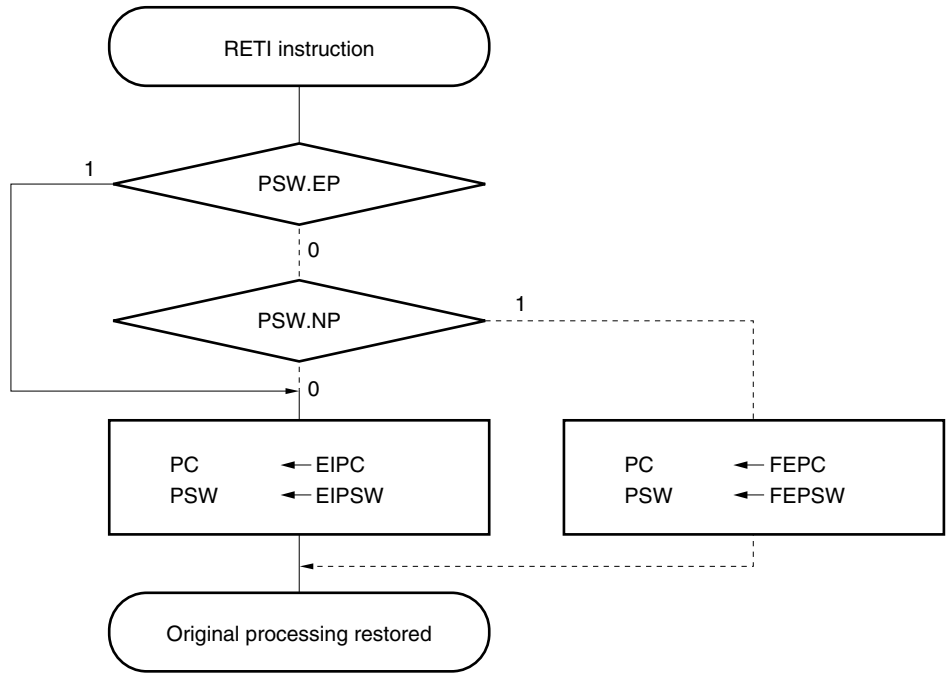
Operation of RETI instruction

When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

(1) Restores 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 processing of the RETI instruction is shown below.

Figure 5-11. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception process, 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.

5.4.3 EP flag

The EP flag in the PSW is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

Figure 5-12. EP Flag (EP)

After reset: 00000020H

Symbol	31 … 8	7	6	5	4	3	2	1	0
PSW	0	NP	**EP**	ID	SAT	CY	OV	S	Z

EP	Exception Processing
0	Exception processing is not in progress
1	Exception processing is in progress

5.5 Exception Trap

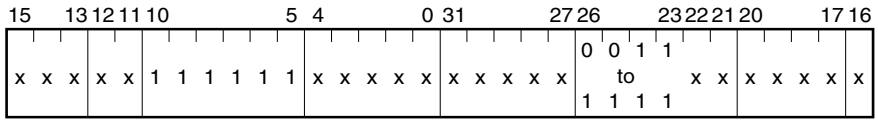
The exception trap is an interrupt that is requested when illegal execution of an instruction takes place. In the V850/SA1, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

- Illegal opcode exception: Occurs if the sub opcode field of the instruction to be executed next is not a valid opcode.

5.5.1 Illegal opcode definition

An illegal opcode is defined to be a 32-bit word with bits 5 to 10 being 111111B and bits 23 to 26 being 0011B to 1111B.

Figure 5-13. Illegal Opcode



x: don't care

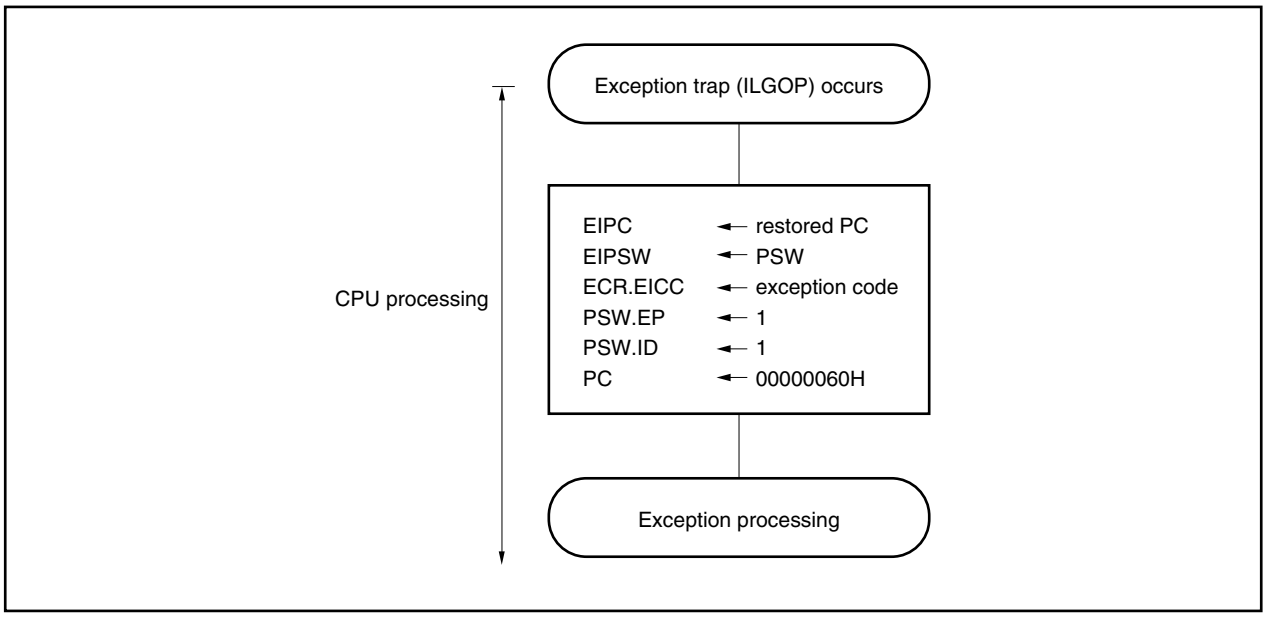
5.5.2 Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code (0060H) to the lower 16 bits (EICC) of ECR.
(4) Sets the EP and ID bits of the PSW.
(5) Loads the handler address (00000060H) for the exception trap routine to the PC, and transfers control.

How the exception trap is processed is shown below.

Figure 5-14. Exception Trap Processing



5.5.3 Restore

To restore or return execution from the exception trap, the RETI instruction is used.

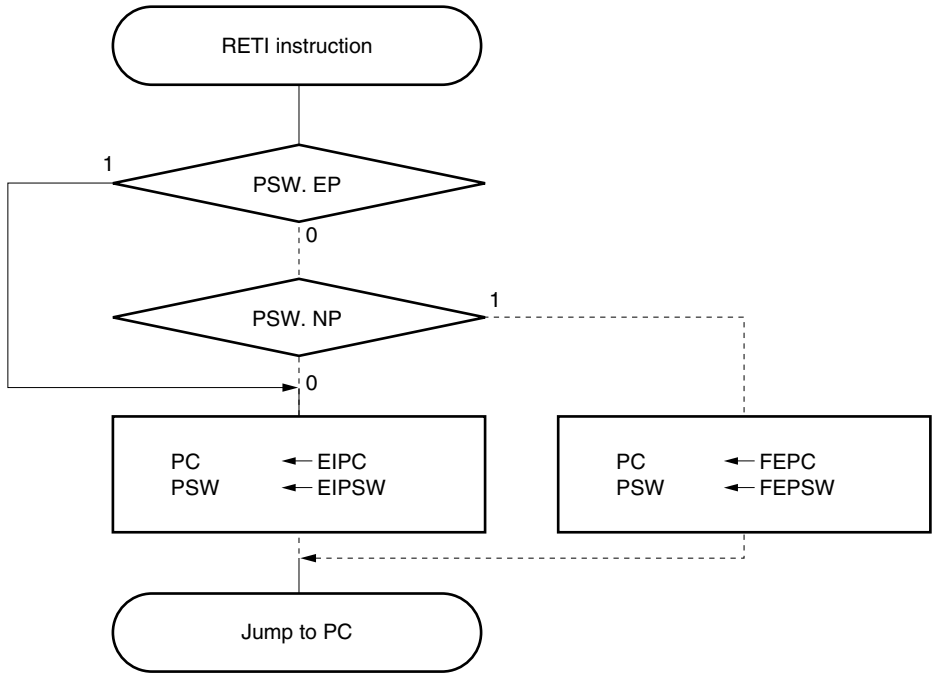
Operation of RETI instruction

When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

(1) Restores 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 processing of the RETI instruction is shown below.

Figure 5-15. RETI Instruction Processing



Caution When the PSW.EP and the PSW.NP bit are changed by the LDSR instruction during the exception trap process, 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.

5.6 Priority Control

5.6.1 Priorities of interrupts and exceptions

Table 5-3. Priorities of Interrupts and Exceptions

	RESET	NMI	INT	TRAP	ILGOP
RESET		*	*	*	*
NMI	×		←	←	←
INT	×	↑		←	←
TRAP	×	↑	↑		←
ILGOP	×	↑	↑	↑	

RESET: Reset
NMI: Non-maskable interrupt
INT: Maskable interrupt
TRAP: Software exception
ILGOP: Illegal opcode exception
*: The item on the left ignores the item above.
×: The item on the left is ignored by the item above.
↑: The item above is higher than the item on the left in priority.
←: The item on the left is higher than the item above in priority.

5.6.2 Multiple interrupts

Multiple interrupt servicing is a function that allows the nesting of interrupts. If a higher priority interrupt is generated and acknowledged, it will be allowed to stop a current interrupt servicing routine in progress. Execution of the original routine will resume once the higher priority interrupt routine is completed.

If an interrupt with a lower or equal priority is generated and a service routine is currently in progress, the later interrupt will be held pending.

Multiple interrupt servicing control is performed when in the interrupt enabled state (ID = 0). Even in an interrupt servicing routine, the interrupt enabled state (ID = 0) must be set.

If a maskable interrupts are enabled or an exception is generated during a service program of a maskable interrupt or exception, EIPC and EIPSW must be saved.

The following example shows the procedure of interrupt nesting.

(1) To acknowledge maskable interrupts in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• Save EIPC to memory or register
• Save EIPSW to memory or register
• EI instruction (enables interrupt acknowledgement)
        ...
        ...
• DI instruction (disables interrupt acknowledgement)
• Restore saved value to EIPSW
• Restore saved value to EIPC
• RETI instruction
```

← Acknowledgement of interrupt such as INTP input.

(2) To generate exception in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• Save EIPC to memory or register
• Save EIPSW to memory or register
• EI instruction (enables interrupt acknowledgement)
        ...
• TRAP instruction
• Illegal opcode
        ...
• Restore saved value to EIPSW
• Restore saved value to EIPC
• RETI instruction
```

← Acknowledgement of exception such as TRAP instruction.

← Acknowledgement of exception such as illegal opcode.

Priorities 0 to 7 (0 is the highest) can be programmed for each maskable interrupt request for multiple interrupt servicing control. To set a priority level, write values to the xxPRn0 to xxPRn2 bits of the interrupt request control register (xxICn) corresponding to each maskable interrupt request. At reset, the interrupt request is masked by the xxMKn bit, and the priority level is set to 7 by the xxPRn0 to xxPRn2 bits.

Remark xx: Identification name of each peripheral unit (refer to **Table 5-2**)
n: Number of each peripheral unit (refer to **Table 5-2**)

Priorities of maskable interrupts

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 > Level 5 > Level 6 > Level 7 (Low)

Interrupt servicing that has been suspended as a result of multiple interrupt servicing is resumed after the interrupt servicing of the higher priority has been completed and the RETI instruction has been executed.

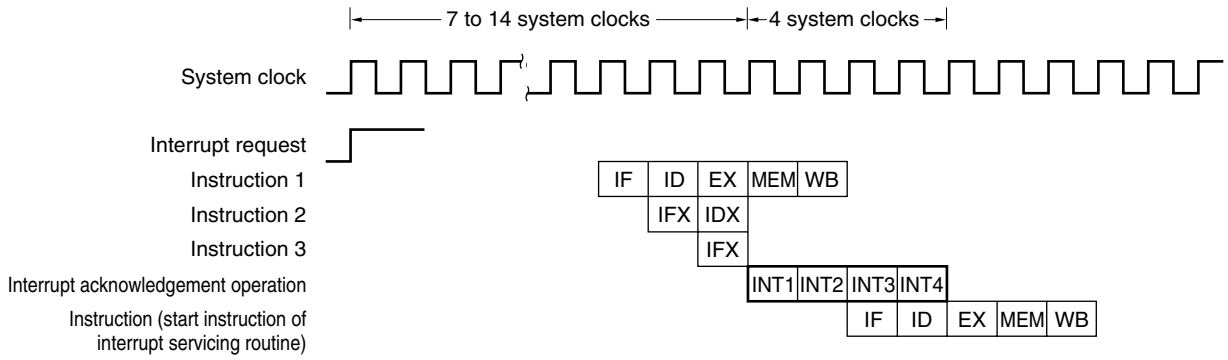
A pending interrupt request is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

Caution In the non-maskable interrupt servicing routine (time until the RETI instruction is executed), maskable interrupts are not acknowledged but are suspended.

5.7 Interrupt Latency Time

The following table describes the V850/SA1 interrupt latency time (from interrupt request generation to start of interrupt servicing).

Figure 5-16. Pipeline Operation at Interrupt Request Acknowledgement



INT1 to INT4: Interrupt acknowledgement processing
IFx: Invalid instruction fetch
IDx: Invalid instruction decode

Interrupt Latency Time (System Clock)			Condition
	Internal Interrupt	External Interrupt	
Minimum	11	13	Time to eliminate noise (2 system clocks) is also necessary for external interrupts, except when: • In IDLE/software STOP mode • External bus is accessed • Two or more interrupt request non-sample instructions are executed in succession • An interrupt control register is accessed
Maximum	18	20	

5.8 Periods in Which Interrupts Are Not Acknowledged

An interrupt is acknowledged while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction.

Interrupt request non-sample instructions

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (vs. PSW)

⋆ 5.8.1 Interrupt request valid timing after EI instruction

When an interrupt request signal is generated (IF flag = 1) in the status in which the DI instruction is executed (interrupts disabled) and interrupts are not masked (MK flag = 0), seven system clocks are required from the execution of the EI instruction (interrupts enabled) to the interrupt request acknowledgement by the CPU. The CPU does not acknowledge interrupt requests if the DI instruction (interrupts disabled) is executed during the seven system clocks.

Therefore, seven system clocks worth of instruction execution clocks must be inserted after the EI instruction (interrupts enabled). However, under the following conditions, interrupt requests cannot be acknowledged even if the seven system clocks are secured, so securing under the following conditions is prohibited.

- In IDLE/software STOP mode
- An interrupt request non-sampling instruction (instruction to manipulate the PSW.ID bit) is executed
- An interrupt request control register (xxICn) is accessed

The following shows an example of program processing.

[Program processing example]

```
      DI
      :             ; (MK flag = 0)
      :             ; ← Interrupt request occurs (IF flag = 1)
      EI            ; EI instruction executed
      NOP           ; 1 system clock                          )
      NOP           ; 1 system clock                          |
      NOP           ; 1 system clock                          } Note
      NOP           ; 1 system clock                          |
      JR    LP1     ; 3 system clocks (branch to LP1 routine) )
      :
LP1   :             ; LPI routine
      DI            ; After EI instruction execution, NOP instruction is
                      executed four times, and DI
                      instruction is executed at the eighth clock by JR instruction
```

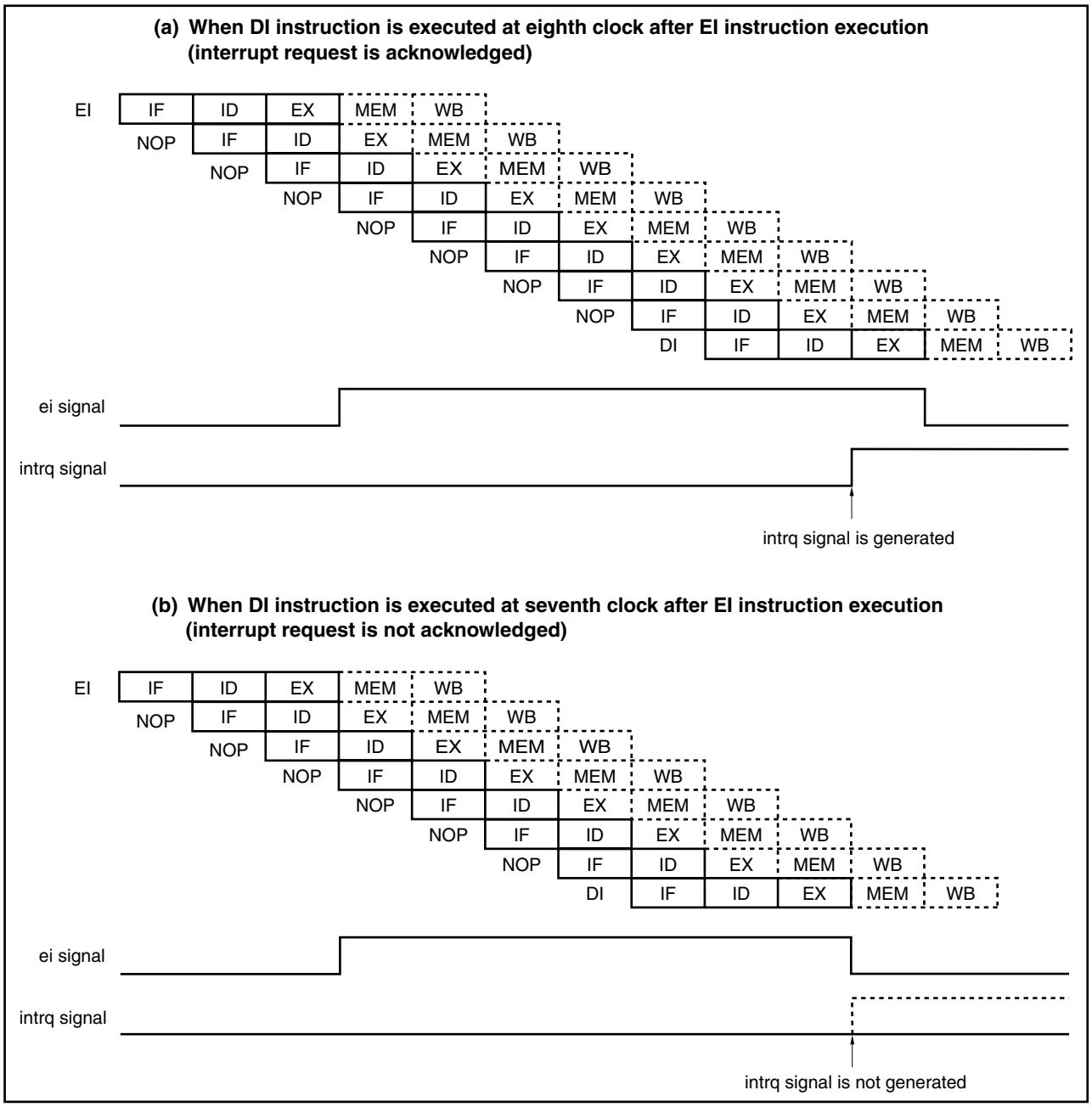
Note Do not execute the DI instruction (PSW.ID = 1) during this period.

Remarks 1. In this example, the DI instruction is executed at the eighth clock after execution of the EI instruction, so the CPU acknowledges an interrupt request signal and performs interrupt servicing.

2. The interrupt servicing routine instructions are not executed at the eighth clock after the EI instruction execution. The interrupt servicing routine instructions are executed the four system clocks after the CPU acknowledges the interrupt request signal.

3. This example shows the case in which an interrupt request signal is generated (IF flag = 1) before the EI instruction is executed. If an interrupt request signal is generated after the EI instruction is executed, the CPU does not acknowledge the interrupt request signal if interrupts are disabled (PSW.ID = 1) for seven clocks after the IF flag is set (1).

Figure 5-17. Pipeline Flow and Interrupt Request Signal Generation Timing



★ 5.9 Interrupt Control Register Bit Manipulation Instructions During DMA Transfer

To manipulate the bits of the interrupt control register (xxICn) in the EI state when using the DMA function, execute the DI instruction before manipulation and EI instruction after manipulation. Alternatively, clear (0) the xxIF bit at the start of the interrupt servicing routine.

When not using the DMA function, these manipulations are not necessary.

Remark xx: Peripheral unit identification name (see **Table 5-2**)
N: Peripheral unit number (see **Table 5-2**)

CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 General

The clock generator is a circuit that generates the clock pulses that are supplied to the CPU and peripheral hardware. There are two types of clock oscillators.

(1) Main clock oscillator

This oscillator has an oscillation frequency of 2 to 20 MHz. Oscillation can be stopped by setting the software STOP mode or by setting the processor clock control register (PCC). Oscillation is also stopped during a reset.

★ External clocks can be directly input. At this time, input a clock signal only to the X1 pin and leave the X2 pin open.

Cautions 1. When the main clock oscillator is stopped by inputting a reset or setting the software STOP mode, the oscillation stabilization time is secured after the stop mode is released. This oscillation stabilization time is set via the oscillation stabilization time select register (OSTS). The watchdog timer is used as the timer that counts the oscillation stabilization time.

2. If stoppage of the main clock is released by clearing MCK to 0 after the main clock is stopped by setting the MCK bit in the PCC register to 1, the oscillation stabilization time is not secured.

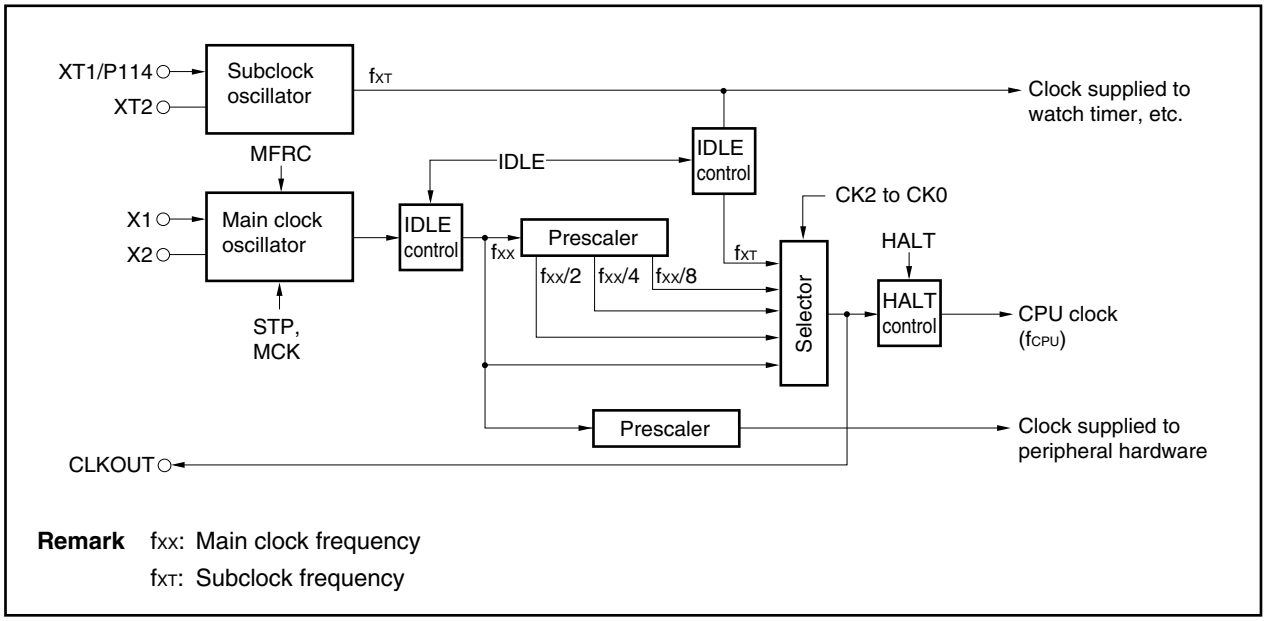
(2) Subclock oscillator

This circuit has an oscillation frequency of 32.768 kHz. Its oscillation is not stopped when the software STOP mode is set, neither is it stopped when a reset is input. To stop oscillation, connect the XT1 pin to V_{SS}.

★ External clocks can be directly input. At this time, input a clock signal to the XT1 pin and input its inverted signal to the XT2 pin.

6.2 Configuration

★ **Figure 6-1. Clock Generator**



Remark f_{XX}: Main clock frequency
f_{XT}: Subclock frequency

6.3 Clock Output Function

This function outputs the CPU clock via the CLKOUT pin.

When clock output is enabled, the CPU clock is output via the CLKOUT pin. When it is disabled, a low-level signal is output via the CLKOUT pin.

Output is stopped in the IDLE or software STOP mode (fixed to low level).

This function is controlled via the DCLK1 and DCLK0 bits in the PSC register.

The high-impedance status is set during the reset period. After reset is released, a low level is output.

Caution While CLKOUT is being output, do not change the CPU clock (CK2 to CK0 bits of PCC register).

6.3.1 Control registers

(1) Processor clock control register (PCC)

This is a specific register. It can be written to only when a specified combination of sequences is used (see **3.4.9 Specific registers**). This register can be read/written in 8-bit or 1-bit units.

After reset: 03H R/W Address: FFFFF074H

	7	6	5	4	3	2	1	0
PCC	0	MCK	MFRC	0	0	CK2	CK1	CK0

MCK	Operation of Main Clock
0	Operating
1	Stopped

MFRC	Selection of Internal Feedback Resistor for Main Clock
0	Use
1	Do not use

CK2[Note1, 2]	CK1	CK0	Selection of CPU Clock
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	X	X	f_{XT} (subclock)

Notes 1. If manipulating CK2, do so in 1-bit units. In the case of 8-bit manipulation, do not change the values of CK1 and CK0.

★ **2.** When the CPU operates on the subclock (CK2 = 1), do not set the software STOP mode.

Cautions 1. While CLKOUT is being output, do not change the CPU clock (the value of the CK2 to CK0 in the PCC register).

2. Even if the MCK bit is set to 1 during main clock operation, the main clock is not stopped. The CPU clock stops after the subclock is selected.

3. Always set bits 3, 4, and 7 to 0.

4. Leakage current can be reduced by avoiding the use of the main clock's on-chip feedback resistor (MFRC = 1) while the external clock is operating. However, the leakage current cannot be reduced during the oscillation stabilization time,.

Remark X: don't care

(a) Example of main clock operation → subclock operation setup

<1> CK2 ← 1: Bit manipulation instructions are recommended. Do not change CK1 and CK0.

<2> Subclock operation: The maximum number of the following instructions is required before subclock operation after the CK2 bit is set.
(CPU clock frequency before setting/subclock frequency) × 2
Therefore, insert the wait described above using a program.

<3> MCK ← 1: Only when the main clock is stopped.

★ **(b) Example of subclock operation → main clock operation setup**

<1> MCK ← 0: Main clock oscillation start

<2> Insert a wait using a program and wait until the main clock oscillation stabilization time elapses.

<3> CK2 ← 0: Bit manipulation instructions are recommended. Do not change CK1 and CK0.

<4> Main clock operation: At least two instructions are required before main clock operation after the CK2 bit is set.

(2) Power save control register (PSC)

This is a specific register. It can be written to only when a specified combination of sequences is used (see **3.4.9 Specific registers**).

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

After reset: C0H R/W Address: FFFFF070H

	7	6	5	4	3	2	1	0
PSC	DCLK1	DCLK0	0	0	0	IDLE	STP	0

DCLK1	DCLK0	Specification of CLKOUT Pin Operation
0	0	Output enabled
0	1	Setting prohibited
1	0	Setting prohibited
1	1	Output disabled (when reset)

IDLE	IDLE Mode Setting
0	Normal mode
1	IDLE mode[Note 1]

STP	Software STOP Mode Setting
0	Normal mode
1	Software STOP mode[Notes 2, 3]

Notes 1. When IDLE mode is released, this bit is automatically reset to 0.

2. When software STOP mode is released, this bit is automatically reset to 0.

★ **3.** When the CPU operates on the subclock (CK2 = 1), do not set the STP bit to 1.

Cautions 1. The DCLK0 and DCLK1 bits should be manipulated in 8-bit units.

★ **2. Do not set (1) the IDLE bit and STP bit at the same time. If they are set simultaneously, the software STOP mode is entered.**

3. Be sure to set bits 3 to 5 to 0.

(3) Oscillation stabilization time select register (OSTS)

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

After reset: 04H R/W Address: FFFFF380H

	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[Note]
0	0	0	$2^{14}/f_{XX}$ (819.2 μs)
0	0	1	$2^{16}/f_{XX}$ (3.3 ms)
0	1	0	$2^{17}/f_{XX}$ (6.6 ms)
0	1	1	$2^{18}/f_{XX}$ (13.1 ms)
1	0	0	$2^{19}/f_{XX}$ (26.2 ms)
Other than above			Setting prohibited

Note The numerical value in parentheses is the value when f_{XX} = 20 MHz.

6.4 Power Save Functions

6.4.1 General

This product provides the following power saving functions.

These modes can be combined and switched to suit the target application, which enables effective implementation of low-power systems.

(1) HALT mode

When in this mode, the clock's oscillator continues to operate but the CPU's operating clock is stopped. A clock continues to be supplied for other on-chip peripheral functions to maintain operation of those functions. This enables the system's total power consumption to be reduced.

A dedicated instruction (the HALT instruction) is used to switch to HALT mode.

(2) IDLE mode

This mode stops the entire system by stopping the CPU's operating clock as well as the operating clock for on-chip peripheral functions other than for the watch timer while the clock oscillator is still operating. However, the subclock continues to operate and supplies a clock to the on-chip peripheral functions.

When this mode is released, there is no need for the oscillator to wait for the oscillation stabilization time, so normal operation can be resumed quickly.

When the IDLE bit of the power save control register (PSC) is set to 1, the system switches to IDLE mode.

(3) Software STOP mode

This mode stops the entire system by stopping the main clock oscillator. The subclock continues to be supplied to keep on-chip peripheral functions operating. If a subclock is not used, ultra low power consumption mode (current that flows through the on-chip feedback resistor of the subclock oscillator and leakage current only are flowing) is set. Software STOP mode setting is prohibited if the CPU is operating via the subclock.

If the STP bit of the PSC register is set to 1, the system enters software STOP mode.

(4) Subclock operation

In this mode, the CPU clock is set to operate using the subclock and the MCK bit of the PCC register is set to 1 to set low power consumption mode in which the entire system operates using only the subclock.

When HALT mode is set, the CPU's operating clock is stopped so that power consumption can be reduced.

When IDLE mode is set, the CPU's operating clock and some peripheral functions (DMAC and BCU) are stopped, so that power consumption can be reduced even more than in HALT mode.

6.4.2 HALT mode

(1) Settings and operating states

In this mode, the clock's oscillator continues to operate but the CPU's operating clock is stopped. A clock continues to be supplied for other on-chip peripheral functions to maintain operation of those functions. When HALT mode is set while the CPU is idle, it enables the system's total power consumption to be reduced.

In HALT mode, execution of programs is stopped but the contents of all registers and internal RAM are retained as they were just before HALT mode was set. In addition, all on-chip peripheral functions that do not depend on instruction processing by the CPU continue operating.

HALT mode can be set by executing the HALT instruction. It can be set when the CPU is operating via either the main clock or subclock.

The operating statuses in the HALT mode are listed in Table 6-1.

(2) Release of HALT mode

HALT mode can be released by an NMI request, an unmasked maskable interrupt request, or $\overline{\text{RESET}}$ input.

(a) Release by interrupt request

HALT mode is released regardless of the priority level when an NMI request or an unmasked maskable interrupt request occurs. However, the following occurs if HALT mode was set as part of an interrupt servicing routine.

(i) Only HALT mode is released when an interrupt request that has a lower priority level than the interrupt currently being serviced occurs, and the lower-priority interrupt request is not acknowledged. The interrupt request itself is retained.

(ii) When an interrupt request (including NMI request) that has a higher priority level than the interrupt currently being serviced occurs, HALT mode is released and the interrupt request is acknowledged.

(b) Release by $\overline{\text{RESET}}$ pin input

This is the same as for normal reset operations.

Table 6-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item		When CPU Operates with Main Clock		When CPU Operates with Subclock	
		When subclock does not exist	When subclock exists	When main clock's oscillation continues	When main clock's oscillation is stopped
CPU		Stopped			
Clock generator		Oscillation for main clock and subclock Clock supply to CPU is stopped			
16-bit timer (TM0)		Operating			Operates when INTWTI is selected as count clock (f_{XT} is selected for watch timer)
16-bit timer (TM1)		Operating			Stopped
8-bit timer (TM2)		Operating			Stopped
8-bit timer (TM3)		Operating			Stopped
8-bit timer (TM4)		Operating			Operates when f_{XT} is selected for count clock
8-bit timer (TM5)		Operating			Operates when f_{XT} is selected for count clock
Watch timer		Operates when $f_{XX}/2^9$ is selected for count clock	Operating		Operates when f_{XT} is selected for count clock
Watchdog timer		Operating (interval timer only)			
Serial interface	CSI0 to CSI2	Operating			Operates when an external clock is selected as the serial clock
	I^2CNote	Operating			Stopped
	UART0, UART1	Operating			Operates when an external clock is selected as the baud rate clock (transmit only)
A/D converter		Operating			Stopped
DMA0 to DMA2		Operating			
Real-time output		Operating			
Port function		Held			
External bus interface		Only bus hold function operates			
External interrupt request	NMI	Operating			
	INTP0 to INTP3	Operating			
	INTP4 to INTP6	Operating			Stopped

★ **Note** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY

Table 6-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item		When CPU Operates with Main Clock		When CPU Operates with Subclock	
		When subclock does not exist	When subclock exists	When main clock's oscillation continues	When main clock's oscillation is stopped
In external expansion mode	AD0 to AD15	High impedance[Note]			
	A16 to A21				
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	Held[Note] (high impedance when $\overline{\text{HLDAK}}$ = 0)			
	R/$\overline{\text{W}}$	High level output[Note] (high impedance when $\overline{\text{HLDAK}}$ = 0)			
	$\overline{\text{DSTB}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$				
	ASTB				
	$\overline{\text{HLDAK}}$	Operating			

Note Even when the HALT instruction has been executed, the instruction fetch operation continues until the on-chip instruction prefetch queue becomes full. Once it is full, operation stops in the status shown in Table 6-1.

6.4.3 IDLE mode

(1) Settings and operating states

This mode stops the entire system except the watch timer by stopping the on-chip main clock supply while the clock oscillator is still operating. Supply to the subclock continues. When this mode is released, there is no need for the oscillator to wait for the oscillation stabilization time, so normal operation can be resumed quickly.

In IDLE mode, program execution is stopped and the contents of all registers and internal RAM are retained as they were just before IDLE mode was set. In addition, on-chip peripheral functions are stopped (except for peripheral functions that are operating with the subclock). External bus hold requests ($\overline{\text{HLDRQ}}$) are not acknowledged.

When the IDLE bit of the power save control register (PSC) is set to 1, the system switches to IDLE mode.

The operating statuses in IDLE mode are listed in Table 6-2.

(2) Release of IDLE mode

IDLE mode can be released by a non-maskable interrupt, an unmasked interrupt request output from an on-chip peripheral I/O that can be operated, or $\overline{\text{RESET}}$ pin input.

Table 6-2. Operating Statuses in IDLE Mode (1/2)

IDLE Mode Settings / Item		When Subclock Exists	When Subclock Does Not Exist
CPU		Stopped	
Clock generator		Both main clock and subclock oscillator Clock supply to CPU and on-chip peripheral functions is stopped	
16-bit timer (TM0)		Operates when INTWTI is selected as count clock (f_{XT} is selected for watch timer)	Stopped
16-bit timer (TM1)		Stopped	
8-bit timer (TM2)		Stopped	
8-bit timer (TM3)		Stopped	
8-bit timer (TM4)		Operates when f_{XT} is selected for count clock	Stopped
8-bit timer (TM5)		Operates when f_{XT} is selected for count clock	Stopped
Watch timer		Operates when f_{XT} is selected for count clock	Stopped
Watchdog timer		Stopped	
Serial interface	CSI0 to CSI2	Operates when an external clock is selected as the serial clock	
	I^2C Note	Stopped	
	UART0, UART1	Operates when an external clock is selected as the baud rate clock (transmit only)	
A/D converter		Stopped	
DMA0 to DMA2		Stopped	
Real-time output		Operates when INTTM4 or INTTM5 is selected (when TM4 or TM5 is operating)	Stopped

★ **Note** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY

Table 6-2. Operating Statuses in IDLE Mode (2/2)

IDLE Mode Settings / Item		When Subclock Exists	When Subclock Does Not Exist
Port function		Held	
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3	Operating	
	INTP4 to INTP6	Stopped	
In external expansion mode	AD0 to AD15	High impedance	
	A16 to A21		
	$\overline{LBEN}$, $\overline{UBEN}$		
	R/$\overline{W}$		
	$\overline{DSTB}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{RD}$		
	ASTB		
	$\overline{HLDAK}$		

6.4.4 Software STOP mode

⋆ (1) Settings and operating states

This mode stops the entire system by stopping the main clock oscillator to stop supplying the internal main clock. The subclock oscillator continues operating and the on-chip subclock supply is continued.

When the subclock is not used, low power consumption of only the current flowing through the on-chip feed-back resistor and leakage current is realized.

In this mode, program execution is stopped and the contents of all registers and internal RAM are retained as they were just before software STOP mode was set. On-chip peripheral functions also stop operation (peripheral functions operating on the subclock are not stopped). External bus hold requests ($\overline{HLDRQ}$) are not acknowledged.

This mode can be set only when the main clock is being used as the CPU clock. This mode is set when the STP bit in the power save control register (PSC) has been set to 1.

Do not set this mode when the subclock has been selected as the CPU clock.

The operating statuses in software STOP mode are listed in Table 6-3.

(2) Release of software STOP mode

Software STOP mode can be released by a non-maskable interrupt, an unmasked interrupt request output from an on-chip peripheral I/O that can be operated, or $\overline{RESET}$ input.

When the software STOP mode is released, the oscillation stabilization time is secured.

Table 6-3. Operating Statuses in Software STOP Mode

Item \ Software STOP Mode Settings		When Subclock Exists	When Subclock Does Not Exist
CPU		Stopped	
Clock generator		Oscillation for main clock is stopped and oscillation for subclock continues Clock supply to CPU and on-chip peripheral functions is stopped	
16-bit timer (TM0)		Operates when INTWTI is selected for count clock (f_{XT} is selected as count clock for watch timer)	Stopped
16-bit timer (TM1)		Stopped	
8-bit timer (TM2)		Stopped	
8-bit timer (TM3)		Stopped	
8-bit timer (TM4)		Operates when f_{XT} is selected for count clock	Stopped
8-bit timer (TM5)		Operates when f_{XT} is selected for count clock	Stopped
Watch timer		Operates when f_{XT} is selected for count clock	Stopped
Watchdog timer		Stopped	
Serial interface	CSI0 to CSI2	Operates when an external clock is selected as the serial clock	
	I^2C[Note]	Stopped	
	UART0, UART1	Operates when an external clock is selected as the baud rate clock (transmit only)	
A/D converter		Stopped	
DMA0 to DMA2		Stopped	
Real-time output		Operates when INTTM4 or INTTM5 has been selected (when TM4 or TM5 is operating)	Stopped
Port function		Held	
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3	Operating	
	INTP4 to INTP6	Stopped	
In external expansion mode	AD0 to AD15	High impedance	
	A16 to A21		
	$\overline{LBEN}$, $\overline{UBEN}$		
	R/$\overline{W}$		
	$\overline{DSTB}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{RD}$		
	ASTB		
	$\overline{HLDAK}$		

★ **Note** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY

6.5 Oscillation Stabilization Time

The following shows the methods for specifying the length of the oscillation stabilization time required to stabilize the oscillator following release of software STOP mode.

(1) Release non-maskable interrupt or by unmasked interrupt request

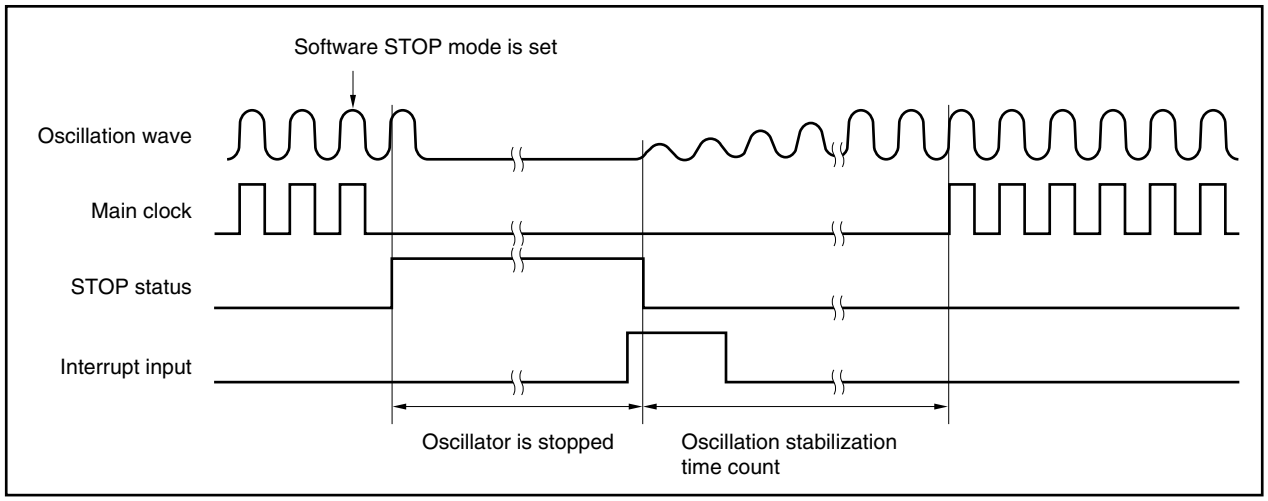
Software STOP mode is released by a non-maskable interrupt or an unmasked interrupt request. When an interrupt is input to this pin, the counter (watchdog timer) starts counting and the count time is the length of time that must elapse for stabilization of the oscillator's clock output.

The oscillation stabilization time is set by the oscillation stabilization time select register (OSTS).

Oscillation stabilization time ≒ WDT count time

After the specified amount of time has elapsed, system clock output starts and processing branches to the interrupt handler address.

Figure 6-2. Oscillation Stabilization Time



(2) Use of $\overline{\text{RESET}}$ pin to secure time ($\overline{\text{RESET}}$ pin input)

For securing time with the $\overline{\text{RESET}}$ pin, refer to **CHAPTER 15 RESET FUNCTION**.

The oscillation stabilization time is 2^{19}/fxx according to the value of the OSTS register after reset.

★ 6.6 Cautions on Power Save Function

(1) While an instruction is being executed on internal ROM

To set the power save mode (IDLE mode or software STOP mode) while an instruction is being executed on the internal ROM, insert a NOP instruction as a dummy instruction to correctly execute the routine after releasing the power save mode.

The following shows the sequence of setting the power save mode.

<1> Disable DMA operation.
<2> Disable interrupts (set NP bit of PSW to 1).
<3> Write 8-bit data to the command register (PRCMD).
<4> Write setting data to the power save control register (PSC) (using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<5> Clear the interrupt disabled state (re-set the NP bit of the PSW to 0).
<6> Insert NOP instructions (2 or 5 instructions).
<7> If DMA operation is necessary, enable DMA operation.

Cautions 1. Insert two NOP instructions if the ID bit value of the PSW is not changed by the execution of the instruction that clears the NP bit to 0 (<5>), and insert five NOP instructions if changed. The following shows a description example.

[Description example] : When using PSC register

```
LDSR rX.5           ; NP bit = 1
ST.B r0, PRCMD[r0]  ; Write to PRCMD
ST.B rD, RSC[r0]    ; PSC register setting
LDSR rY, 5          ; NP bit = 0
NOP                 ; Dummy instructions (2 or 5 instructions)
:
NOP
(next instruction)  ; Execution routine after releasing IDLE/software STOP
                      mode
:
```

rX: Value to be written to PSW
rY: Value to be rewritten to PSW
rD: Value to be set to PSC

When saving the PSW value, transfer the PSW value before setting the NP bit to the rY register.

2. The instructions (<5> interrupt disable clear, <6> NOP instruction) following the store instruction to the PSC register for setting the IDLE mode and software STOP mode are executed before entering the power save mode.

(2) While an instruction is being executed on external ROM

(i) Do not set the power save mode (IDLE or software STOP mode) while an instruction is being executed on the external ROM.

(ii) To set the power save mode (IDLE or software STOP mode) while an instruction is being executed on the external ROM, handle as follows.

<1> Insert six NOP instructions 4 bytes after the instruction that writes to the PSC register.

<2> Insert the BR $+2 instruction following the NOP instruction to eliminate the discrepancy of the program counter (PC).

[Processing program example]

```
LDSR rX.5             ; NP bit = 1
ST.B r0, PRCMD[r0]    ; Write to PRCMD
ST.B rD, RSC[r0]      ; PSC register setting
LDSR rY, 5            ; NP bit = 0
NOP                   ; NOP instruction (6 instructions)
NOP
NOP
NOP
NOP
NOP
BR   $+2              ; Eliminate discrepancy of PC
```

rX: Value to be written to PSW

rY: Value to be rewritten to PSW

rD: Value to be set to PSC

CHAPTER 7 TIMER/COUNTER FUNCTION

7.1 16-Bit Timers (TM0, TM1)

7.1.1 Outline

- 16-bit capture/compare registers: 2 (CRn0, CRn1)
- Independent capture/trigger inputs: 2 (TIn0, TIn1)
- Support of output of capture/match interrupt request signals (INTTMn0, INTTMn1)
- Event input (shared with TIn0) via digital noise eliminator and support of edge specification
- Timer output operated by match detection: 1 each (TOn)
  When using the P34/TO0 and P35/TO1 pins as the TO0 and TO1 pins (timer output), set the value of port 3 (P3) to 0 (low-level output) and the port 3 mode register (PM3) to 0 (port output mode). The logical sum (ORed) value of the output of the port and the timer is output.

Remark n = 0, 1

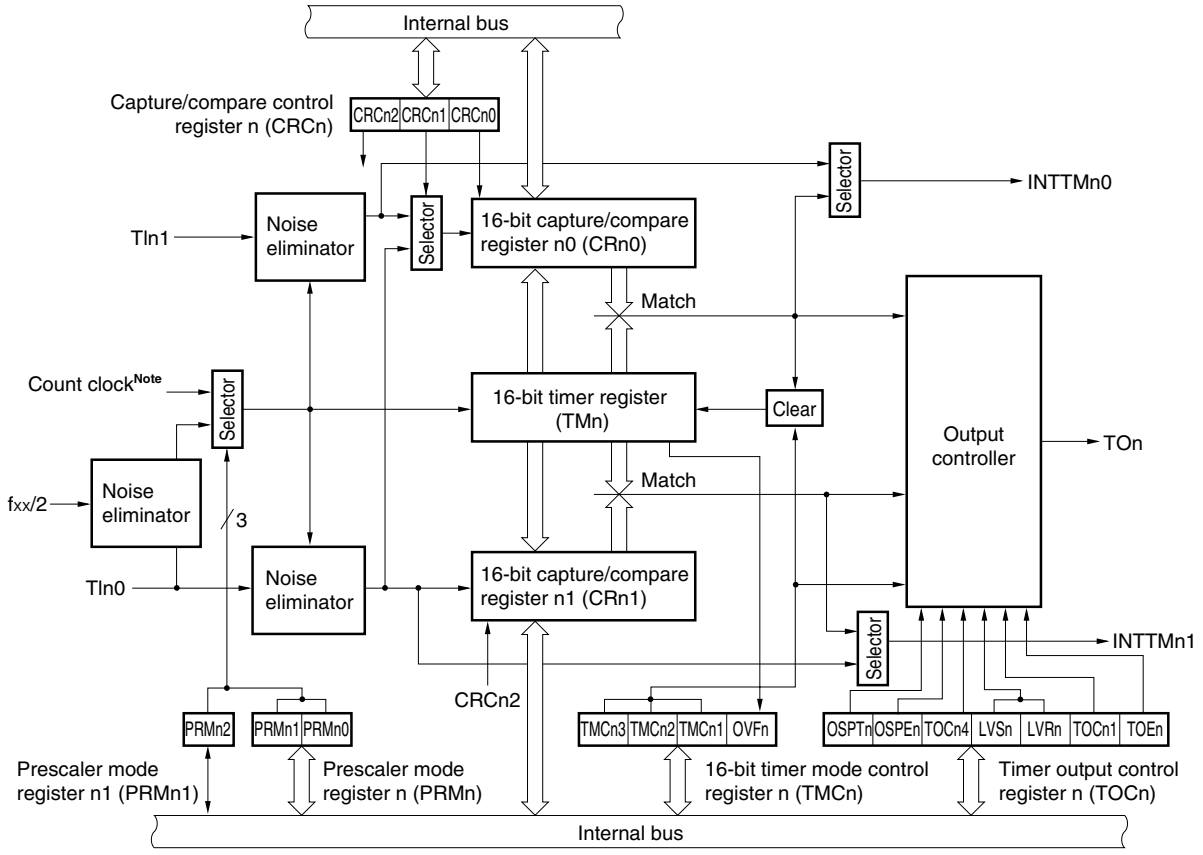
7.1.2 Functions

TM0 and TM1 have the following functions.

- Interval timer
- PPG output
- Pulse width measurement
- External event counter
- Square wave output
- One-shot pulse output

The block diagram is shown below.

Figure 7-1. Block Diagram of TM0 and TM1



Note The count clock is set by the PRMn and PRMn1 registers.

Remark n = 0, 1

(1) Interval timer

Generates an interrupt at predetermined time intervals.

(2) PPG output

Can output a square wave whose frequency and output pulse width can be changed arbitrarily.

(3) Pulse width measurement

Can measure the pulse width of a signal input from an external source.

(4) External event counter

Can measure the number of pulses of a signal input from an external source.

(5) Square wave output

Can output a square wave of any frequency.

(6) One-shot pulse output

Can output a one-shot pulse with any output pulse width.

7.1.3 Configuration

Timers 0 and 1 consist of the following hardware.

Table 7-1. Configuration of Timers 0 and 1

Item	Configuration
Timer registers	16 bits × 2 (TM0, TM1)
Registers	Capture/compare registers: 16 bits × 2 (CRn0, CRn1)
Timer outputs	2 (TO0, TO1)
Control registers	16-bit timer mode control registers 0, 1 (TMC0, TMC1) Capture/compare control registers 0, 1 (CRC0, CRC1) 16-bit timer output control registers 0, 1 (TOC0, TOC1) Prescaler mode register n, n1 (PRMn, PRMn1)

Remark n = 0, 1

(1) 16-bit timer registers 0, 1 (TM0, TM1)

TMn is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the input clock. If the count value is read during operation, input of the count clock is temporarily stopped, and the count value at that point is read. The count value is reset to 0000H in the following cases.

<1> At $\overline{\text{RESET}}$ input
<2> If TMCn3 and TMCn2 are cleared
<3> If the valid edge of TIn0 is input in the mode in which the timer is created and started on inputting the valid edge of TIn0
<4> If TMn and CRn0 match in the clear and start mode entered on a match between TMn and CRn0
<5> If OSPTn is set or if the valid edge of TIn0 is input in the one-shot pulse output mode

(2) Capture/compare registers 00, 10 (CR00, CR10)

CRn0 is a 16-bit register that functions as a capture register and as a compare register. Whether this register functions as a capture or compare register is specified by using bit 0 (CRCn0) of the CRCn register.

(a) When using CRn0 as compare register

The value set to CRn0 is always compared with the count value of the TMn register. When the values of the two match, an interrupt request (INTTMn0) is generated. When TMn is used as an interval timer, CRn0 can also be used as a register that holds the interval time.

(b) When using CRn0 as capture register

The valid edge of the TIn0 or TIn1 pin can be selected as a capture trigger. The valid edge of TIn0 or TIn1 is set by using the PRMn register.

When the valid edge of the TIn0 pin is specified as the capture trigger, refer to **Table 7-2**. When the valid edge of the TIn1 pin is specified as the capture trigger, refer to **Table 7-3**.

Table 7-2. Valid Edge of TIn0 Pin and Capture Trigger of CRn0

ESn01	ESn00	Valid Edge of TIn0 Pin	CRn0 Capture Trigger
0	0	Falling edge	Rising edge
0	1	Rising edge	Falling edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation

Remark n = 0, 1

Table 7-3. Valid Edge of TIn1 Pin and Capture Trigger of CRn0

ESn11	ESn10	Valid Edge of TIn1 Pin	CRn0 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

Remark n = 0, 1

CRn0 is set by using a 16-bit memory manipulation instruction.

These registers can be read/written when used as compare registers and can only be read when used as capture registers.

$\overline{\text{RESET}}$ input sets this register to 0000H.

★ **Caution In a mode in which the timer is cleared and started on a match between TMn and CRn0, set the CRn0 register to other than 0000H. In the free-running mode or the TIn0 valid edge clear mode, however, an interrupt request (INTTMn0) is generated after an overflow (FFFFH) when CRn0 is set to 0000H.**

(3) Capture/compare registers 01, 11 (CR01, CR11)

This is a 16-bit register that can be used as a capture register and a compare register. Whether it is used as a capture register or compare register is specified by bit 2 (CRCn2) of the CRCn register.

(a) When using CRn1 as compare register

The value set to CRn1 is always compared with the count value of TMn. When the values of the two match, an interrupt request (INTTMn1) is generated.

(b) When using CRn1 as capture register

The valid edge of the TIn0 pin can be selected as a capture trigger. The valid edge of TIn0 is specified by using the PRMn register.

Table 7-4. Valid Edge of TIn0 Pin and Capture Trigger of CRn1

ESn01	ESn00	Valid Edge of TIn0 Pin	CRn1 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

Remark n = 0, 1

CRn1 is set by using a 16-bit memory manipulation instruction. These registers can be read/written when used as compare registers and can only be read when used as capture registers.

The value of this register is set to 0000H after the $\overline{\text{RESET}}$ signal is input.

★ **Caution In a mode in which the timer is cleared and started on a match between TMn and CRn0, set the CRn1 register to other than 0000H. In the free-running mode or the TIn0 valid edge clear mode, however, an interrupt request (INTTMn1) is generated after an overflow (FFFFH) when CRn1 is set to 0000H.**

7.1.4 Timer 0, 1 control registers

The following four types of registers control timers 0 and 1.

- 16-bit timer mode control register n (TMCn)
- Capture/compare control register n (CRCn)
- 16-bit timer output control register n (TOCn)
- Prescaler mode register n, n1 (PRMn, PRMn1)

Remark n = 0, 1

(1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

TMCn specifies the operation mode of the 16-bit timer, and the clear mode, output timing, and overflow detection of 16-bit timer register n.

TMCn is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC0 and TMC1 to 00H.

Caution 16-bit timer register n starts operating when TMCn2 and TMCn3 are set to values other than 0, 0 (operation stop mode). To stop the operation, set TMCn2 and TMCn3 to 0, 0.

After reset: 00H R/W Address: FFFFF208H, FFFFF218H

	7	6	5	4	3	2	1	0
TMCn	0	0	0	0	TMCn3	TMCn2	TMCn1	OVFn

(n = 0, 1)

TMCn3	TMCn2	TMCn1	Operation Mode and Clear Mode Selection	TOn Output Timing Selection	Generation of Interrupt
0	0	0	Operation stops (TMn is cleared to 0)	Not affected	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TMn and CRn0 or match between TMn and CRn1	Generated on match between TMn and CRn0 and match between TMn and CRn1
0	1	1		Match between TMn and CRn0, match between TMn and CRn1, or valid edge of TIn0	
1	0	0	Clears and starts at valid edge of TIn0	Match between TMn and CRn0 or match between TMn and CRn1	
1	0	1		Match between TMn and CRn0, match between TMn and CRn1, or valid edge of TIn0	
1	1	0	Clears and starts on match between TMn and CRn0	Match between TMn and CRn0 or match between TMn and CRn1	
1	1	1		Match between TMn and CRn0, match between TMn and CRn1, or valid edge of TIn0	

OVFn	Detection of Overflow of 16-Bit Timer Register n
0	Did not overflow
1	Overflowed

Cautions 1. When a bit other than the OVFn flag is written, be sure to stop the timer operation.

2. The valid edge of the TIn0 pin is set by using prescaler mode register n (PRMn).

3. When a mode in which the timer is cleared and started on a match between TMn and CRn0 is selected, the OVFn flag is set to 1 when the count value of TMn changes from FFFFH to 0000H with CRn0 set to FFFFH.

4. Be sure to set bits 4 to 7 to 0.

Remark TOn: Output pin of timer n
TIn0: Input pin of timer n
TMn: 16-bit timer register n
CRn0: Compare register n0
CRn1: Compare register n1

(2) Capture/compare control registers 0, 1 (CRC0, CRC1)

CRCn controls the operation of capture/compare register n (CRn0 and CRn1).

CRCn is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CRC0 and CRC1 to 00H.

After reset: 00H R/W Address: FFFFF20AH, FFFFF21AH

	7	6	5	4	3	2	1	0
CRCn	0	0	0	0	0	CRCn2	CRCn1	CRCn0

(n = 0, 1)

CRCn2	Selection of Operation Mode of CRn1
0	Operates as compare register
1	Operates as capture register

CRCn1	Selection of Capture Trigger of CRn0
0	Captured at valid edge of TIn1
1	Captured in reverse phase of valid edge of TIn0

CRCn0	Selection of Operation Mode of CRn0
0	Operates as compare register
1	Operates as capture register

Cautions 1. Before setting CRCn, be sure to stop the timer operation.

2. When the mode in which the timer is cleared and started on a match between TMn and CRn0 is selected by 16-bit timer mode control register n (TMCn), do not specify CRn0 as a capture register.

3. When both the rising edge and falling edge are specified for the TIn0 valid edge, the capture operation does not work.

4. To ensure that the capture trigger captures the signals from TIn0 and TIn1 correctly, a pulse longer than two of the count clocks selected by prescaler mode registers n and n1 (PRMn, PRMn1) is required.

5. Be sure to set bits 3 to 7 to 0.

(3) 16-bit timer output control registers 0, 1 (TOC0, TOC1)

TOCn controls the operation of the timer n output controller by setting or resetting the R-S flip-flop (LV0), enabling or disabling reverse output, enabling or disabling output of timer n, enabling or disabling one-shot pulse output operation, and selecting the output trigger for the one-shot pulse by software.

TOCn is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TOC0 and TOC1 to 00H.

After reset: 00H R/W Address: FFFFF20CH, FFFFF21CH

	7	6	5	4	3	2	1	0
TOCn (n = 0, 1)	0	OSPTn	OSPEn	TOCn4	LVSn	LVRn	TOCn1	TOEn

OSPTn	Control of Output Trigger of One-shot Pulse by Software
0	No one-shot pulse trigger
1	One-shot pulse trigger used

OSPEn	Control of One-shot Pulse Output Operation
0	Successive pulse output
1	One-shot pulse output[Note]

TOCn4	Control of Timer Output F/F on Match between CRn1 and TMn
0	Reverse timer output F/F disabled
1	Reverse timer output F/F enabled

LVSn	LVRn	Setting of Status of Timer Output F/F of Timer n
0	0	Not affected
0	1	Resets timer output F/F (0)
1	0	Sets timer output F/F (1)
1	1	Setting prohibited

TOCn1	Control of Timer Output F/F on Match between CRn0 and TMn
0	Reverse timer output F/F disabled
1	Reverse timer output F/F enabled

TOEn	Control of Output of Timer n
0	Output disabled (output is fixed to 0 level)
1	Output enabled

Note The one-shot pulse output operates normally in the free-running mode and clear and start mode set by the valid edge of TIn0.

Cautions 1. Before setting TOCn, be sure to stop the timer operation.

2. LVSn and LVRn are 0 when read after data has been set to them.

3. OSPTn is 0 when read because it is automatically cleared after data has been set.

4. Do not set OSPTn other than for one-shot pulse output.

(4) Prescaler mode registers 0, 01 (PRM0, PRM01)

PRM0 and PRM01 select the count clock of the 16-bit timer (TM0) and the valid edge of TI0n input. PRM0 and PRM01 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM0 and PRM01 to 00H.

Remark n = 0, 1

After reset: 00H R/W Address: FFFFF20EH

	7	6	5	4	3	2	1	0
PRM01	0	0	0	0	0	0	0	PRM02

After reset: 00H R/W Address: FFFFF206H

	7	6	5	4	3	2	1	0
PRM0	ES011	ES010	ES001	ES000	0	0	PRM01	PRM00

ES011	ES010	Selection of Valid Edge of TI01
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES001	ES000	Selection of Valid Edge of TI00
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM02	PRM01	PRM00	Count Clock Selection				
			Count Clock	f_{XX}			
				20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	$f_{XX}/2$	100 ns	118 ns	200 ns	1 μs
0	0	1	$f_{XX}/16$	800 ns	941 ns	1.6 μs	8 μs
0	1	0	INTWTI	–	–	–	–
0	1	1	TI00 valid edge[Note]	–	–	–	–
1	0	0	$f_{XX}/4$	200 ns	235 ns	400 ns	2 μs
1	0	1	$f_{XX}/64$	3.2 μs	3.8 μs	6.4 μs	32 μs
1	1	0	$f_{XX}/256$	12.8 μs	15.1 μs	25.6 μs	128 μs
1	1	1	Setting prohibited	–	–	–	–

Note The external clock requires a pulse longer than two internal clocks ($f_{XX}/2$).

Cautions 1. When selecting the valid edge of TI00 as the count clock, do not specify the valid edge of TI00 to clear and start the timer and as a capture trigger.

2. Before setting data to the PRM0 and PRM01 registers, always stop the timer operation.

3. If the 16-bit timer (TM0) operation is enabled by specifying the rising edge or both edges as the valid edge of the TI0n pin while the TI0n pin is high level immediately after system reset, the rising edge is detected immediately after the rising edge or both edges is specified. Be careful when pulling up the TI0n pin. However, the rising edge is not detected when operation is enabled after it has been stopped.

(5) Prescaler mode registers 1, 11 (PRM1, PRM11)

PRM1 and PRM11 select the count clock of the 16-bit timer (TM1) and the valid edge of TI1n input. PRM1 and PRM11 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM1 and PRM11 to 00H.

Remark n = 0, 1

After reset: 00H R/W Address: FFFFF21EH

	7	6	5	4	3	2	1	0
PRM11	0	0	0	0	0	0	0	PRM12[Note]

After reset: 00H R/W Address: FFFFF216H

	7	6	5	4	3	2	1	0
PRM1	ES111	ES110	ES101	ES100	0	0	PRM11	PRM10

ES111	ES110	Selection of Valid Edge of TI11
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES101	ES100	Selection of Valid Edge of TI10
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM12	PRM01	PRM00	Count Clock Selection				
			Count Clock	f_{XX}			
				20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	$f_{XX}/2$	100 ns	118 ns	200 ns	1 μs
0	0	1	$f_{XX}/4$	200 ns	235 ns	400 ns	2 μs
0	1	0	$f_{XX}/16$	800 ns	941 ns	1.6 μs	8 μs
0	1	1	TI10 valid edge[Note]	–	–	–	–
1	0	0	$f_{XX}/32$	1.6 μs	1.9 μs	3.2 μs	16 μs
1	0	1	$f_{XX}/128$	6.4 μs	7.5 μs	12.8 μs	64 μs
1	1	0	$f_{XX}/256$	12.8 μs	15.1 μs	25.6 μs	128 μs
1	1	1	Setting prohibited	–	–	–	–

Note The external clock requires a pulse longer than two internal clocks ($f_{XX}/2$).

Cautions 1. When selecting the valid edge of TI10 as the count clock, do not specify the valid edge of TI10 to clear and start the timer and as a capture trigger.

2. Before setting data to the PRM1 and PRM11 registers, always stop the timer operation.

3. If the 16-bit timer (TM1) operation is enabled by specifying the rising edge or both edges for the valid edge of the TI1n pin while the TI1n pin is high level immediately after system reset, the rising edge is detected immediately after the rising edge or both edges is specified. Be careful when pulling up TI1n pin. However, the rising edge is not detected when operation is enabled after it has been stopped.

7.2 16-Bit Timer Operation

7.2.1 Operation as interval timer (16 bits)

TMn operates as an interval timer when 16-bit timer mode control register n (TMCn) and capture/compare control register n (CRCn) are set as shown in Figure 7-2.

In this case, TMn repeatedly generates an interrupt at the time interval specified by the count value set in advance to 16-bit capture/compare register n0 (CRn0).

When the count value of TMn matches the set value of CRn0, the value of TMn is cleared to 0, and the timer continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.

The count clock of the 16-bit timer/event counter can be selected by bits 0 and 1 (PRMn0 and PRMn1) of prescaler mode register n (PRMn) and by bit 0 (PRMn2) of prescaler mode register n1 (PRMn1).

Remark n = 0, 1

Figure 7-2. Control Register Settings When TMn Operates as Interval Timer

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

	TMCn3	TMCn2	TMCn1	OVFn
TMCn	1	1	0/1	0

Clears and starts on match between TMn and CRn0.

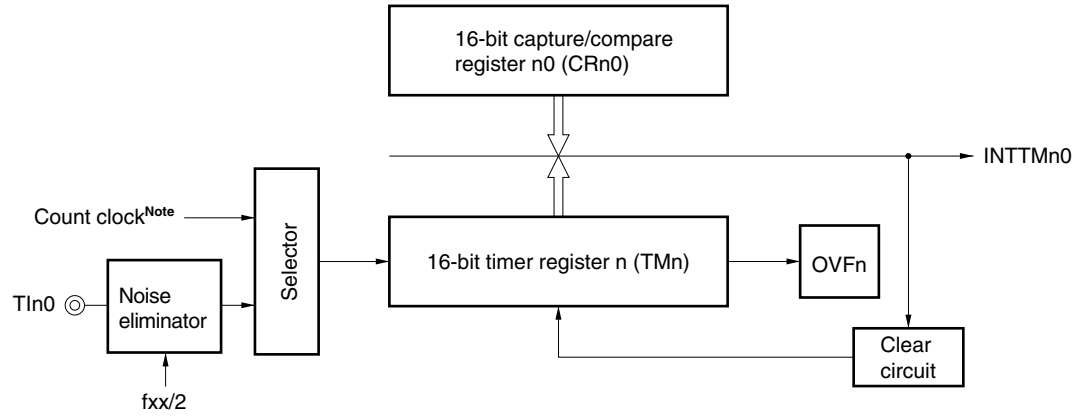
(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

	CRCn2	CRCn1	CRCn0
CRCn	0/1	0/1	0

CRn0 used as compare register

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the interval timer function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)** and **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**.

Figure 7-3. Configuration of Interval Timer

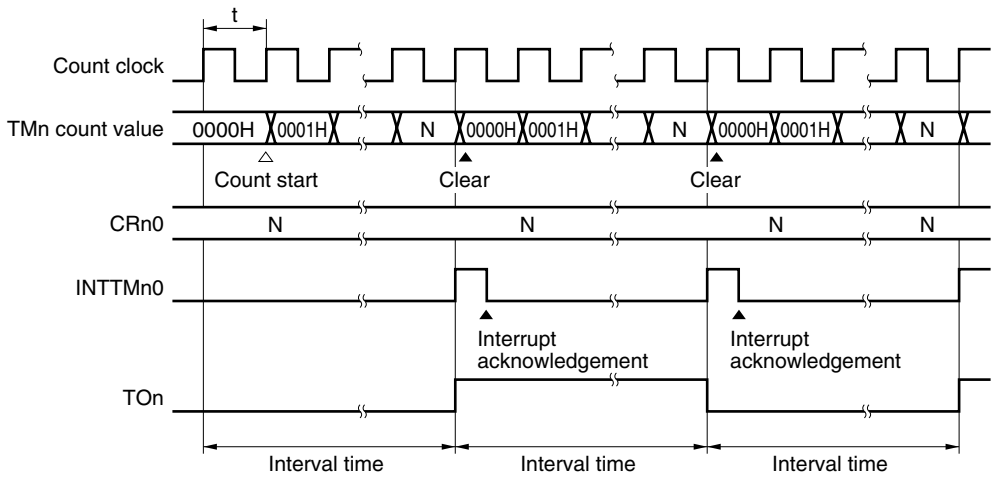


Note The count clock is set by the PRMn and PRMn1 registers.

Remarks 1. " ◎—— " indicates a signal that can be directly connected to a port.

2. n = 0, 1

Figure 7-4. Timing of Interval Timer Operation



Remarks 1. Interval time = (N + 1) × t: N = 0001H to FFFFH

2. n = 0,1

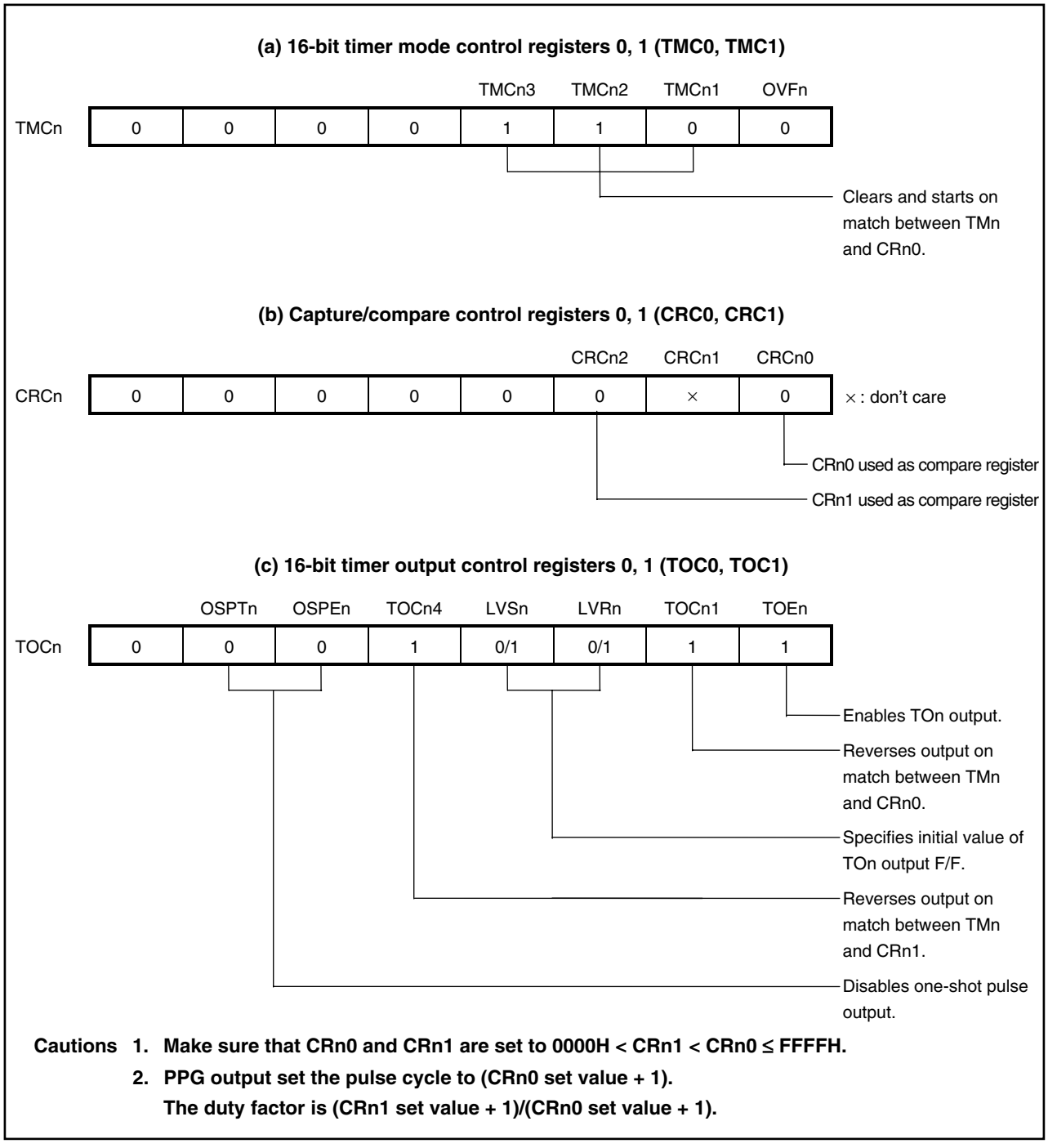
7.2.2 PPG output operation

TMn can be used for PPG (Programmable Pulse Generator) output by setting 16-bit timer mode control register n (TMCn) and capture/compare control register n (CRCn) as shown in Figure 7-5.

The PPG output function outputs a square wave from the TOn pin at the cycle specified by the count value set in advance to 16-bit capture/compare register n0 (CRn0) and the pulse width specified by the count value set in advance to 16-bit capture/compare register n1 (CRn1).

Remark n = 0, 1

Figure 7-5. Control Register Settings in PPG Output Operation



Cautions 1. Make sure that CRn0 and CRn1 are set to 0000H < CRn1 < CRn0 ≤ FFFFH.

2. PPG output set the pulse cycle to (CRn0 set value + 1).
The duty factor is (CRn1 set value + 1)/(CRn0 set value + 1).

7.2.3 Pulse width measurement

16-bit timer register n (TMn) can be used to measure the pulse widths of the signals input to the TIn0 and TIn1 pins.

Measurement can be carried out with TMn used as a free-running counter or by restarting the timer in synchronization with the edge of the signal input to the TIn0 pin.

(1) Pulse width measurement with free running counter and one capture register

If the edge specified by prescaler mode register n (PRMn) is input to the TIn0 pin when 16-bit timer register n (TMn) is used as a free-running counter (refer to **Figure 7-6**), the value of TMn is loaded to 16-bit capture/compare register n1 (CRn1), and an external interrupt request signal (INTTMn1) is set.

The edge is specified by using bits 6 and 7 (ESn10 and ESn11) of prescaler mode register n (PRMn). The rising edge, falling edge, or both the rising and falling edges can be selected.

The valid edge is detected by sampling at the count clock cycle selected by prescaler mode register n, n1 (PRMn, PRMn1), and a capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Remark n = 0, 1

Figure 7-6. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

					TMCn3	TMCn2	TMCn1	OVFn
TMCn	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

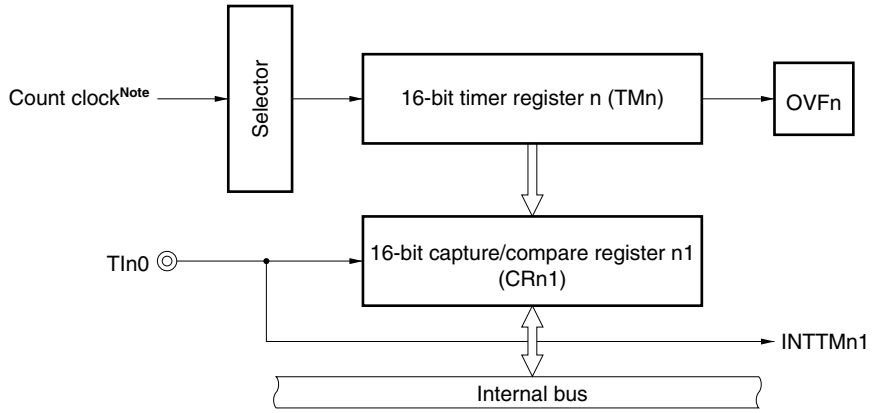
						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	1	0/1	0

CRn0 used as compare register

CRn1 used as capture register

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the pulse width measurement function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)** and **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**.

Figure 7-7. Configuration for Pulse Width Measurement with Free-Running Counter

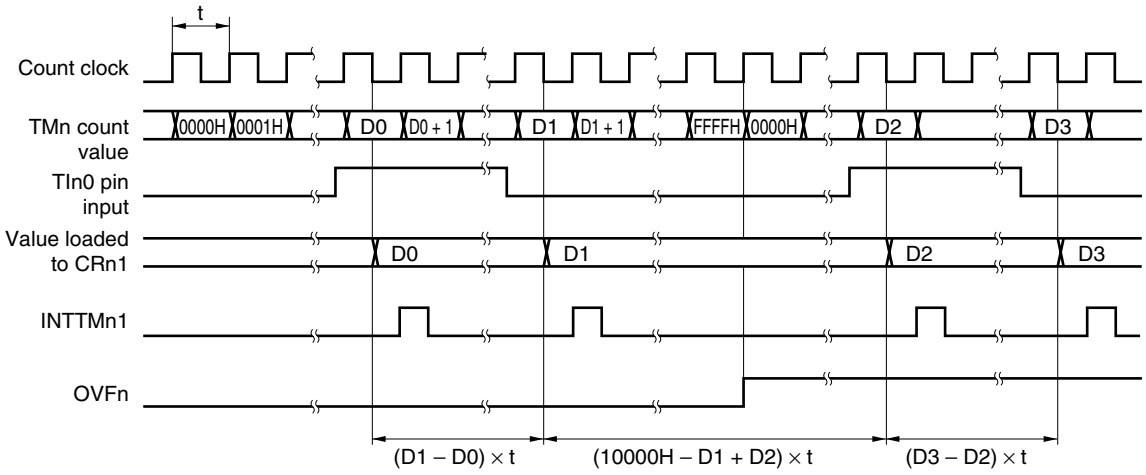


Note The count clock is set by the PRMn and PRMn1 registers.

Remarks 1. "◎——" indicates a signal that can be directly connected to a port.

2. n = 0, 1

Figure 7-8. Timing of Pulse Width Measurement with Free-Running Counter and One Capture Register (with Both Edges Specified)



Remark n = 0, 1

(2) Measurement of two pulse widths with free running counter

The pulse widths of the two signals respectively input to the TIn0 and TIn1 pins can be measured when 16-bit timer register n (TMn) is used as a free-running counter (refer to **Figure 7-9**).

When the edge specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn) is input to the TIn0 pin, the value of TMn is loaded to 16-bit capture/compare register n1 (CRn1) and an external interrupt request signal (INTTMn1) is set.

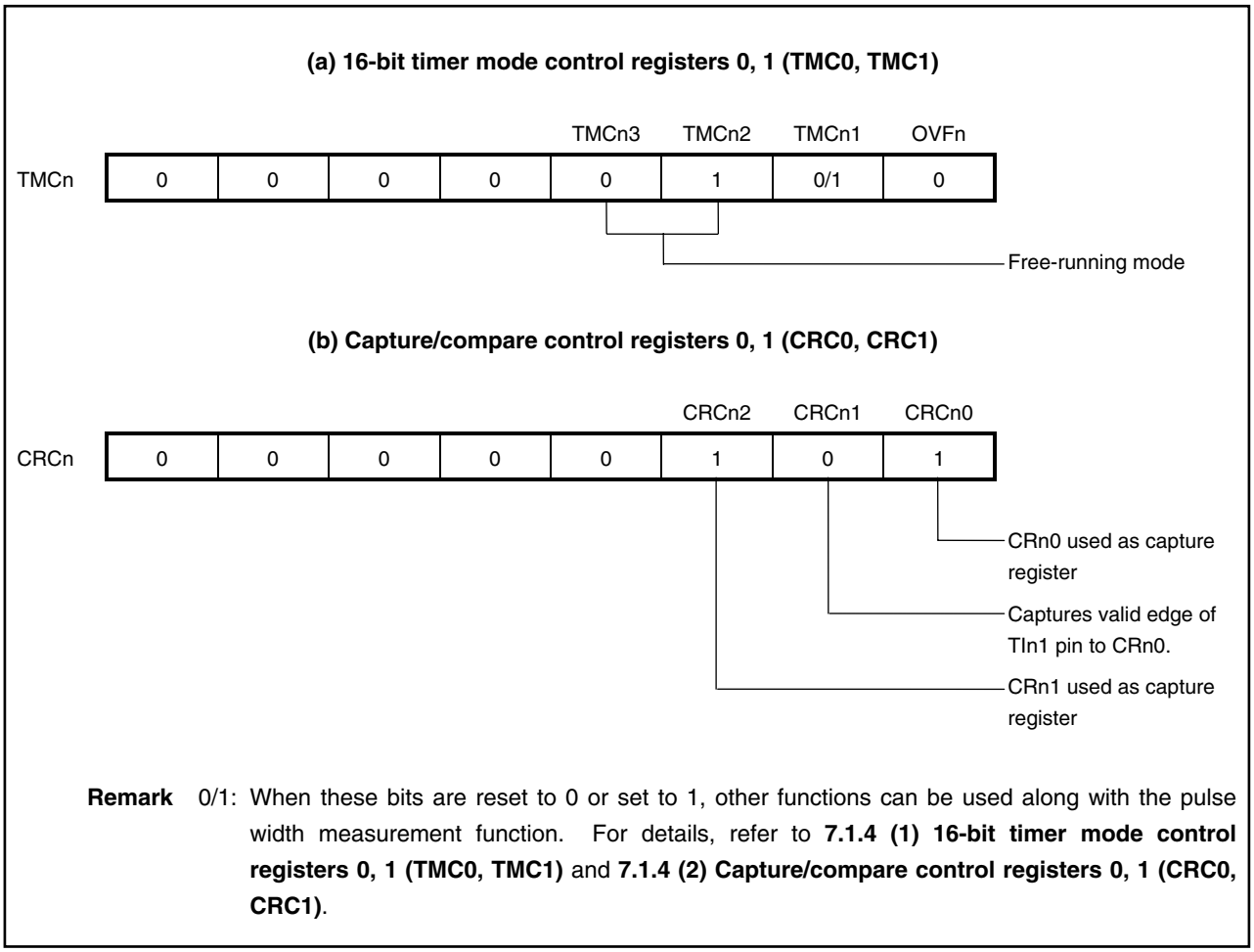
When the edge specified by bits 6 and 7 (ESn10 and ESn11) in PRMn is input to the TIn1 pin, the value of TMn is loaded to 16-bit capture/compare register n0 (CRn0), and an external interrupt request signal (INTTMn0) is set.

The edges of the TIn0 and TIn1 pins are specified by bits 4 and 5 (ESn00 and ESn01) and bits 6 and 7 (ESn10 and ESn11) of PRMn0, respectively. The rising, falling, or both rising and falling edges can be specified.

The valid edge is detected by sampling at the count clock cycle selected by prescaler mode register n, n1 (PRMn, PRMn1), and a capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Remark n = 0, 1

Figure 7-9. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

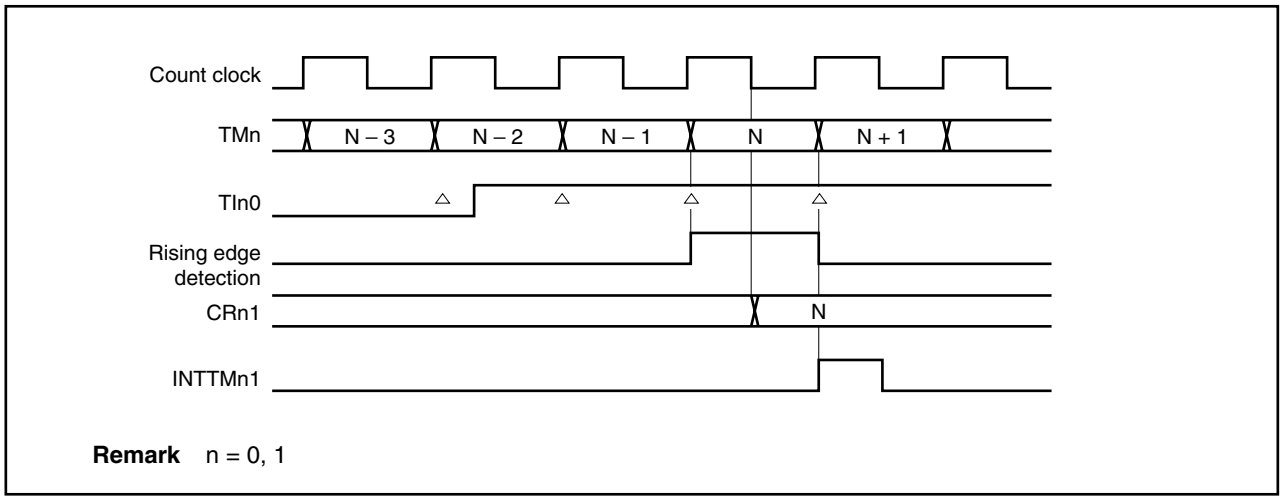


Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the pulse width measurement function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)** and **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**.

- **Capture operation (free running mode)**

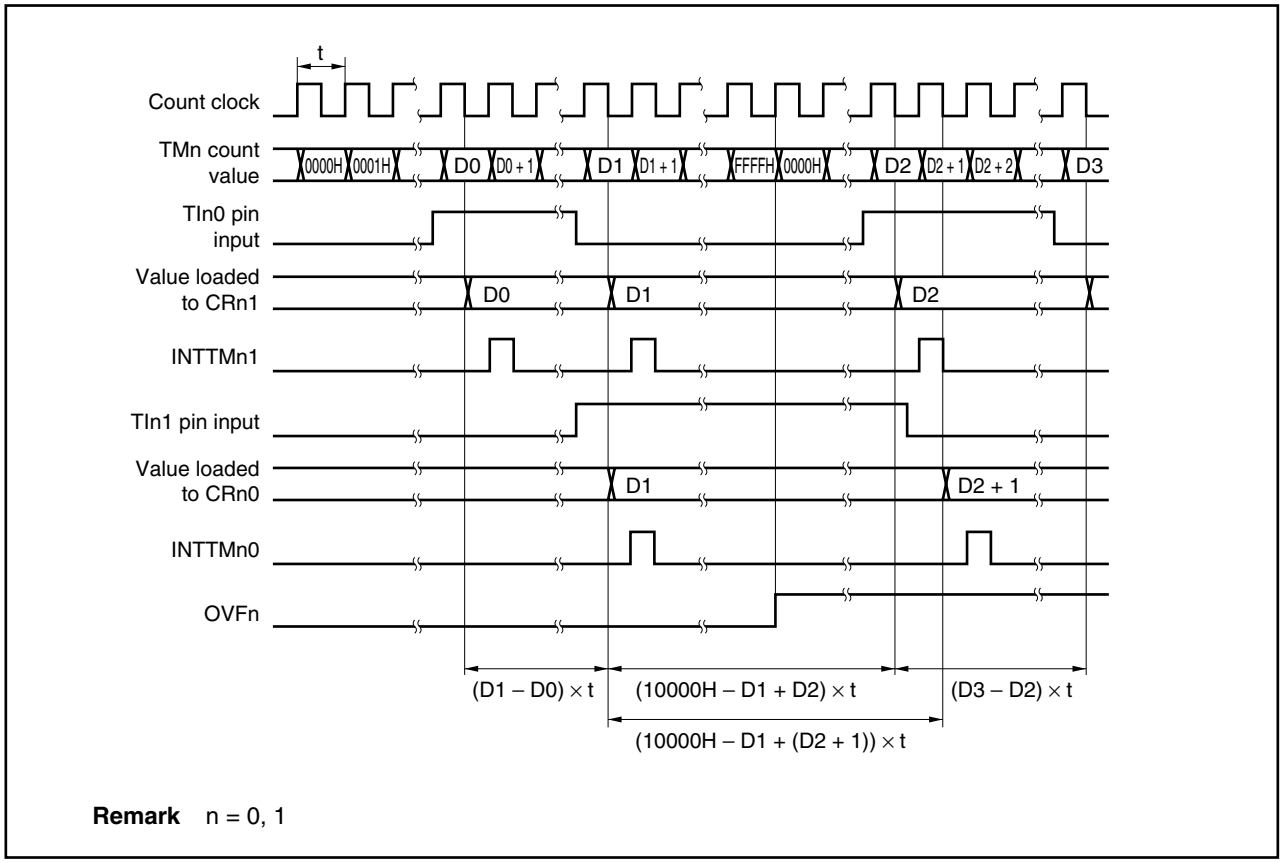
  The following figure illustrates the operation of the capture register when the capture trigger is input.

Figure 7-10. CRn1 Capture Operation with Rising Edge Specified



Remark n = 0, 1

Figure 7-11. Timing of Pulse Width Measurement with Free-Running Counter (with Both Edges Specified)



Remark n = 0, 1

(3) Pulse width measurement with free running counter and two capture registers

When 16-bit timer register n (TMn) is used as a free running counter (refer to **Figure 7-19**), the pulse width of the signal input to the TIn0 pin can be measured.

When the edge specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn) is input to the TIn0 pin, the value of TMn is loaded to 16-bit capture/compare register n1 (CRn1), and an external interrupt request signal (INTTMn1) is set.

The value of TMn is also loaded to 16-bit capture/compare register n0 (CRn0) when an edge reverse to the one that triggers capturing to CRn1 is input.

The edge of the TIn0 pin is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn). The rising or falling edge can be eliminated.

The valid edge of TIn0 is detected by sampling at the count clock cycle selected by prescaler mode register n, n1 (PRMn, PRMn1), and the capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Caution If the valid edge of the TIn0 pin is specified to be both the rising and falling edges, capture/compare register n0 (CRn0) cannot perform its capture operation.

Remark n = 0, 1

Figure 7-12. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

	TMCn3	TMCn2	TMCn1	OVFn
TMCn	0	1	0/1	0

Free-running mode

(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

	CRCn2	CRCn1	CRCn0
CRCn	1	1	1

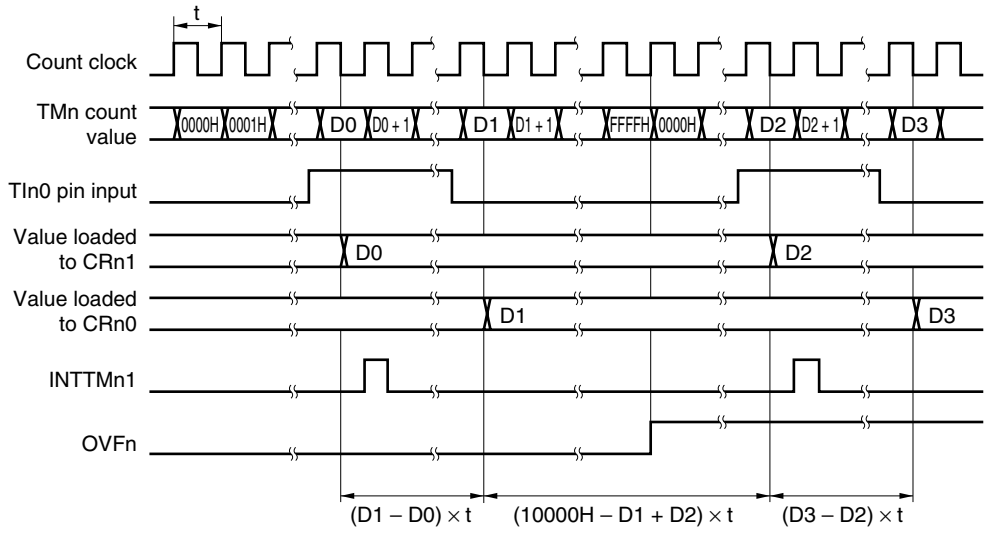
CRn0 used as capture register

Captures to CRn0 at edge reverse to valid edge of TIn0 pin.

CRn1 used as capture register

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the pulse width measurement function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)** and **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**.

Figure 7-13. Timing of Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Remark n = 0, 1

(4) Pulse width measurement by restarting

When the valid edge of the TIn0 pin is detected, the pulse width of the signal input to the TIn0 pin can be measured by clearing 16-bit timer register n (TMn) once and then resuming counting after loading the count value of TMn to 16-bit capture/compare register n1 (CRn1) (See **Figure 7-13**).

The edge is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn). The rising or falling edge can be specified.

The valid edge is detected by sampling at the count clock cycle selected by prescaler mode register n, n1 (PRMn, PRMn1) and the capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Caution If the valid edge of the TIn0 pin is specified to be both the rising and falling edges, capture/compare register n0 (CRn0) cannot perform its capture operation.

Remark n = 0, 1

Figure 7-14. Control Register Settings for Pulse Width Measurement by Restarting

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

					TMCn3	TMCn2	TMCn1	OVFn
TMCn	0	0	0	0	1	0	0/1	0

Clears and starts at valid edge of TIn0 pin.

(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	1	1	1

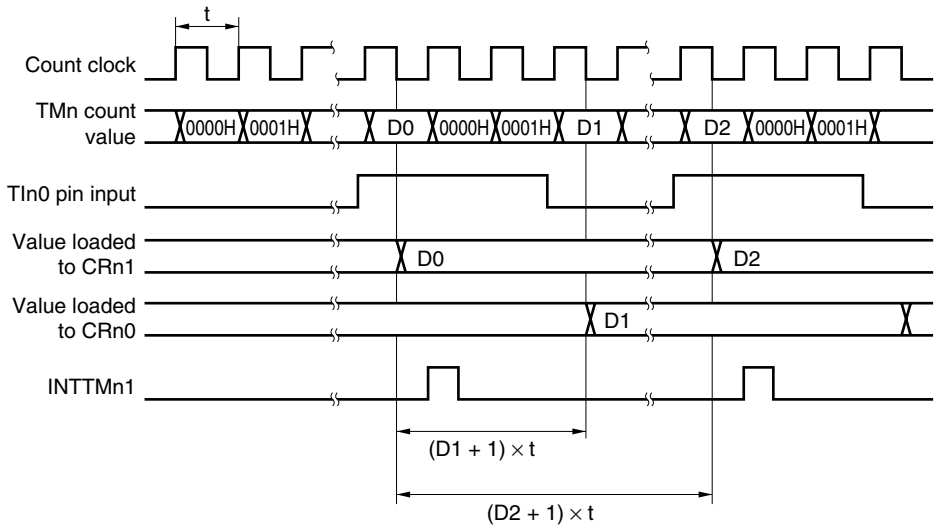
CRn0 used as capture register

Captures to CRn0 at edge reverse to valid edge of TIn0.

CRn1 used as capture register

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the pulse width measurement function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)** and **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**.

Figure 7-15. Timing of Pulse Width Measurement by Restarting (with Rising Edge Specified)



Remark n = 0, 1

7.2.4 Operation as external event counter

TMn can be used as an external event counter that counts the number of clock pulses input to the TIn0 pin from an external source by using 16-bit timer register n (TMn).

Each time the valid edge specified by prescaler mode register n (PRMn) has been input, TMn is incremented.

When the count value of TMn matches the value of 16-bit capture/compare register n0 (CRn0), TMn is cleared to 0, and an interrupt request signal (INTTMn0) is generated.

The edge is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn). The rising, falling, or both the rising and falling edges can be specified.

The valid edge is detected by sampling at a count clock cycle of fxx/2, and a capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Remark n = 0, 1

Figure 7-16. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

					TMCn3	TMCn2	TMCn1	OVFn
TMCn	0	0	0	0	1	1	0/1	0

Clears and starts on match between TMn and CRn0.

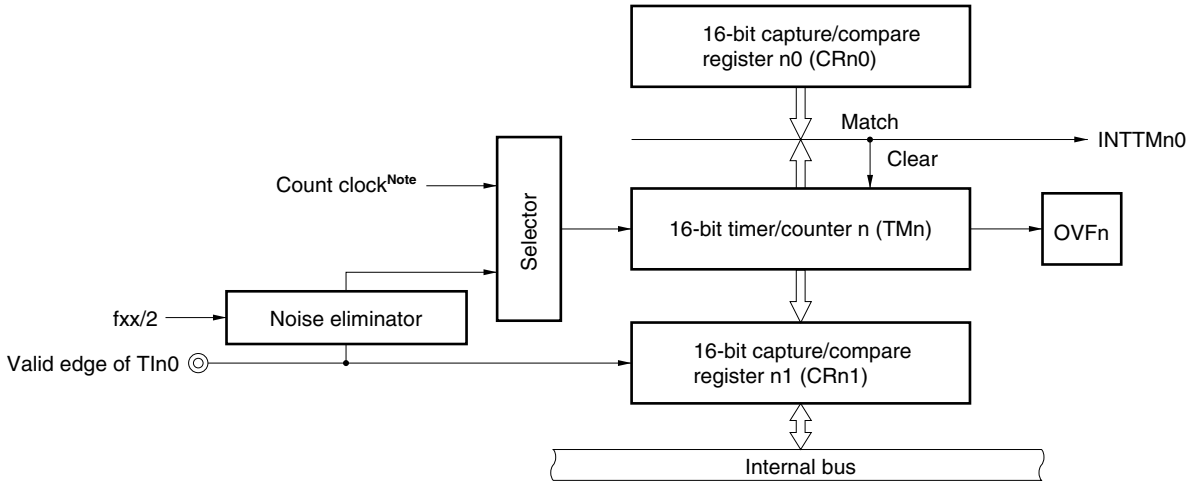
(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	0/1	0/1	0

CRn0 used as compare register

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the external event counter function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)** and **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**.

Figure 7-17. Configuration of External Event Counter

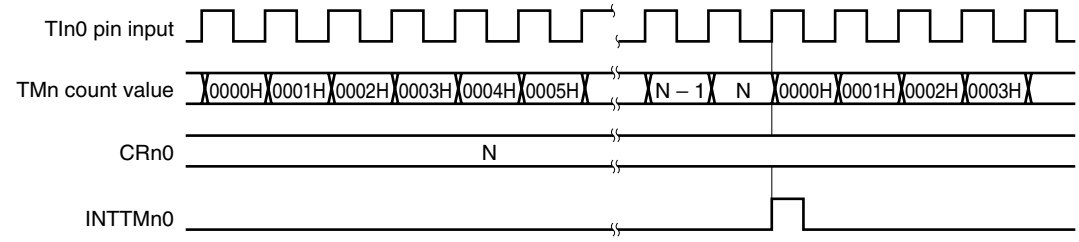


Note The count clock is set by the PRMn and PRMn1 registers.

Remarks 1. "◎——" indicates a signal that can be directly connected to a port.

2. n = 0, 1

Figure 7-18. Timing of External Event Counter Operation (with Rising Edge Specified)



Caution Read TMn when reading the count value of the external event counter.

Remark n = 0, 1

7.2.5 Operation to output square wave

TMn can be used to output a square wave with any frequency at the interval specified by the count value set in advance to 16-bit capture/compare register n0 (CRn0).

By setting bits 0 (TOEn) and 1 (TOCn1) of 16-bit timer output control register n (TOCn) to 1, the output status of the TOn pin is reversed at the interval specified by the count value set in advance to CRn1. In this way, a square wave of any frequency can be output.

Remark n = 0, 1

Figure 7-19. Control Register Settings in Square Wave Output Mode

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

					TMCn3	TMCn2	TMCn1	OVFn
TMCn	0	0	0	0	1	1	0	0

Clears and starts on match between TMn and CRn0.

(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	0/1	0/1	1

CRn0 used as compare register

(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)

		OSPTn	OSPEn	TOCn4	LVSn	LVRn	TOCn1	TOEn
TOCn	0	0	0	0	0/1	0/1	1	1

Enables TOn output.

Reverses output on match between TMn and CRn0.

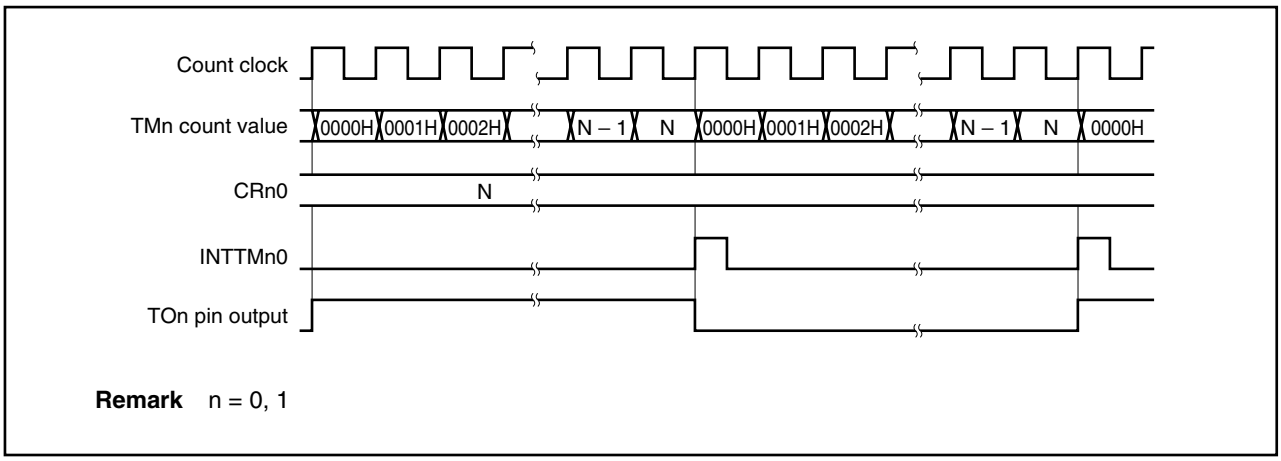
Specifies initial value of TOn output F/F.

Does not reverse output on match between TMn and CRn1.

Disables one-shot pulse output.

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the square wave output function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1), 7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**, and **7.1.4 (3) 16-bit timer output control registers 0, 1 (TOC0, TOC1)**.

Figure 7-20. Timing of Square Wave Output Operation



Remark n = 0, 1

7.2.6 Operation to output one-shot pulse

TMn can output a one-shot pulse in synchronization with a software trigger and an external trigger (TIn0 pin input).

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TOn pin by setting 16-bit timer mode control register n (TMCn), capture/compare control register n (CRCn), and 16-bit timer output control register n (TOCn) as shown in Figure 7-21, and by setting bit 6 (OSPTn) of TOCn by software.

By setting OSPTn to 1, the 16-bit timer/event counter is cleared and started, and its output is asserted at the count value (N) set in advance to 16-bit capture/compare register n1 (CRn1). After that, the output is deasserted at the count value (M) set in advance to 16-bit capture/compare register n0 (CRn0)[Note].

Even after the one-shot pulse has been output, TMn continues its operation. To stop TMn, TMCn must be reset to 00H.

Note This is an example when N < M. When N > M, output of CRn0 is asserted and output of CRn1 is deasserted.

Cautions 1. Do not set OSPTn to 1 while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is complete.

2. During a one-shot pulse output operation started by a software trigger, the TIn0 pin cannot be used as a general-purpose port.

Remark n = 0, 1

Figure 7-21. Control Register Settings for One-Shot Pulse Output with Software Trigger

(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)

					TMCn3	TMCn2	TMCn1	OVFn
TMCn	0	0	0	0	0	1	0	0

Free-running mode

(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	0	0/1	0

CRn0 used as compare register

CRn1 used as compare register

(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)

		OSPTn	OSPEn	TOCn4	LVSn	LVRn	TOCn1	TOEn
TOCn	0	0	1	1	0/1	0/1	1	1

Enables TOn output.

Reverses output on match between TMn and CRn0.

Specifies initial value of TOn output F/F.

Reverses output on match between TMn and CRn1.

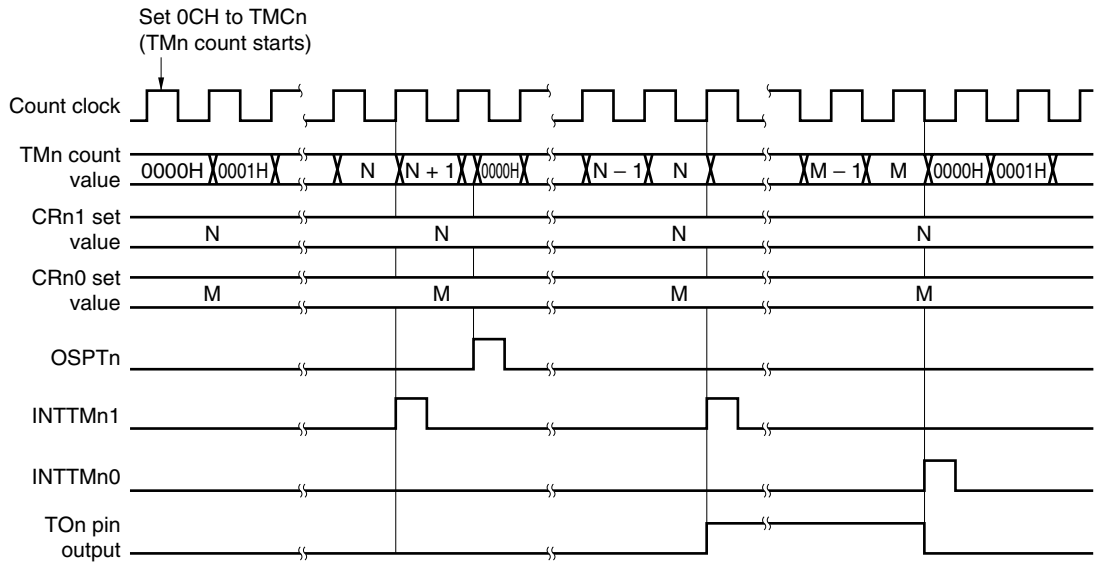
Sets one-shot pulse output mode.

Set to 1 for output.

Caution Do not set the CRn0 and CRn1 registers to 0000H.

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the one-shot pulse output function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1), 7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**, and **7.1.4 (3) 16-bit timer output control registers 0, 1 (TOC0, TOC1)**.

Figure 7-22. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution 16-bit timer register n starts operating as soon as TMCn2 and TMCn3 have been set to values other than 0, 0 (operation stop mode).

Remark n = 0, 1
N < M

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TOn pin by setting 16-bit timer mode control register n (TMCn), capture/compare control register n (CRCn), and 16-bit timer output control register n (TOCn) as shown in Figure 7-23, and by using the valid edge of the TIn0 pin as an external trigger.

The valid edge of the TIn0 pin is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn). The rising, falling, or both the rising and falling edges can be specified.

When the valid edge of the TIn0 pin is detected, the 16-bit timer/event counter is cleared and started, and the output is asserted at the count value (N) set in advance to 16-bit capture/compare register n1 (CRn1).

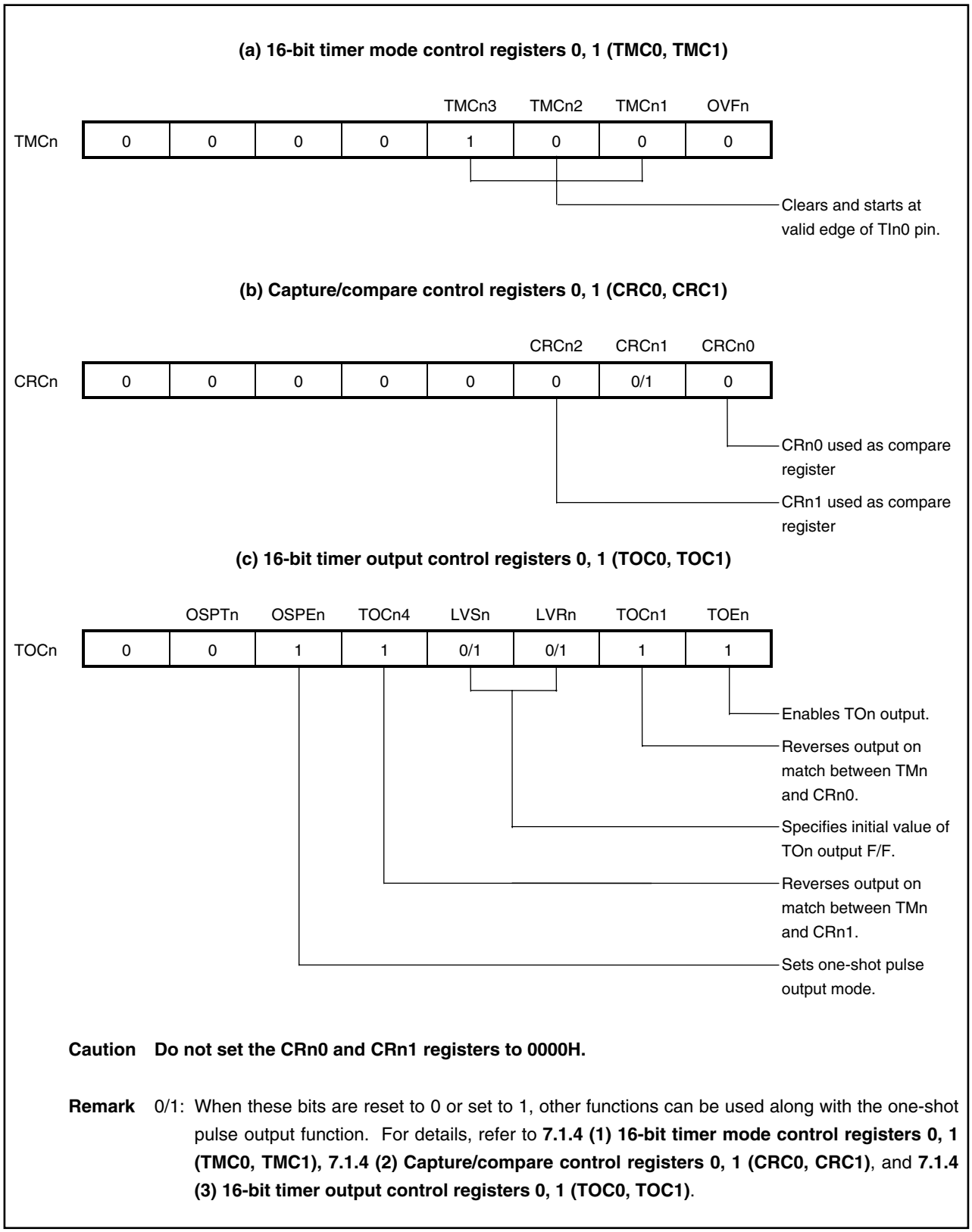
After that, the output is deasserted at the count value (M) set in advance to 16-bit capture/compare register n0 (CRn0)[Note].

Note This is an example when N < M. When N > M, output of CRn0 is asserted and output of CRn1 is deasserted.

Caution Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.

Remark n = 0, 1

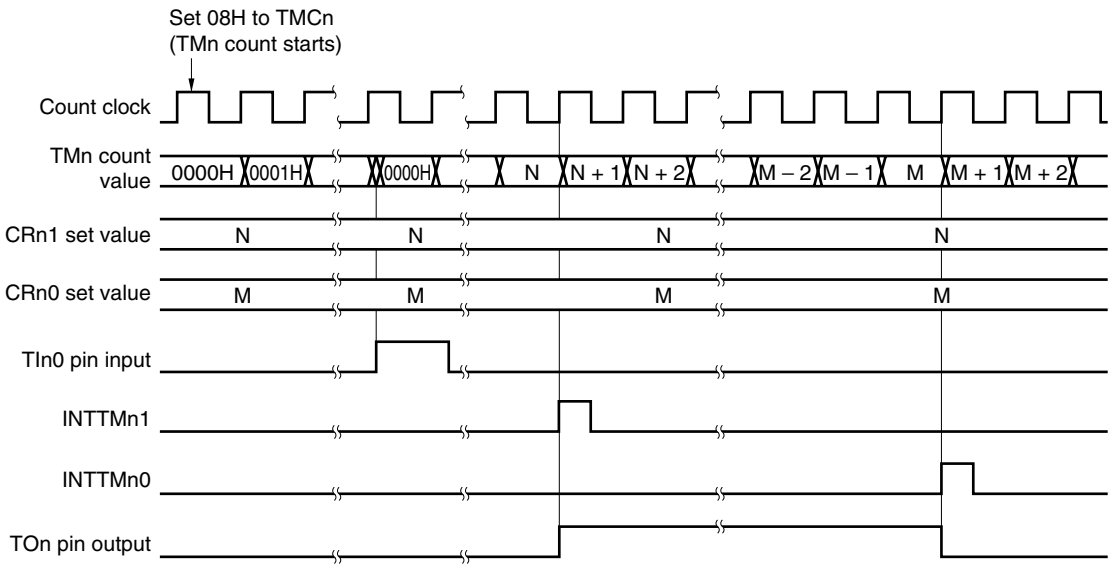
Figure 7-23. Control Register Settings for One-Shot Pulse Output with External Trigger



Caution Do not set the CRn0 and CRn1 registers to 0000H.

Remark 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the one-shot pulse output function. For details, refer to **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1), 7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**, and **7.1.4 (3) 16-bit timer output control registers 0, 1 (TOC0, TOC1)**.

Figure 7-24. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution 16-bit timer register n starts operating as soon as TMCn2 and TMCn3 have been set to values other than 0, 0 (operation stop mode).

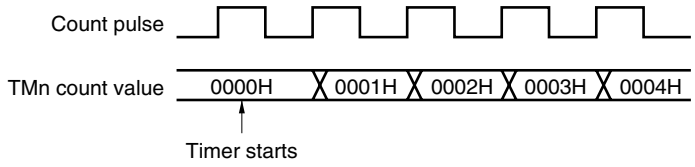
Remark n = 0, 1
N < M

7.2.7 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 16-bit timer register n (TMn) is started asynchronously to the count pulse.

Figure 7-25. Start Timing of 16-Bit Timer Register n



Remark n = 0, 1

(2) Setting 16-bit capture/compare register (in mode in which clear & start occurs on match between TMn and CRn0)

Set 16-bit capture/compare registers n0 and n1 (CRn0, CRn1) to a value other than 0000H. When using these registers as event counters, a one-pulse count operation is not possible.

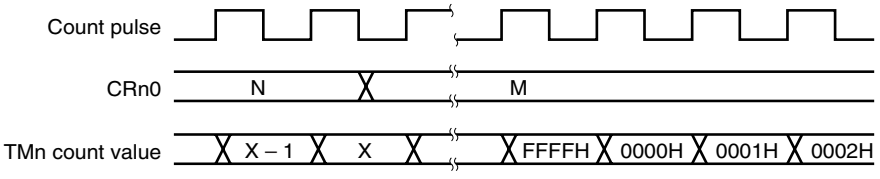
Remark n = 0, 1

(3) Setting compare register during timer count operation

If the value to which the current value of 16-bit capture/compare register n0 (CRn0) has been changed is less than the value of 16-bit timer register n (TMn), TMn continues counting, overflows, and starts counting again from 0.

If the new value of CRn0 (M) is less than the old value (N), the timer must be reset and restarted after the value of CRn0 has been changed.

Figure 7-26. Timing After Changing Compare Register During Timer Count Operation



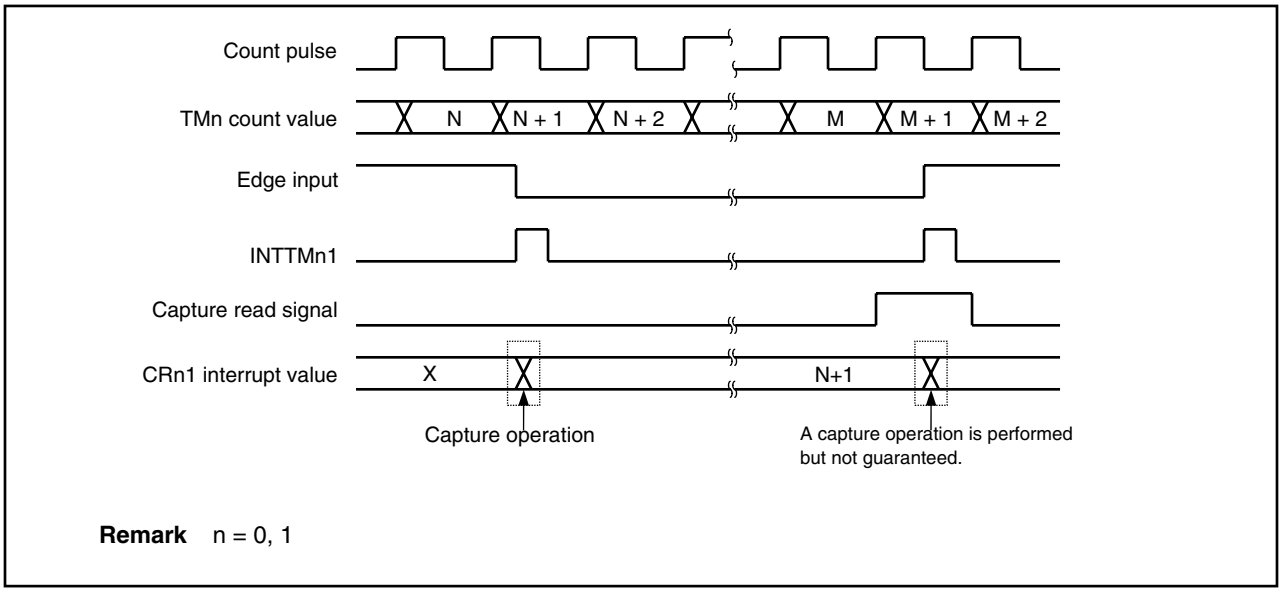
Remarks 1. N > X > M

2. n = 0, 1

(4) Data hold timing of capture register

If the valid edge is input to the TIn0 pin while 16-bit capture/compare register n1 (CRn1) is being read, CRn1 performs a capture operation, but this capture value is not guaranteed. However, the interrupt request signal (INTTMn1) is set as a result of detection of the valid edge.

★ **Figure 7-27. Data Hold Timing of Capture Register**



Remark n = 0, 1

(5) Setting valid edge

Before setting the valid edge of the TIn0 pin, stop the timer operation by resetting bits 2 and 3 (TMCn2 and TMCn3) of 16-bit timer mode control register n to 0, 0. Set the valid edge by using bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn).

Remark n = 0, 1

(6) Re-triggering one-shot pulse

(a) One-shot pulse output by software

When a one-shot pulse is being output, do not set OSPTn to 1. To output a one-shot pulse again, wait until the current one-shot pulse output is complete.

(b) One-shot pulse output with external trigger

If the external trigger occurs while a one-shot pulse is being output, it is ignored.

★ **(c) One-shot pulse output function**

When using the one-shot pulse output function of timer 0 or 1 via a software trigger, the TIn0 pin cannot be used as a general-purpose port pin.

Remark n = 0, 1

(7) Operation of OVFn flag

(a) OVFn flag set

The OVFn flag is set to 1 in the following case in addition to when the TMn register overflows:

Select the mode in which the timer is cleared and started on a match between TMn and CRn0.

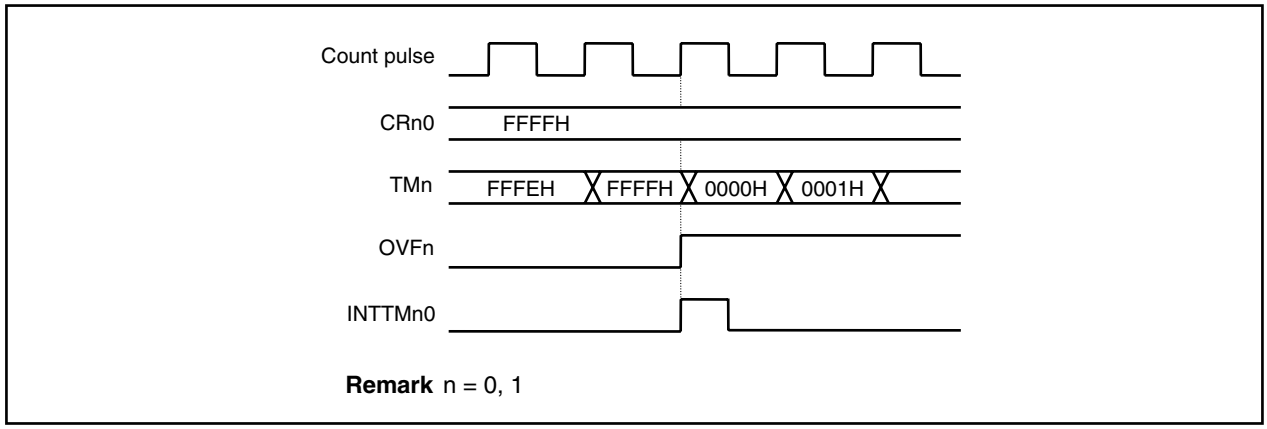
↓

Set the CRn0 register to FFFFH.

↓

When TMn is cleared from FFFFH to 0000H on a match with the CRn0 register.

Figure 7-28. Operation Timing of OVFn Flag



Remark n = 0, 1

(b) Clear OVFn flag

Even if the OVFn flag is cleared before the next count clock is counted (before TMn becomes 0001H) after TMn has overflowed, the OVFn flag is set again and the clear becomes invalid.

Remark n = 0, 1

(8) Conflict operation

(a) If the read period and capture trigger input conflict

When 16-bit capture/compare registers n0 and n1 (CRn0, CRn1) are used as capture registers, if the read period and capture trigger input conflict, the capture trigger has priority. The read data of the CRn0 and CRn1 registers is undefined.

(b) If the match timing of the write period and TMn conflict

When 16-bit capture/compare registers n0 and n1 (CRn0, CRn1) are used as capture registers, because match detection cannot be performed correctly if the match timing of the write period and 16-bit timer register n (TMn) conflict, do not write to the CRn0 and CRn1 registers close to the match timing.

Remark n = 0, 1

(9) Timer operation

(a) CRn1 capture

Even if 16-bit timer register n (TMn) is read, a capture to 16-bit capture/compare register n1 (CRn1) is not performed.

(b) Acknowledgement of TIn0 and TIn1 pins

When the timer is stopped, input signals to the TIn0 and TIn1 pins are not acknowledged, regardless of the CPU operation.

(c) One-shot pulse output

The one-shot pulse output operates correctly only in free-running mode or in clear & start mode at the valid edge of the TIn0 pin. The one-shot pulse cannot be output in the clear & start mode on a match of TMn and CRn0 because an overflow does not occur.

Remark n = 0, 1

(10) Capture operation

(a) If the valid edge of TIn0 is specified for the count clock

When the valid edge of TIn0 is specified for the count clock, the capture register with TIn0 specified as a trigger will not operate correctly.

(b) If both rising and falling edges are selected as the valid edge of TIn0, a capture operation is not performed.

(c) To capture the signals correctly from TIn0 and TIn1

The capture trigger needs a pulse longer than twice the count clock selected by prescaler mode registers n0 and n1 (PRMn0, PRMn1) in order to correctly capture the signals from TIn1 and TIn0.

(d) Interrupt request input

Although a capture operation is performed a the falling edge of the count clock, interrupt request inputs (INTTMn0, INTTMn1) are generated at the rising edge of the next count clock.

Remark n = 0, 1

(11) Compare operation

(a) When rewriting CRn0 and CRn1 during timer operation

When rewriting 16-bit timer capture/compare registers n0 and n1 (CRn0, CRn1), if the value is close to or larger than the timer value, the match interrupt request generation or clear operation may not be performed correctly.

(b) When CRn0 and CRn1 are set to compare mode

When CRn0 and CRn1 are set to compare mode, they do not perform a capture operation even if a capture trigger is input.

Remark n = 0, 1

(12) Edge detection

(a) When the TIn0 or TIn1 pin is high level immediately after a system reset

When the TIn0 or TIn1 pin is high level immediately after a system reset, if the valid edge of the TIn0 or TIn1 pin is specified as the rising edge or both rising and falling edges, and the operation of 16-bit timer/counter n (TMn) is then enabled, the rising edge will be detected immediately. Care is therefore needed when the TIn0 or TIn1 pin is pulled up. However, when operation is enabled after being stopped, the rising or falling edge is not detected.

(b) Sampling clock for noise elimination

The sampling clock for noise elimination differs depending on whether the TIn0 valid edge is used as a count clock or a capture trigger. The former is sampled by $f_{XX}/2$, and the latter is sampled by the count clock selected using prescaler mode register n0 or n1 (PRMn0, PRMn1). Detecting the valid edge can eliminate short pulse width noise because a capture operation is performed only after the valid edge is sampled and a valid level is detected twice.

Remark n = 0, 1

7.3 8-Bit Timers (TM2 to TM5)

★ 7.3.1 Outline

- 8-bit compare registers: 4 (CRn0)
  Can be used as 16-bit compare registers by connecting in cascade (2 max.).
- Compare match/overflow interrupt request signal (INTTMn) output enabled
- Event input (TIn) count enabled
- Timer outputs that operate on match detection: 1 each (TOn)
  If using the P26/TI2/TO2, P27/TI3/TO3, P36/TI4/TO4, and P37/TI5/TO5 pins as the TO2 to TO5 pins (timer outputs), set the value of ports 2 and 3 (P2, P3) to 0 (low-level output) and the value of the port 3 mode register (PM3) to 0 (port output mode). The logical sum (OR) of the output value of the port and the timer is output. Since the TOn pin and TIn pin share a pin, one or other of these functions (but not both) can be used.

Remark n = 2 to 5

7.3.2 Functions

8-bit timer n has the following two modes (n = 2 to 5).

- Mode using timer alone (individual mode)
- Mode using cascade connection (16-bit resolution: cascade connection mode)

Caution When used without cascade connection, do not access to the following registers.
- **16-bit counters (TM23, TM45)**
- **16-bit compare registers (CR23, CR45)**

The two modes are described next.

(1) Mode using timer alone (individual mode)

The timer operates as an 8-bit timer/event counter.
It can have the following functions.

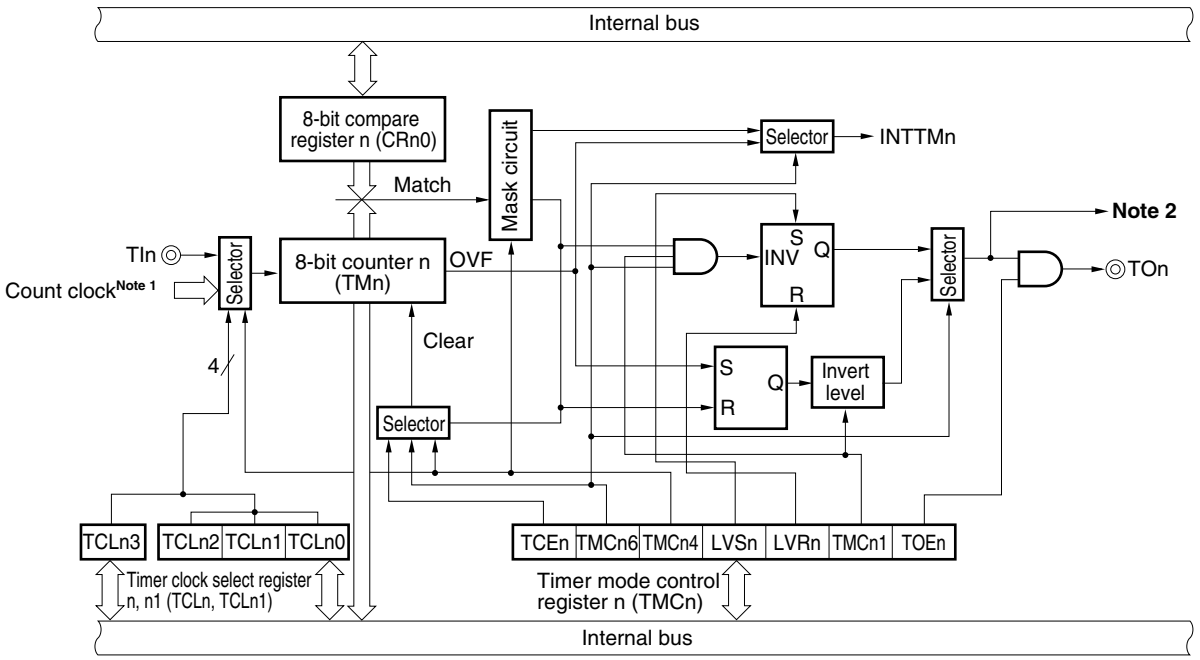
- Interval timer
- External event counter
- Square wave output
- PWM output

(2) Mode using cascade connection (16-bit resolution: cascade connection mode)

The timer operates as a 16-bit timer/event counter by connecting TM2 and TM3 or TM4 and TM5 in cascade.
It can have the following functions.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

Figure 7-29. Block Diagram of TM2 to TM5



Notes **1.** The count clock is set by the TCLn or TCLn1 register.

2. Serial interface clocks (TM2 and TM3 only)

Remarks 1. " ——◎" is a signal that can be directly connected to a port.

2. n = 2 to 5

7.3.3 Configuration

Timers 2 to 5 consist of the following hardware.

Table 7-5. Configuration of Timers 2 to 5

Item	Configuration
Timer registers	8-bit counter 2 to 5 (TM2 to TM5) 16-bit counter 23 and 45 (TM23, TM45): Only when connecting in cascade
Registers	8-bit compare register 2 to 5 (CR20 to CR50) 16-bit compare register 23 and 45 (CR23, CR45): Only when connecting in cascade
Timer outputs	TO2 to TO5
Control registers	Timer clock select register 2 to 5, 21 to 51 (TCL2 to TCL5, TCL21 to TCL51) 8-bit timer mode control register 2 to 5 (TMC2 to TMC5)

(1) 8-bit counters 2 to 5 (TM2 to TM5)

TMn 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.

Through cascade connection, TM2 and TM3, and TM4 and TM5 can be used as 16-bit timers.

When using TMm and TMm + 1 in cascade as a 16-bit timer, the timer can be read using a 16-bit memory manipulation instruction. However, because these timers are connected by an internal 8-bit bus, TMm and TMm + 1 must be read twice. Therefore, read these timers twice and compare the values, bearing in mind that the reading occurs during a count change.

When the count is read out during operation, the count clock input temporarily stops and the count is read at that time. In the following cases, the count becomes 00H.

(1) $\overline{\text{RESET}}$ is input.
(2) TCEn is cleared.
(3) TMn and CRn0 match in the clear and start mode that occurs when TMn and CRn0 match.

Caution When connected in cascade, these registers become 00H even when TCEn in the lowest timer (TM2, TM4) is cleared.

Remark n = 2 to 5
m = 2, 4

(2) 8-bit compare registers 2 to 5 (CR20 to CR50)

The CRn0 register is set by an 8-bit memory manipulation instruction.

The value set in CRn0 is always compared to the count value in 8-bit counter n (TMn). If the two values match, an interrupt request (INTTMn) is generated (except in the PWM mode).

The value of CRn0 can be set in the range of 00H to FFH, and can be written during counting.

When using TMm and TMm + 1 in cascade as a 16-bit timer, CRm0 and CR (m + 1) 0 operate as a 16-bit compare register that is set by a 16-bit memory manipulation instruction. The counter and register values are compared in 16-bit lengths, and if they match, an interrupt request (INTTMm) is generated. Because the interrupt request INTTMm + 1 is also generated at this time, be sure to mask interrupt request INTTMm + 1 when using TMm and TMm + 1 in cascade connection.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Caution If data is set in a cascade connection, always set after stopping the timer.

Remark n = 2 to 5
m = 2, 4

7.3.4 Timer n control register

The following two types of registers control timer n.

- Timer clock select registers n and n1 (TCLn, TCLn1)
- 8-bit timer mode control register n (TMCn)

(1) Timer clock select registers 2 to 5, 21 to 51 (TCL2 to TCL5 and TCL21 to TCL51)

These registers set the count clock of timer n.

TCLn and TCLn1 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Remark n = 2 to 5

(1/2)

After reset: 00H R/W Address: FFFFF244H, FFFFF254H

	7	6	5	4	3	2	1	0
TCLn	0	0	0	0	0	TCLn2	TCLn1	TCLn0

(n = 2, 3)

After reset: 00H R/W Address: FFFFF24EH, FFFFF25EH

	7	6	5	4	3	2	1	0
TCLn1	0	0	0	0	0	0	0	TCLn3

(n = 2, 3)

TCLn3	TCLn2	TCLn1	TCLn0	Count Clock Selection				
				Count Clock	f_{XX}			
					20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	0	TIn falling edge	–	–	–	–
0	0	0	1	TIn rising edge	–	–	–	–
0	0	1	0	$f_{XX}/4$	200 ns	235 ns	400 ns	2 μs
0	0	1	1	$f_{XX}/8$	400 ns	471 ns	800 ns	4 μs
0	1	0	0	$f_{XX}/16$	800 ns	941 ns	1.6 μs	8 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	1.9 μs	3.2 μs	16 μs
0	1	1	0	$f_{XX}/128$	6.4 μs	7.5 μs	12.8 μs	64 μs
0	1	1	1	$f_{XX}/512$	25.6 μs	30.1 μs	51.2 μs	256 μs
1	0	0	0	Setting prohibited	–	–	–	–
1	0	0	1	Setting prohibited	–	–	–	–
1	0	1	0	$f_{XX}/64$	3.2 μs	3.8 μs	6.4 μs	32 μs
1	0	1	1	$f_{XX}/256$	12.8 μs	15.1 μs	25.6 μs	128 μs
1	1	0	0	Setting prohibited	–	–	–	–
1	1	0	1	Setting prohibited	–	–	–	–
1	1	1	0	Setting prohibited	–	–	–	–
1	1	1	1	Setting prohibited	–	–	–	–

Cautions 1. When TCLn and TCLn1 are overwritten by different data, write after temporarily stopping the timer.

2. Always set bits 3 to 7 of TCLn and bits 1 to 7 of TCLn1 to 0.

Remark When connected in cascade, the settings of TCL33 to TCL30 of TM3 are invalid.

(2/2)

After reset: 00H R/W Address: FFFFF264H, FFFFF274H

	7	6	5	4	3	2	1	0
TCLn	0	0	0	0	0	TCLn2	TCLn1	TCLn0

(n = 4, 5)

After reset: 00H R/W Address: FFFFF26EH, FFFFF27EH

	7	6	5	4	3	2	1	0
TCLn1	0	0	0	0	0	0	0	TCLn3

(n = 4, 5)

TCLn3	TCLn2	TCLn1	TCLn0	Count Clock Selection				
				Count Clock	f_{XX}			
					20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	0	TIn falling edge	–	–	–	–
0	0	0	1	TIn rising edge	–	–	–	–
0	0	1	0	$f_{XX}/4$	200 ns	235 ns	400 ns	2 μs
0	0	1	1	$f_{XX}/8$	400 ns	471 ns	800 ns	4 μs
0	1	0	0	$f_{XX}/16$	800 ns	941 ns	1.6 μs	8 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	1.9 μs	3.2 μs	16 μs
0	1	1	0	$f_{XX}/128$	6.4 μs	7.5 μs	12.8 μs	64 μs
0	1	1	1	f_{XT} (Subclock)	30.5 μs	30.5 μs	30.5 μs	30.5 μs
1	0	0	0	Setting prohibited	–	–	–	–
1	0	0	1	Setting prohibited	–	–	–	–
1	0	1	0	$f_{XX}/64$	3.2 μs	3.8 μs	6.4 μs	32 μs
1	0	1	1	$f_{XX}/256$	12.8 μs	15.1 μs	25.6 μs	128 μs
1	1	0	0	Setting prohibited	–	–	–	–
1	1	0	1	Setting prohibited	–	–	–	–
1	1	1	0	Setting prohibited	–	–	–	–
1	1	1	1	Setting prohibited	–	–	–	–

Cautions 1. When TCLn and TCLn1 are overwritten by different data, write after temporarily stopping the timer.

2. Always set bits 3 to 7 of TCLn and bits 1 to 7 of TCLn1 to 0.

Remark When connected in cascade, the settings of TCL53 to TCL50 of TM5 are invalid.

(2) 8-bit timer mode control registers 2 to 5 (TMC2 to TMC5)

The TMCn register makes the following six settings.

(1) Controls counting by 8-bit counter n (TMn)
(2) Selects the operating mode of 8-bit counter n (TMn)
(3) Selects the individual mode or cascade connection mode
(4) Sets the state of the timer output flip-flop
(5) Controls the timer flip-flop or selects the active level in the PWM (free-running) mode
(6) Controls timer output

TMCn is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 04H (the hardware status is initialized to 04H, but 00H is read when read).

After reset: 04H R/W Address: TMC2 FFFFF246H TMC4 FFFFF266H
TMC3 FFFFF256H TMC5 FFFFF276H

	7	6	5	4	3	2	1	0
TMCn	TCEn	TMCn6	0	TMCn4	LVSn	LVRn	TMCn1	TOEn

(n = 2 to 5)

TCEn	TMn Count Operation Control
0	Counting is disabled after the counter is cleared to 0 (prescaler disabled)
1	Start count operation

TMCn6	TMn Operating Mode Selection
0	Clear and start mode when TMn and CRn0 match
1	PWM (free-running) mode

TMCn4	Individual Mode or Cascade Connection Mode Selection
0	Individual mode (fixed to 0 when n = 2, 4)
1	Cascade connection mode (connection to lower timer)

LVSn	LVRn	Setting State of Timer Output Flip-Flop
0	0	Not change
0	1	Reset timer output flip-flop to 0
1	0	Set timer output flip-flop to 1
1	1	Setting prohibited

TMCn1	Other than PWM (Free-Running) Mode (TMCn6 = 0)	PWM (Free-Running) Mode (TMCn6 = 1)
	Control of Timer F/F	Selection of Active Level
0	Disable inversion operation	Active high
1	Enable inversion operation	Active low

TOEn	Timer Output Control
0	Disable output (port mode)
1	Enable output

★ **Caution Stop the timer operation temporarily to rewrite the TMCn4 and TMCn6 bits.**

Remarks 1. In the PWM mode, the PWM output is set to the inactive level by TCEn = 0.

2. If the LVSn and LVRn bits are read after setting data, 0 is read.

7.4 8-Bit Timer Operation

7.4.1 Operation as interval timer (8-bit operation)

The timer operates as an interval timer that repeatedly generates interrupts at the interval of the count preset by 8-bit compare register n (CRn0).

If the count in 8-bit counter n (TMn) matches the value set in CRn0, the value of TMn is cleared to 0 and continues counting. At the same time, an interrupt request signal (INTTMn) is generated.

The TMn count clock can be selected by bits 0 to 2 (TCLn0 to TCLn2) of timer clock select register n (TCLn) and by bit 0 (TCLn3) of timer clock select register n1 (TCLn1) (n = 2 to 5).

Setting method

(1) Set each register.
- TCLn, TCLn1: Select the count clock.
- CRn0: Compare value
- TMCn: Selects the clear and start mode when TMn and CRn0 match.
  (TMCn = 0000xxx0B, x is don't care)

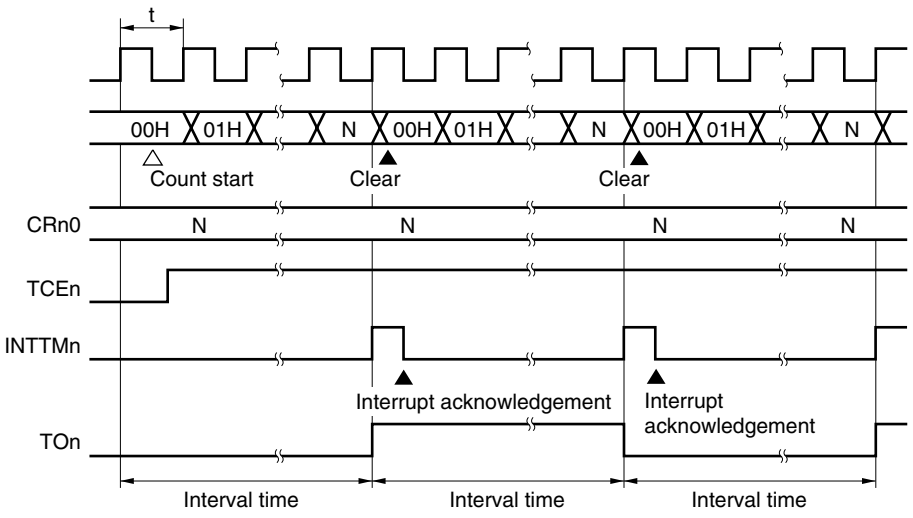
(2) When TCEn = 1 is set, counting starts.

(3) When the values of TMn and CRn0 match, INTTMn is generated (TMn is cleared to 00H).

(4) Then, INTTMn is repeatedly generated at the same interval. When counting stops, set TCEn = 0.

Figure 7-30. Timing of Interval Timer Operation (1/3)

Basic operation

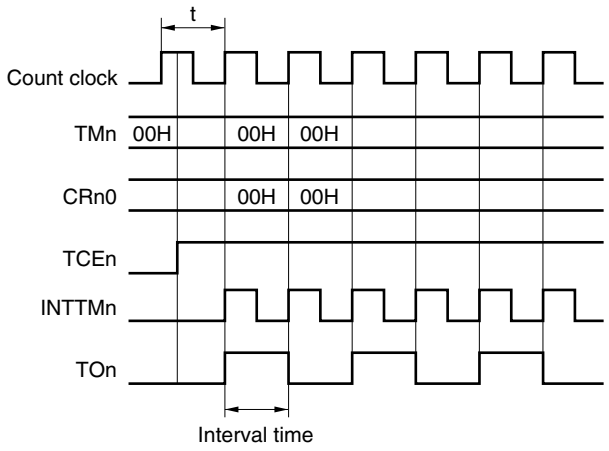


Remarks 1. Interval time = (N + 1) × t; N = 00H to FFH

2. n = 2 to 5

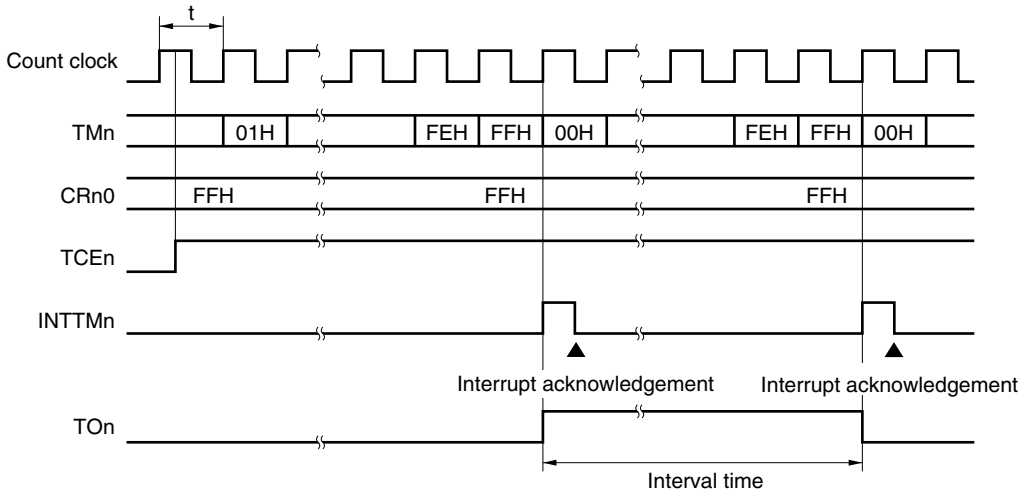
Figure 7-30. Timing of Interval Timer Operation (2/3)

When CRn0 = 00H



Remark n = 2 to 5

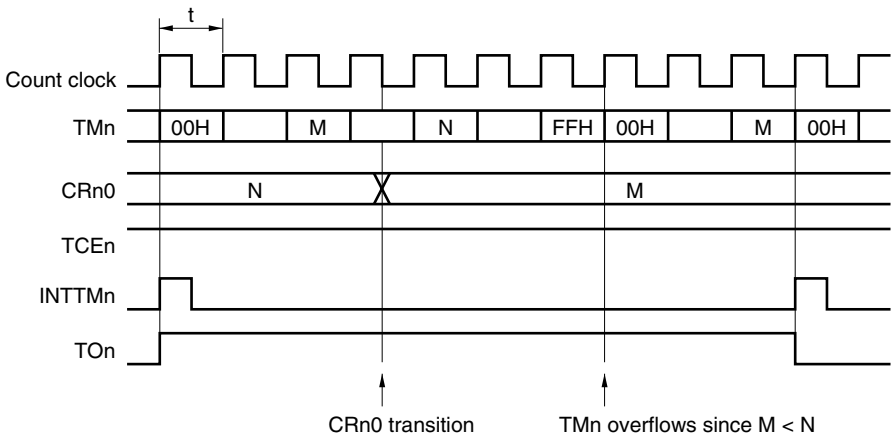
When CRn0 = FFH



Remark n = 2 to 5

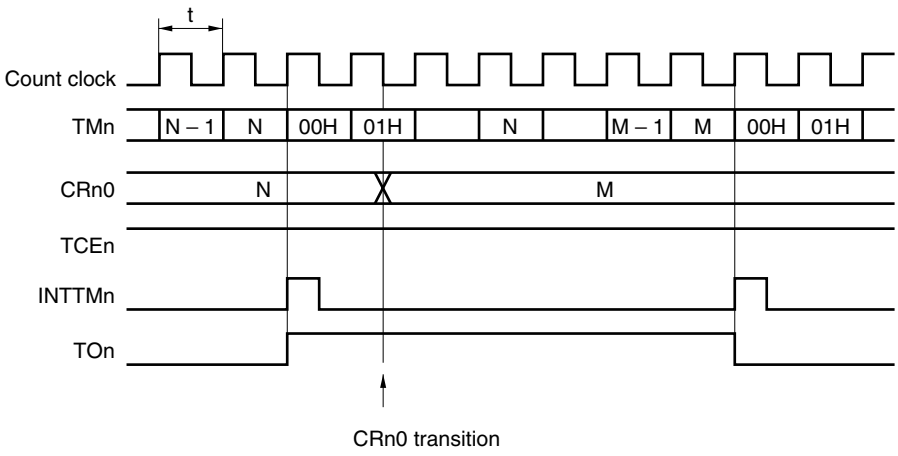
Figure 7-30. Timing of Interval Timer Operation (3/3)

Operated by CRn0 transition (M < N)



Remark n = 2 to 5

Operated by CRn0 transition (M > N)



Remark n = 2 to 5

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses that are input to TIn.

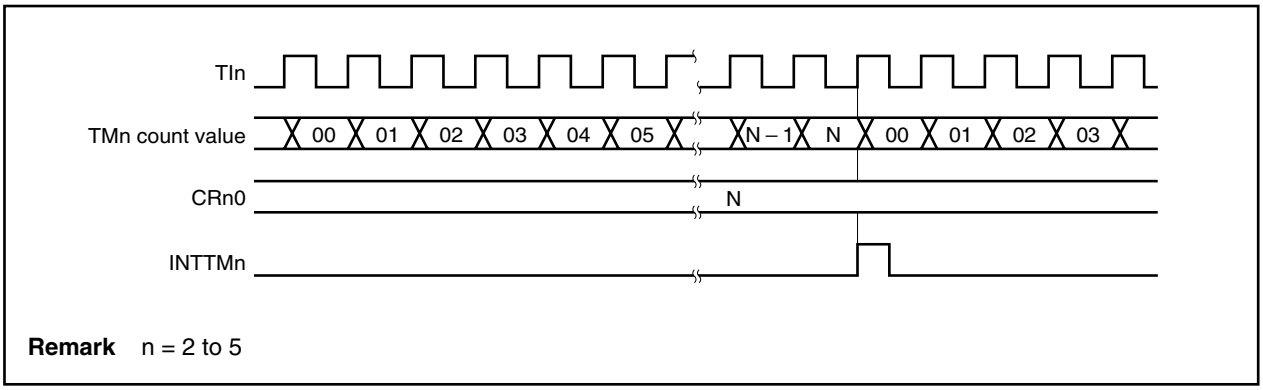
Each time a valid edge specified by timer clock select register n, n1 (TCLn, TCLn1) is input, TMn is incremented. The edge setting can be selected as either the rising or falling edge.

If the total of TMn and the value of 8-bit compare register n (CRn0) match, TMn is cleared to 0 and an interrupt request signal (INTTMn) is generated.

INTTMn is generated each time the TMn value matches the CRn0 value.

Remark n = 2 to 5

Figure 7-31. Timing of External Event Counter Operation (When Rising Edge Is Set)



Remark n = 2 to 5

7.4.3 Operation as square wave output (8-bit resolution)

A square wave with any frequency is output at the interval preset by 8-bit compare register n (CRn0).

By setting bit 0 (TOEn) of 8-bit timer mode control register n (TMCn) to 1, the output state of TOn is inverted with the count preset in CRn0 as the interval. Therefore, a square wave output with any frequency (duty factor = 50%) is possible.

Setting method

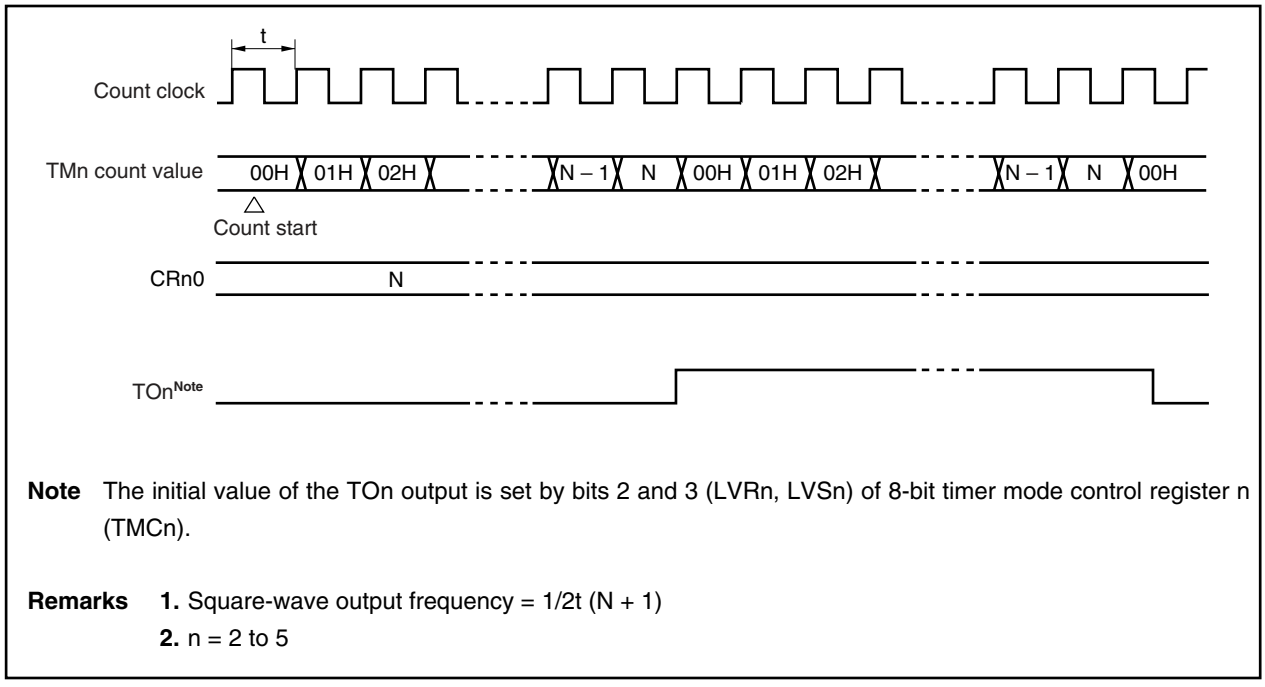
(1) Set the registers.
- Set the port latch and port mode register to 0
- TCLn, TCLn1: Select the count clock
- CRn0: Compare value
- TMCn: Clear and start mode entered when TMn and CRn0 match

LVSn	LVRn	Setting State of Timer Output Flip-flop
1	0	High-level output
0	1	Low-level output

Inversion of timer output flip-flop enabled
Timer output enabled → TOEn = 1

(2) When TCEn = 1 is set, the counter starts operating.
(3) If the values of TMn and CRn0 match, the timer output flip-flop inverts. Also, INTTMn is generated and TMn is cleared to 00H.
(4) Then, the timer output flip-flop is inverted at the same interval to output a square wave from TOn.

Figure 7-32. Timing of Square Wave Output Operation



Note The initial value of the TOn output is set by bits 2 and 3 (LVRn, LVSn) of 8-bit timer mode control register n (TMCn).

Remarks **1.** Square-wave output frequency = 1/2t (N + 1)
2. n = 2 to 5

7.4.4 Operation as 8-bit PWM output

By setting bit 6 (TMCn6) of 8-bit timer mode control register n (TMCn) to 1, the timer operates as a PWM output.

Pulses with the duty factor determined by the value set to 8-bit compare register n (CRn0) are output from TOn.

Set the width of the active level of the PWM pulse to CRn0. The active level can be selected by bit 1 (TMCn1) of TMCn.

The count clock can be selected by bits 0 to 2 (TCLn0 to TCLn2) of timer clock select register n (TCLn) and by bit 0 (TCLn3) of timer clock select register n1 (TCLn1).

The PWM output can be enabled and disabled by bit 0 (TOEn) of TMCn.

Caution CRn0 can be rewritten only once in one cycle while in the PWM mode.

Remark n = 2 to 5

(1) Basic operation of the PWM output

Setting method

(1) Set the port latch and port mode register n to 0.
(2) Set the active level width in 8-bit compare register n (CRn0).
(3) Select the count clock using timer clock select register n, n1 (TCLn, TCLn1).
(4) Set the active level in bit 1 (TMCn1) of TMCn.
(5) If bit 7 (TCEn) of TMCn is set to 1, counting starts. When counting stops, set TCEn to 0.

PWM output operation

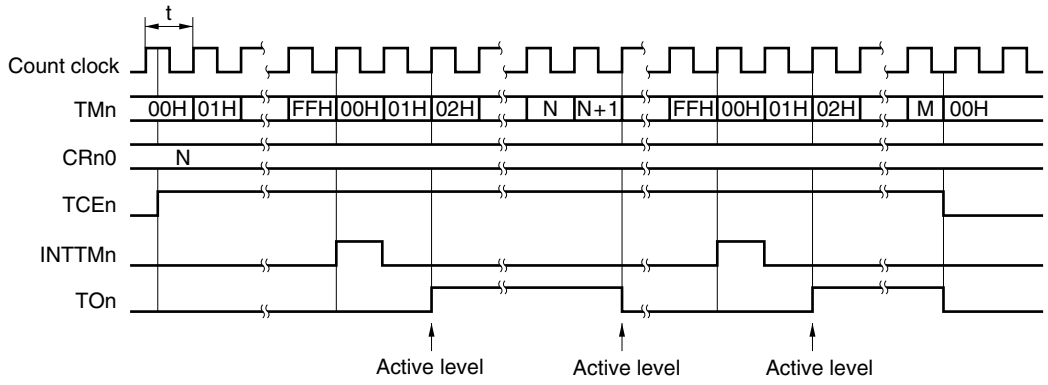
(1) When counting starts, the PWM output (output from TOn) outputs the inactive level until an overflow occurs.
(2) When the overflow occurs, the active level specified in step (1) in the setting method is output. The active level is output until CRn0 and the count of 8-bit counter n (TMn) match.
(3) The PWM output after CRn0 and the count match is the inactive level until an overflow occurs again.
(4) Steps (2) and (3) repeat until counting stops.
(5) If counting is stopped by TCEn = 0, the PWM output goes to the inactive level.

Remark n = 2 to 5

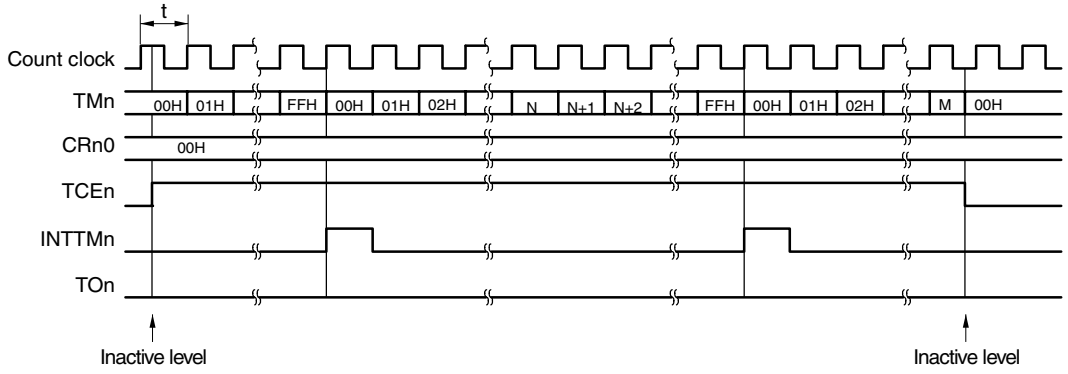
(a) Basic operation of PWM output

Figure 7-33. Timing of PWM Output

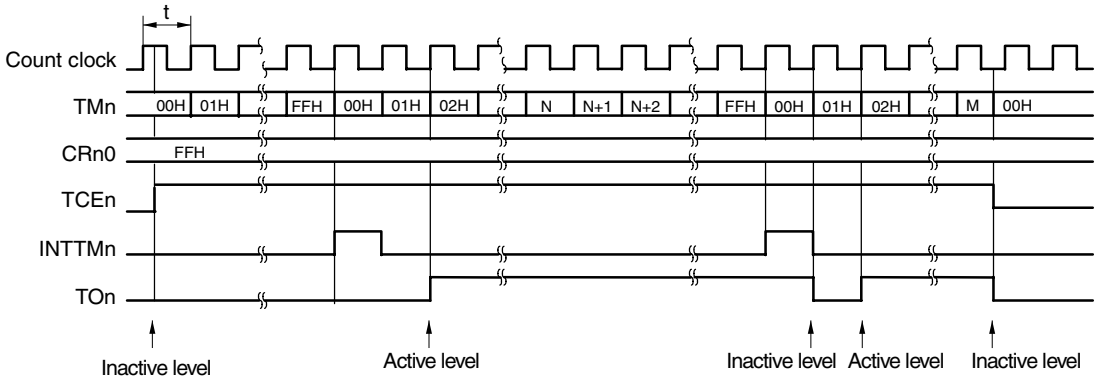
Basic operation (active level = H)



When CRn0 = 00H



When CRn0 = FFH



Remarks 1. PWM frequency = $1/2^8t$

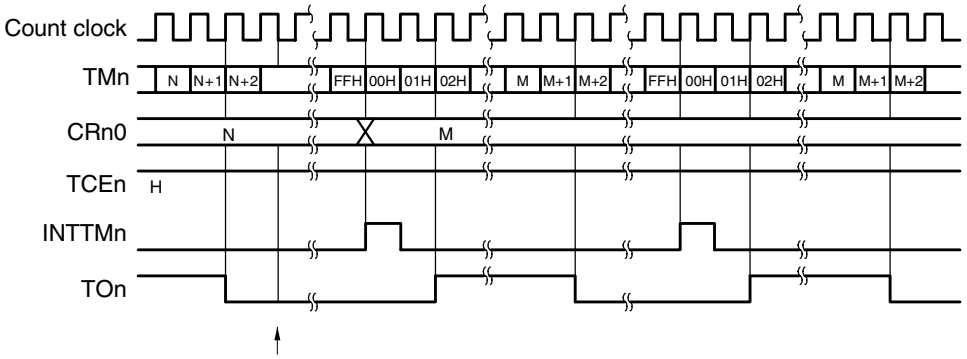
Duty = $N/2^8$

2. n = 2 to 5

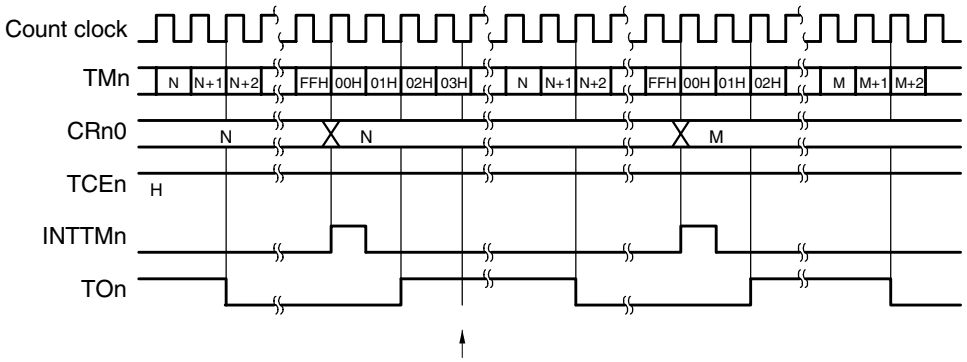
(b) Operation based on CRn0 transitions

Figure 7-34. Timing of Operation Based on CRn0 Transition

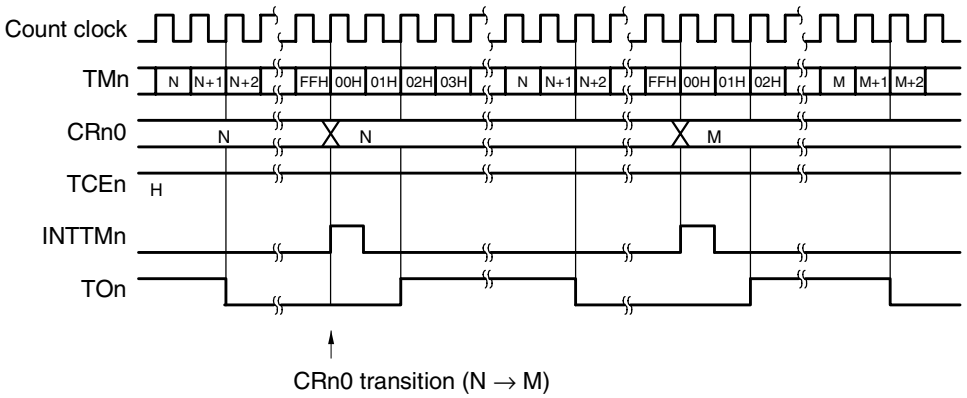
When the CRn0 value changes from N to M before TMn overflows



When the CRn0 value changes from N to M after TMn overflows



When the CRn0 value changes from N to M during two clocks (00H, 01H) immediately after TMn overflows



Remark n = 2 to 5

7.4.5 Operation as interval timer (16 bits)

(1) Cascade connection (16-bit timer) mode

The V850/SA1 provides 16-bit registers that can be used only when connected in cascade.
The following registers are available.

TM2 to TM3 cascade connection:	16-bit counter TM23 (Address: FFFFF24AH)
	16-bit compare register CR23 (Address: FFFFF24CH)
TM4 to TM5 cascade connection:	16-bit counter TM45 (Address: FFFFF26AH)
	16-bit compare register CR45 (Address: FFFFF26CH)

By setting bit 4 (TMCm4) of 8-bit timer mode control register m (TMCm) to 1, the timer enters the timer/counter mode with 16-bit resolution (m = 3, 5).
The timer operates as an interval timer by repeatedly generating interrupts (n = 2 to 5) with the count preset in 8-bit compare register n (CRn0) as the interval.
The following is an explanation of how to use TM2 and TM3. Substitute TM4 and TM5 for TM2 and TM3 as appropriate when using TM4 and TM5.

Example of setting method (when TM2 and TM3 are connected in cascade)

<1> Set each register.
- TCL20, TCL21: Select the count clock for TM2 (TM3 does not need to be set in a cascade connection)
- CR20, CR30: Compare values (each compare value can be set from 00H to FFH)
- TMC2: Selects clear and start mode when TM2 and CR20 match (x: don't care)

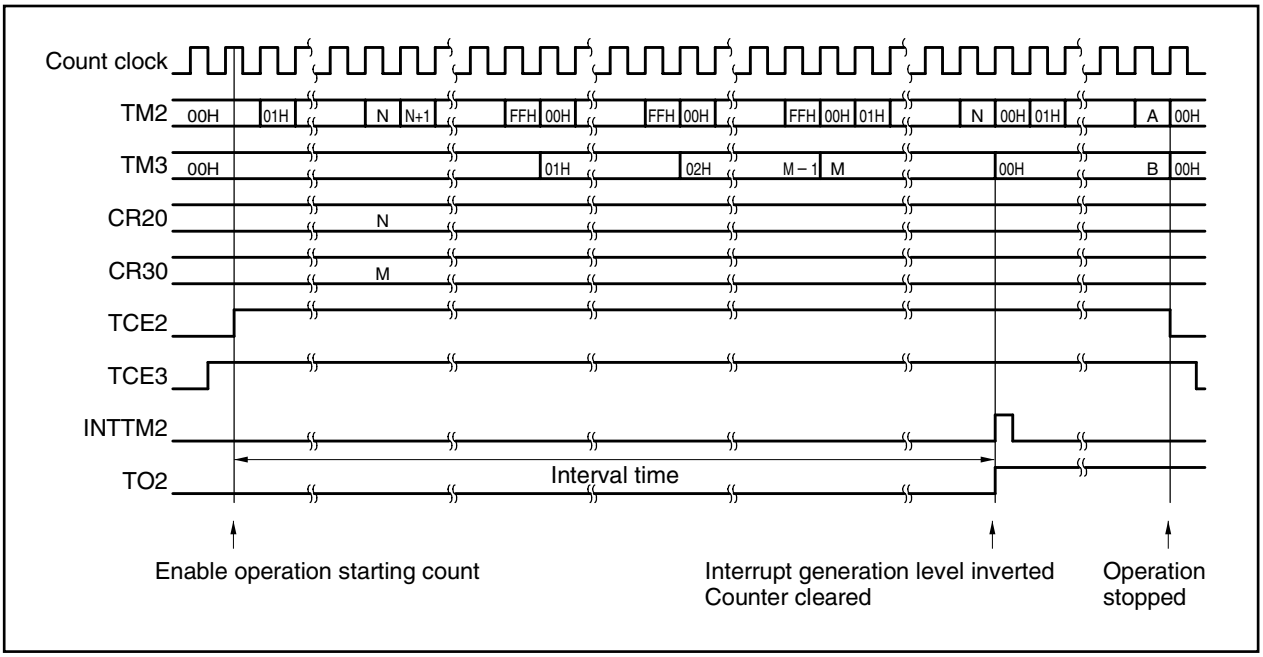
$$\begin{cases} \text{TM2} \rightarrow \text{TMC2} = 0000\text{xxx0B} \\ \text{TM3} \rightarrow \text{TMC3} = 0001\text{xxx0B} \end{cases}$$

<2> Start the count operation by first setting the TCE3 bit of TMC3 to 1, and then setting the TCE2 bit of TMC2 to 1.
<3> If a match occurs between cascade-connected timer TM2 and CR20, the INTTM2 of TM2 will be generated (TM2 and TM3 are cleared to 00H).
<4> IMTTM2 will then be generated repeatedly at the same interval.

Cautions 1. To change the set value of the compare register (CR23) while the 8-bit timers (TM2, TM3) are connected in cascade and being used as a 16-bit timer (TM23), change the CR23 value after stopping the count operation of each of the 8-bit timers connected in cascade. If the CR23 value is changed without stopping the timers, the value of the higher 8 bits (TM3) will be undefined.
2. If the count value of the higher timer (TM3) matches CR30, the higher timer (TM3) interrupt request (INTTM3) will be generated, even when the timers are being used in a cascade connection. TM3 must therefore always be masked to disable interrupts.
3. Set the TCE3 bit of TMC3 before setting the TCE2 bit of TMC2.
4. Restarting and stopping the count is possible just by setting the TCE2 bit of TMC2 to 1 or 0 respectively.

The following shows a timing example of the cascade connection mode with 16-bit resolution.

Figure 7-35. Cascade Connection Mode with 16-Bit Resolution

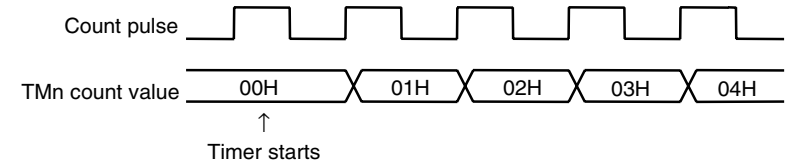


7.4.6 Cautions

(1) Error when timer starts

An error of up to 1 clock occurs in the time until the match signal is generated after the timer starts. The reason is that 8-bit counter n (TMn) starts asynchronously to the count pulse.

Figure 7-36. Start Timing of Timer n

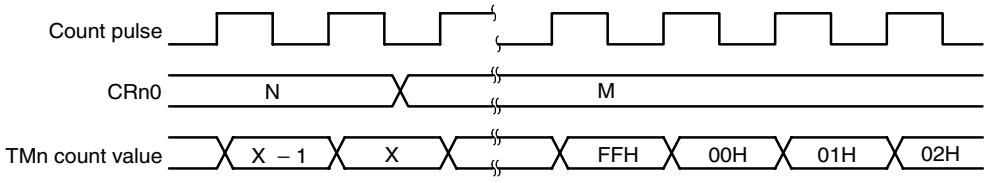


Remark n = 2 to 5

(2) Operation after compare register is changed while timer is counting

If the value after 8-bit compare register n (CRn0) changes is less than the value of the 8-bit timer register (TMn), counting continues, overflows, and starts again from 0. Consequently, when the value after CRn0 changes (M) is less than the value before the change (N) and less than the count value of the TMn register, the timer must restart after CRn0 changes (n = 2 to 5).

Figure 7-37. Timing After Compare Register Changes During Timer Count Operation



Remarks 1. N > X > M

2. n = 2 to 5

Caution Except when the TIn input is selected, always set TCEn = 0 before setting the stop state.

(3) TMn read out during timer operation

Since reading out TMn during operation occurs while the selected clock is temporarily stopped, select a high- or low-level waveform that is longer than the selected clock (n = 2 to 5).

CHAPTER 8 WATCH TIMER

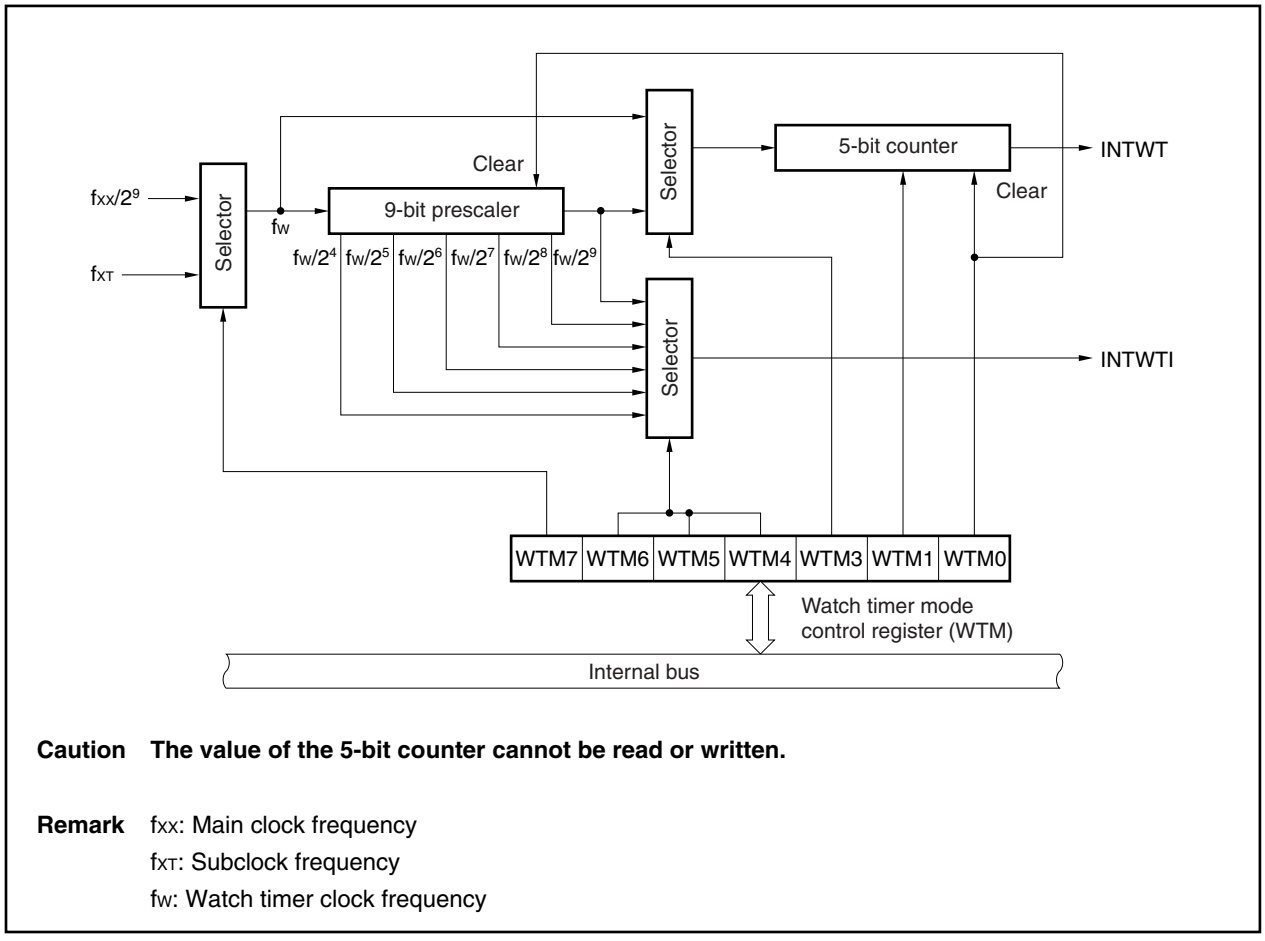
8.1 Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and interval timer functions can be used at the same time.
The block diagram of the watch timer is shown below.

Figure 8-1. Block Diagram of Watch Timer



Caution The value of the 5-bit counter cannot be read or written.

Remark fXX: Main clock frequency
fXT: Subclock frequency
fW: Watch timer clock frequency

(1) Watch timer

The watch timer generates an interrupt request signal (INTWT) at time intervals of 0.5 seconds by using the main clock or subclock.

(2) Interval timer

The watch timer generates an interrupt request (INTWTI) at time intervals specified in advance.

Table 8-1. Interval Time of Interval Timer

Interval Time	f_W = 32.768 kHz
2^4 1/f_W	488 s
2^5 1/f_W	977 s
2^6 1/f_W	1.95 ms
2^7 1/f_W	3.91 ms
2^8 1/f_W	7.81 ms
2^9 1/f_W	15.6 ms

Remark f_W: Watch timer clock frequency

8.2 Configuration

The watch timer consists of the following hardware.

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits 1
Prescaler	9 bits 1
Control register	Watch timer mode control register (WTM)

8.3 Watch Timer Control Register

The watch timer mode control register (WTM) controls the watch timer.

(1) Watch timer mode control register (WTM)

This register enables or disables the count clock and operation of the watch timer, sets the interval time of the prescaler, controls the operation of the 5-bit counter, and sets the interrupt time of the watch timer.

WTNM is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears WTM to 00H.

After reset: 00H R/W Address: FFFFF360H

	7	6	5	4	3	2	1	0
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	0	WTM1	WTM0

WTM7	Selection of Counter Clock of Watch Timer
0	$f_{XX}/2^9$ (main clock)
1	f_{XT} (subclock)

WTM6	WTM5	WTM4	Selection of Interval Time of Prescaler
0	0	0	$2^4/f_W$ (488 s)
0	0	1	$2^5/f_W$ (977 s)
0	1	0	$2^6/f_W$ (1.95 ms)
0	1	1	$2^7/f_W$ (3.91 ms)
1	0	0	$2^8/f_W$ (7.81 ms)
1	0	1	$2^9/f_W$ (15.6 ms)
Other than above			Setting prohibited

WTM3	Selection of Interrupt Time of Watch Timer
0	$2^{14}/f_W$ (0.5 s)
1	$2^5/f_W$ (977 s)

WTM1	Control of Operation of 5-Bit Counter
0	Clears after operation stops
1	Starts

WTM0	Watch Timer Operation Enable
0	Operation stopped (enabled both prescaler and timer cleared)
1	Operation

Remarks 1. f_W: Watch timer clock frequency ($f_{XX}/2^9$ or f_{XT})

2. Values in parentheses apply when f_W = 32.768 kHz.

8.4 Operation

8.4.1 Operation as watch timer

The watch timer operates at time intervals of 0.5 seconds with the subclock (32.768 kHz) or main clock (16.777 MHz).

The watch timer generates an interrupt request at fixed time intervals.

The count operation of the watch timer is started when bits 0 (WTM0) and 1 (WTM1) of the watch timer mode control register (WTM) are set to 1. When these bits are cleared to 0, the 9-bit prescaler and 5-bit counter are cleared, and the watch timer stops the count operation.

The watch timer clears the 5-bit counter by setting the WTM1 bit to 0. At this time, an error of up to 15.6 ms may occur.

The interval timer can be cleared by setting the WTM0 bit to 0. However, because the 5-bit counter is cleared at the same time, an error of up to 0.5 seconds may occur when the watch timer overflows (INTWT).

8.4.2 Operation as interval timer

The watch timer can also be used as an interval timer that repeatedly generates an interrupt at intervals specified by a count value set in advance.

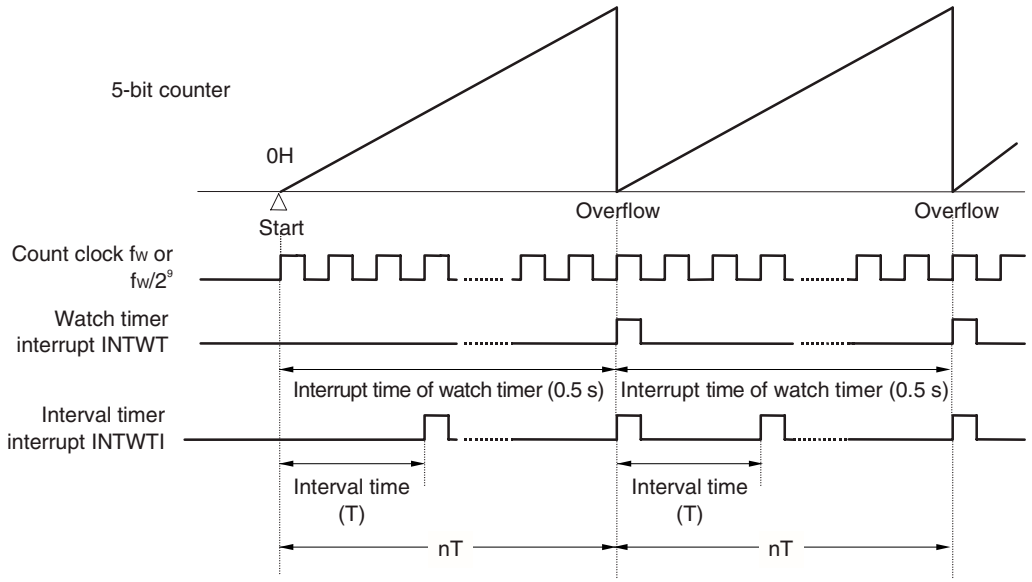
The interval time can be selected by bits 4 through 6 (WTM4 through WTM6) of the watch timer mode control register (WTM).

Table 8-3. Interval Time of Interval Timer

WTM6	WTM5	WTM4	Interval Time	f_W = 32.768 kHz
0	0	0	2^4 1/f_W	488 s
0	0	1	2^5 1/f_W	977 s
0	1	0	2^6 1/f_W	1.95 ms
0	1	1	2^7 1/f_W	3.91 ms
1	0	0	2^8 1/f_W	7.81 ms
1	0	1	2^9 1/f_W	15.6 ms
Other than above			Setting prohibited	

Remark f_W: Watch timer clock frequency

Figure 8-2. Operation Timing of Watch Timer/Interval Timer



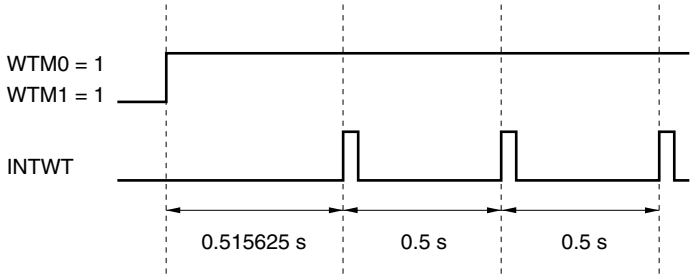
Remark fw: Watch timer clock frequency
( ): fw = 32.768 kHz
n: Interval timer operation count

8.4.3 Cautions

Some time is required before the first watch timer interrupt request (INTWT) is generated after operation is enabled (WTM1 and WTM0 bits of watch timer mode control register (WTM) = 1, 1).

Figure 8-3. Example of Generation of Watch Timer Interrupt Request (INTWT) (When Interrupt Period = 0.5 s)

It takes 0.515625 seconds for the first INTWT to be generated (2^9 1/32.768 = 0.015625 s). INTWT is then generated every 0.5 seconds.



CHAPTER 9 WATCHDOG TIMER

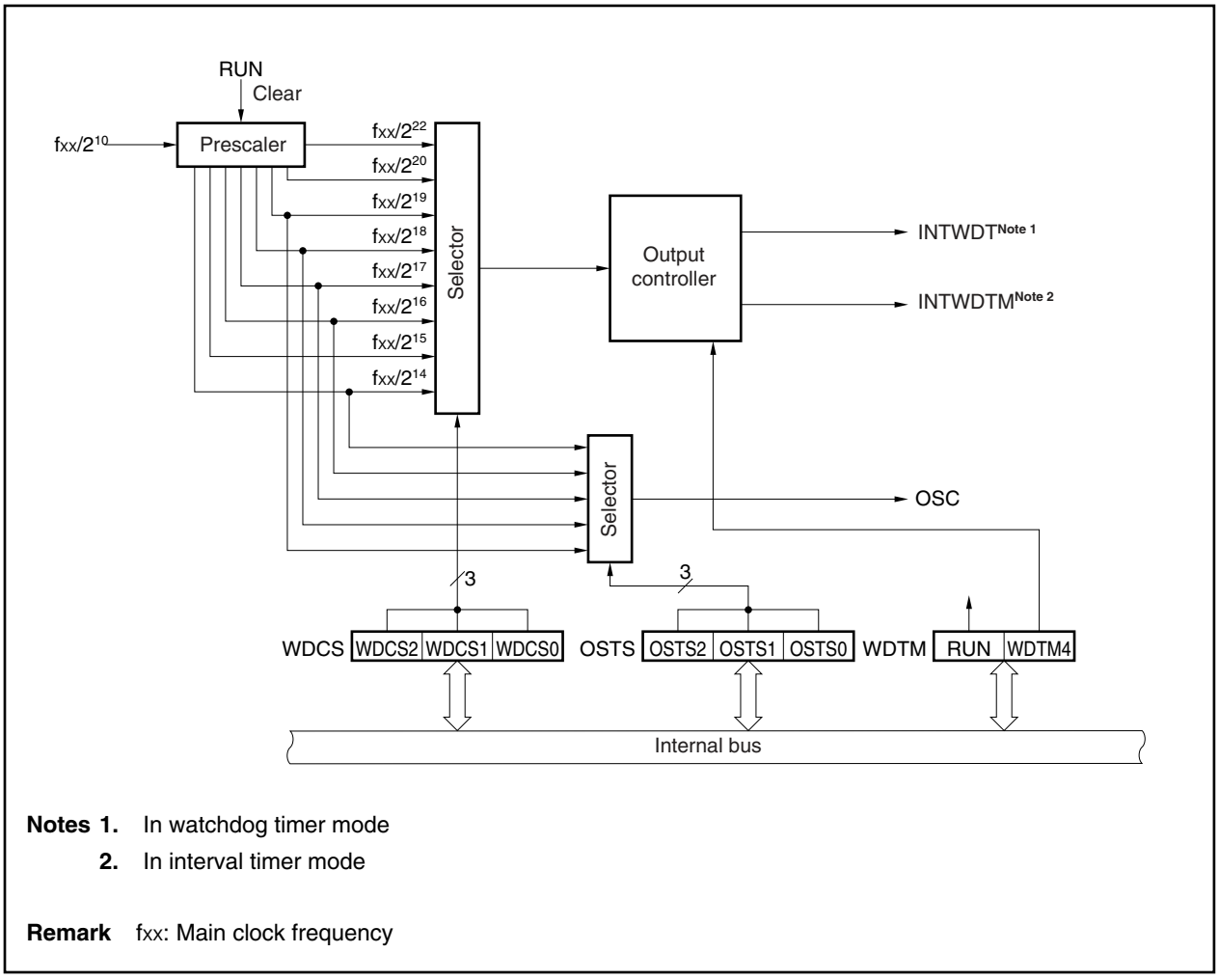
9.1 Functions

The watchdog timer has the following functions. Figure 9-1 shows a block diagram of the watchdog timer.

- Watchdog timer
- Interval timer
- Selecting the oscillation stabilization time

Caution Use the watchdog timer mode register (WDTM) to select the watchdog timer mode or the interval timer mode.

Figure 9-1. Block Diagram of Watchdog Timer



Notes 1. In watchdog timer mode

2. In interval timer mode

Remark fxx: Main clock frequency

(1) Watchdog timer mode

This mode detects a program loop. When a loop is detected, a non-maskable interrupt can be generated.

Table 9-1. Loop Detection Time of Watchdog Timer

Clock	Loop Detection Time			
	fxx = 20 MHz	fxx = 17 MHz	fxx = 10 MHz	fxx = 2 MHz
2^{14}/fxx	819.2 s	964 s	1.6 ms	8.2 ms
2^{15}/fxx	1.6 ms	1.928 ms	3.2 ms	16.4 ms
2^{16}/fxx	3.3 ms	3.855 ms	6.6 ms	32.8 ms
2^{17}/fxx	6.6 ms	7.710 ms	13.1 ms	65.5 ms
2^{18}/fxx	13.1 ms	15.42 ms	26.2 ms	131.1 ms
2^{19}/fxx	26.2 ms	30.84 ms	52.4 ms	262.1 ms
2^{20}/fxx	52.4 ms	61.68 ms	104.9 ms	524.3 ms
2^{22}/fxx	209.7 ms	246.7 ms	419.4 ms	2.1 s

(2) Interval timer mode

Interrupts are generated at a preset time interval.

Table 9-2. Interval Time of Interval Timer

Clock	Interval Time			
	fxx = 20 MHz	fxx = 17 MHz	fxx = 10 MHz	fxx = 2 MHz
2^{14}/fxx	819.2 s	964 s	1.6 ms	8.2 ms
2^{15}/fxx	1.6 ms	1.928 ms	3.2 ms	16.4 ms
2^{16}/fxx	3.3 ms	3.855 ms	6.6 ms	32.8 ms
2^{17}/fxx	6.6 ms	7.710 ms	13.1 ms	65.5 ms
2^{18}/fxx	13.1 ms	15.42 ms	26.2 ms	131.1 ms
2^{19}/fxx	26.2 ms	30.84 ms	52.4 ms	262.1 ms
2^{20}/fxx	52.4 ms	61.68 ms	104.9 ms	524.3 ms
2^{22}/fxx	209.7 ms	246.7 ms	419.4 ms	2.1 s

9.2 Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Watchdog Timer Configuration

Item	Configuration
Control registers	Oscillation stabilization time select register (OSTS) Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM)

9.3 Watchdog Timer Control Register

Three registers control the watchdog timer.

- Oscillation stabilization time select register (OSTS)
- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Oscillation stabilization time select register (OSTS)

This register selects the oscillation stabilization time after a reset is applied or the software STOP mode is released until the oscillation is stable.

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H.

After reset: 04H R/W Address: FFFFF380H

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

OSTS2	OSTS1	OSTS0	Oscillation Stabilization Time Selection				
			Clock	fxx			
				20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	$2^{14}/f_{XX}$	819 s	964 s	1.6 ms	8.2 ms
0	0	1	$2^{16}/f_{XX}$	3.3 ms	3.855 ms	6.6 ms	32.8 ms
0	1	0	$2^{17}/f_{XX}$	6.6 ms	7.710 ms	13.1 ms	65.5 ms
0	1	1	$2^{18}/f_{XX}$	13.1 ms	15.42 ms	26.2 ms	131.1 ms
1	0	0	$2^{19}/f_{XX}$ (after reset)	26.2 ms	30.84 ms	52.4 ms	262.1 ms
Other than above			Setting prohibited				

(2) Watchdog timer clock select register (WDCS)

This register selects the overflow time of the watchdog timer and the interval timer.

WDCS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets WDCS to 00H.

After reset: 00H R/W Address: FFFFF382H

	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Watchdog Timer/Interval Timer Overflow Time				
			Clock	fxx			
				20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	2^{14}/fxx	819.2 s	964 s	1.6 ms	8.2 ms
0	0	1	2^{15}/fxx	1.6 ms	1.928 ms	3.2 ms	16.4 ms
0	1	0	2^{16}/fxx	3.3 ms	3.855 ms	6.6 ms	32.8 ms
0	1	1	2^{17}/fxx	6.6 ms	7.710 ms	13.1 ms	65.5 ms
1	0	0	2^{18}/fxx	13.1 ms	15.42 ms	26.2 ms	131.1 ms
1	0	1	2^{19}/fxx	26.2 ms	30.84 ms	52.4 ms	262.1 ms
1	1	0	2^{20}/fxx	52.4 ms	61.68 ms	104.9 ms	524.3 ms
1	1	1	2^{22}/fxx	209.7 ms	246.7 ms	419.4 ms	2.1 s

Caution Be sure to set bits 3 to 7 to 0.

(3) Watchdog timer mode register (WDTM)

This register sets the operating mode of the watchdog timer, and enables and disables counting.

WDTM is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets WDTM to 00H.

After reset: 00H R/W Address: FFFFF384H

	7	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	0	0	0	0

RUN	Operating Mode Selection for Watchdog Timer[Note 1]
0	Disable count
1	Clear count and start counting

WDTM4	Operating Mode Selection for Watchdog Timer[Note 2]
0	Interval timer mode (If an overflow occurs, the maskable interrupt INTWDTM is generated.)
1	Watchdog timer mode 1 (If an overflow occurs, the non-maskable interrupt INTWDT is generated.)

Notes 1. Once RUN is set (1), the register cannot be cleared (0) by software. Therefore, when counting starts, counting cannot be stopped except by $\overline{\text{RESET}}$ input.

2. Once WDTM4 is set (1), the register cannot be cleared (0) by software.

Caution If RUN is set (1) and the watchdog timer is cleared, the actual overflow time can be up to $2^{10}/f_{XX}$ [seconds] shorter than the set time.

9.4 Operation

9.4.1 Operating as watchdog timer

Set bit 4 (WDTM4) of the watchdog timer mode register (WDTM) to 1 to operate as a watchdog timer to detect a program loop.

Setting bit 7 (RUN) of WDTM to 1 starts the count operation. After counting starts, if RUN is set to 1 again within the set time interval for loop detection, the watchdog timer is cleared and counting starts again.

If RUN is not set to 1 and the loop detection time has elapsed, a non-maskable interrupt (INTWDT) is generated (no reset function).

The watchdog timer stops running in the IDLE mode and software STOP mode. Consequently, set RUN to 1 and clear the watchdog timer before entering the IDLE mode or software STOP mode. Do not set the watchdog timer when operating in the HALT mode since the watchdog timer runs in HALT mode.

Cautions 1. The actual loop detection time can be up to $2^{10}/f_{XX}$ [seconds] shorter than the set time.

2. When the subclock is selected as the CPU clock, the watchdog timer stops (suspends) counting.

Table 9-4. Loop Detection Time of Watchdog Timer

Clock	Loop Detection Time			
	fxx = 20 MHz	fxx = 17 MHz	fxx = 10 MHz	fxx = 2 MHz
$2^{14}/f_{XX}$	819.2 s	964 s	1.6 ms	8.2 ms
$2^{15}/f_{XX}$	1.6 ms	1.928 ms	3.2 ms	16.4 ms
$2^{16}/f_{XX}$	3.3 ms	3.855 ms	6.6 ms	32.8 ms
$2^{17}/f_{XX}$	6.6 ms	7.710 ms	13.1 ms	65.5 ms
$2^{18}/f_{XX}$	13.1 ms	15.42 ms	26.2 ms	131.1 ms
$2^{19}/f_{XX}$	26.2 ms	30.84 ms	52.4 ms	262.1 ms
$2^{20}/f_{XX}$	52.4 ms	61.68 ms	104.9 ms	524.3 ms
$2^{22}/f_{XX}$	209.7 ms	246.7 ms	419.4 ms	2.1 s

9.4.2 Operating as interval timer

Set bit 4 (WDTM4) to 0 in the watchdog timer mode register (WDTM) to operate the watchdog timer as an interval timer that repeatedly generates interrupts with a preset count value as the interval.

When operating as an interval timer, the interrupt mask flag (WDTMK) of the WDTIC register and the priority setting flag (WDTPR0 to WDTPR2) become valid, and a maskable interrupt (INTWDTM) can be generated. The default priority of INTWDTM has the highest priority setting of the maskable interrupts.

The interval timer continues operating in the HALT mode and stops in the IDLE mode and software STOP mode. Therefore, before entering the IDLE mode/software STOP mode, set the RUN bit of WDTM register to 1 and clear the interval timer. Then set the IDLE mode/software STOP mode.

Cautions 1. Once bit 4 (WDTM4) of WDTM is set to 1 (selecting the watchdog timer mode), the interval timer mode is not entered as long as $\overline{\text{RESET}}$ is not input.

2. The interval time immediately after setting in WDTM can be up to $2^{10}/f_{XX}$ [seconds] shorter than the set time.

3. When the subclock is selected as the CPU clock, the watchdog timer stops (suspends) counting.

Table 9-5. Interval Time of Interval Timer

Clock	Interval Time			
	fxx = 20 MHz	fxx = 17 MHz	fxx = 10 MHz	fxx = 2 MHz
$2^{14}/f_{XX}$	819.2 s	964 s	1.6 ms	8.2 ms
$2^{15}/f_{XX}$	1.6 ms	1.928 ms	3.2 ms	16.4 ms
$2^{16}/f_{XX}$	3.3 ms	3.855 ms	6.6 ms	32.8 ms
$2^{17}/f_{XX}$	6.6 ms	7.710 ms	13.1 ms	65.5 ms
$2^{18}/f_{XX}$	13.1 ms	15.42 ms	26.2 ms	131.1 ms
$2^{19}/f_{XX}$	26.2 ms	30.84 ms	52.4 ms	262.1 ms
$2^{20}/f_{XX}$	52.4 ms	61.68 ms	104.9 ms	524.3 ms
$2^{22}/f_{XX}$	209.7 ms	246.7 ms	419.4 ms	2.1 s

9.5 Standby Function Control Register

(1) Oscillation stabilization time select register (OSTS)

The wait time from releasing the software STOP mode until the oscillation stabilizes is controlled by the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

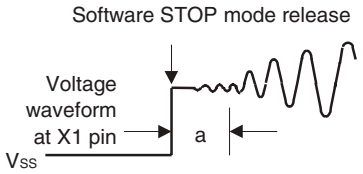
$\overline{\text{RESET}}$ input sets OSTS to 04H.

After reset: 04H R/W Address: FFFFF380H

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

OSTS2	OSTS1	OSTS0	Oscillation Stabilization Time Selection				
			Clock	fxx			
				20 MHz	17 MHz	10 MHz	2 MHz
0	0	0	2^{14}/fxx	819.2 s	964 s	1.6 ms	8.2 ms
0	0	1	2^{16}/fxx	3.3 ms	3.855 ms	6.6 ms	32.8 ms
0	1	0	2^{17}/fxx	6.6 ms	7.710 ms	13.1 ms	65.5 ms
0	1	1	2^{18}/fxx	13.1 ms	15.42 ms	26.2 ms	131.1 ms
1	0	0	2^{19}/fxx (after reset)	26.2 ms	30.84 ms	52.4 ms	262.1 ms
Other than above			Setting prohibited				

Caution The wait time at the release of the software STOP mode does not include the time ("a" in the figure below) until clock oscillation starts after releasing the software STOP mode when $\overline{\text{RESET}}$ is input or an interrupt is generated.



CHAPTER 10 SERIAL INTERFACE FUNCTION

10.1 Overview

The V850/SA1 supports the following on-chip serial interfaces.

- Channel 0: 3-wire serial I/O (CSI0)/I^2C bus interface (I^2C)[Note]
- Channel 1: 3-wire serial I/O (CSI1)/Asynchronous serial interface (UART0)
- Channel 2: 3-wire serial I/O (CSI2)
- Channel 3: Asynchronous serial interface (UART1)

★ **Note** I^2C supports multi-masters (PD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only).
Either 3-wire serial I/O or I^2C can be used as a serial interface.

10.2 3-Wire Serial I/O (CSI0 to CSI2)

CSIn (n = 0 to 2) has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed.

(2) 3-wire serial I/O mode (fixed to MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{SCKn}$), serial output line (SOn), and serial input line (SIn).

Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfer is reduced.

The first bit in the 8-bit data in serial transfers is fixed to the MSB.

The $\overline{SCKn}$ and SOn pins are set to normal output or N-ch open-drain output by setting the port 1 function register (PF1) and port 2 function register (PF2).

3-wire serial I/O mode is useful for connection to a peripheral I/O device that includes a clocked serial interface, a display controller, etc.

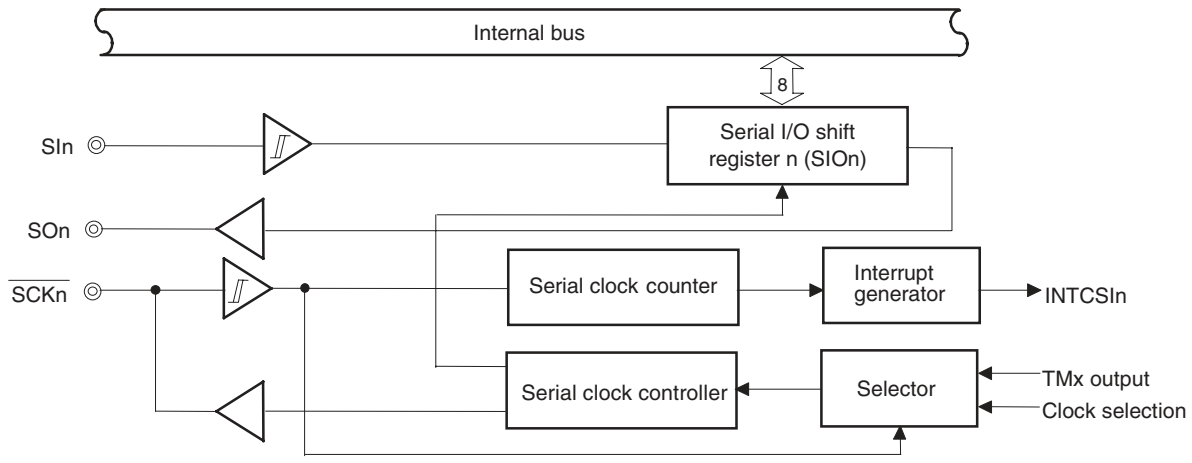
10.2.1 Configuration

CSIn consists of the following hardware.

Table 10-1. Configuration of CSIn

Item	Configuration
Registers	Serial I/O shift registers 0 to 2 (SIO0 to SIO2)
Control registers	Serial clock select registers 0 to 2 (CSIS0 to CSIS2)
	Serial operation mode registers 0 to 2 (CSIM0 to CSIM2)

Figure 10-1. Block Diagram of 3-Wire Serial I/O



Remark n = 0 to 2

Remarks 1. n = 0 to 2

2. TMx output is as follows:

When n = 0: TM2

When n = 1, 2: TM3

(1) Serial I/O shift registers 0 to 2 (SIO0 to SIO2)

SIOn is an 8-bit register that performs parallel-serial conversion and serial transmission/reception (shift operations) synchronized with the serial clock.

SIOn is set by an 8-bit memory manipulation instruction.

When "1" is set to bit 7 (CSIEn) of serial operation mode register n (CSIMn), a serial operation can be started by writing data to or reading data from SIOn.

When transmitting, data written to SIOn is output via the serial output (SOn).

When receiving, data is read from the serial input (SIn) and written to SIOn.

$\overline{\text{RESET}}$ input resets these registers to 00H.

Caution Do not execute SIOn accesses except for accesses that become the transfer start trigger during a transfer operation (read is disabled when MODEn = 0 and write is disabled when MODEn = 1).

10.2.2 CSIn control registers

CSIn uses the following registers for control functions.

- Serial clock select register n (CSISn)
- Serial operation mode register n (CSIMn)

(1) Serial clock select registers 0 to 2 (CSIS0 to CSIS2), serial operation mode registers 0 to 2 (CSIM0 to CSIM2)

The CSISn register is used to set of the serial clock serial interface channel n.

The CSISn register can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSISn register the to 00H.

The CSIMn register is used to enable or disable serial interface channel n's serial clock, operation modes, and specific operations.

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

$\overline{\text{RESET}}$ input sets the CSIMn register to 00H.

After reset: 00H R/W Address: CSIS0 FFFFF2A4H
CSIS1 FFFFF2B4H
CSIS2 FFFFF2C4H

	7	6	5	4	3	2	1	0
CSISn	0	0	0	0	0	0	0	SCLn2

After reset: 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 2)

CSIEn	SIOn Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disable	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note]

MODEn	Transfer Operation Mode Flag		
	Operation Mode	Transfer Start Trigger	SOn Output
0	Transmit/receive mode	SIOn write	Normal output
1	Receive-only mode	SIOn read	Port function

SCLn2	SCLn1	SCLn0	Clock Selection
0	0	0	External clock input ($\overline{SCKn}$)
0	0	1	at n = 0: TM2 output at n = 1, 2: TM3 output
0	1	0	fxx/8 (2.5 MHz)
0	1	1	fxx/16 (1.25 MHz)
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	fxx/32 (625 kHz)
1	1	1	fxx/64 (312.5 kHz)

Notes 1. When CSIEn = 0 (SIOn operation stop status), the SIn, SOn, and $\overline{SCKn}$ pins can be used as port functions.

2. When CSIEn = 1 (SIOn operation enable status), the SIn pin when only using the transmit function and the SOn pin in receive-only mode can be used as port functions.

Cautions 1. Do not perform bit manipulation of SCLn1 and SCLn0.

2. Be sure to set bits 3 to 6 of the CSIMn register to 0.

Remarks 1. Values in parentheses apply when fxx = 20 MHz.

2. The selected clock is output from the timer, it is not necessary to set the P26/TI2/TO2 and P27/T31/TO3 pins to timer output mode.

10.2.3 Operations

CSIn has the following two operation modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

Serial transfers are not performed in this mode, enabling a reduction in power consumption.

In operation stop mode, if the SIn, SOn, and $\overline{\text{SCKn}}$ pins are also used as I/O ports, they can be used as normal I/O ports as well.

(a) Register settings

Operation stop mode is set via the CSIEn bit of serial operation mode register n (CSIMn).

Figure 10-2. Settings of CSIMn (Operation Stop Mode)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 2)

CSIEn	SIOn Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disable	Clear	Port function

(2) 3-wire serial I/O mode

3-wire serial I/O mode is useful when connecting to a peripheral I/O device that includes a clocked serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCKn}}$), serial output line (SOn), and serial input line (SIn).

(a) Register settings

3-wire serial I/O mode is set via serial operation mode register n (CSIMn).

Figure 10-3. Settings of CSIMn (3-Wire Serial I/O Mode)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 2)

CSIEn	SIOn Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
1	Operation enable	Count operation enable	Serial function + port function

MODEn	Transfer Operation Mode Flag		
	Operation Mode	Transfer Start Trigger	SOn Output
0	Transmit/receive mode	SIOn write	Normal output
1	Receive-only mode	SIOn read	Port function

SCLn2	SCLn1	SCLn0	Clock Selection
0	0	0	External clock input ($\overline{\text{SCKn}}$)
0	0	1	When n = 0: TM2 output When n = 1, 2: TM3 output
0	1	0	fxx/8 (2.5 MHz)
0	1	1	fxx/16 (1.25 MHz)
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	fxx/32 (625 kHz)
1	1	1	fxx/64 (312.5 kHz)

Remarks 1. Values in parentheses apply when f_{XX} = 20 MHz.

2. Refer to **10.2.2 (1) Serial clock select registers 0 to 2 (CSIS0 to CSIS2), serial operation mode registers 0 to 2 (CSIM0 to CSIM2)** for the SCLn2 bit.

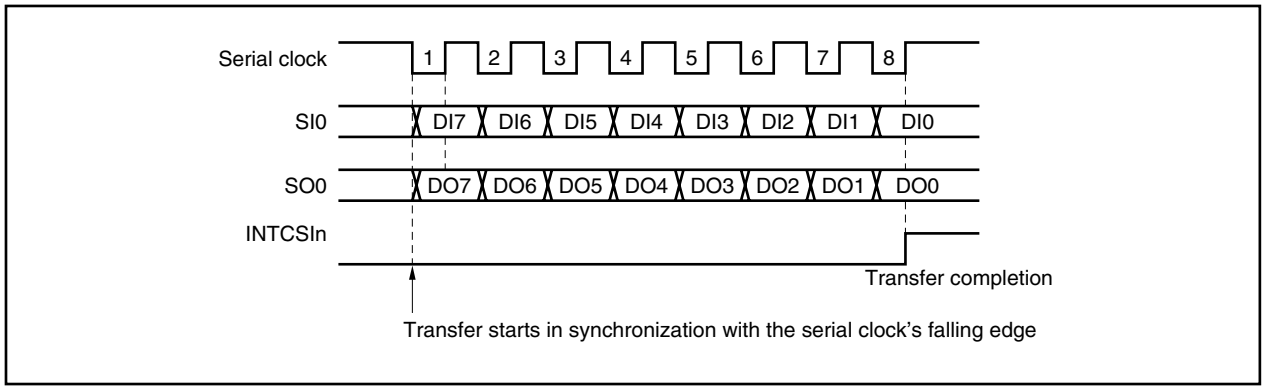
(b) Communication operations

In 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is transmitted or received in synchronization with the serial clock.

Serial I/O shift register n (SIOn) is shifted in synchronization with the falling edge of the serial clock. Transmission data is held in the SOn latch and is output from the SOn pin. Data that is received via the SIn pin in synchronization with the rising edge of the serial clock is latched to SIOn.

Completion of an 8-bit transfer automatically stops operation of SIOn and sets the interrupt request flag (INTCSIn).

Figure 10-4. Timing of 3-Wire Serial I/O Mode



(c) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set to serial I/O shift register n (SIOn).

- The SIOn operation control bit (CSIEn) = 1
- After an 8-bit serial transfer, the internal serial clock is either stopped or is set to high level.

The transfer data is set to SIOn as follows.

- Transmit/receive mode
  When CSIEn = 1 and MODEn = 0, transfer starts when writing to SIOn.
- Receive-only mode
  When CSIEn = 1 and MODEn = 1, transfer starts when reading from SIOn.

Caution After data has been written to SIOn, transfer will not start even if the CSIEn bit value is set to "1".

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets the interrupt request flag (INTCSIn).

10.3 I²C Bus Interface (I²C)

To use the I²C bus function, set the P10/SDA and P12/SCL pins to N-ch open-drain output.

The products that incorporate I²C are shown below.

★ μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, 70F3017AY

I²C has the following two modes.

- Operation stop mode
- I²C (Inter IC) bus mode (multi-masters supported)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

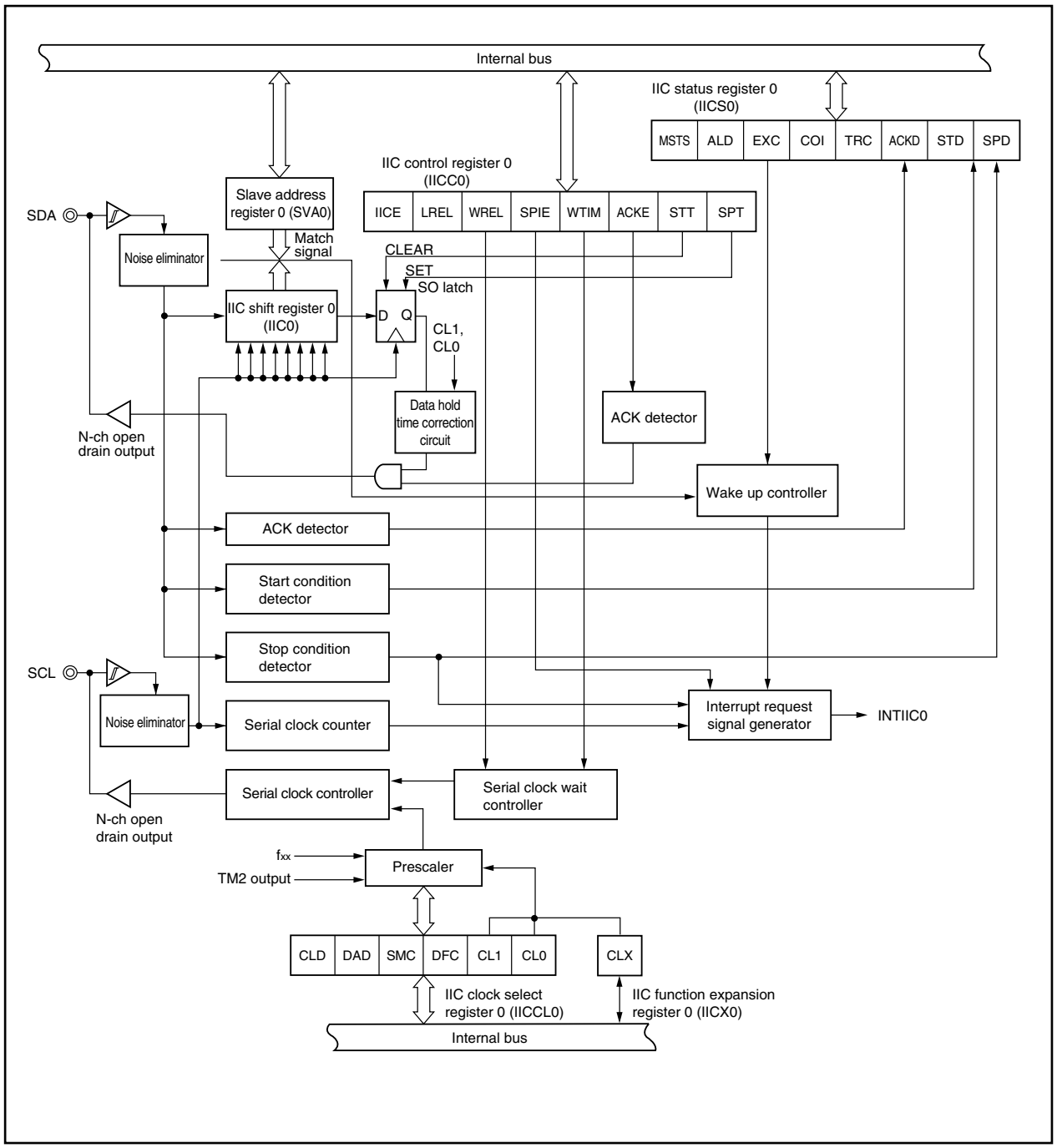
(2) I²C bus mode (multi-masters supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL) line and a serial data bus (SDA) line.

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 an application program that controls the I²C bus.

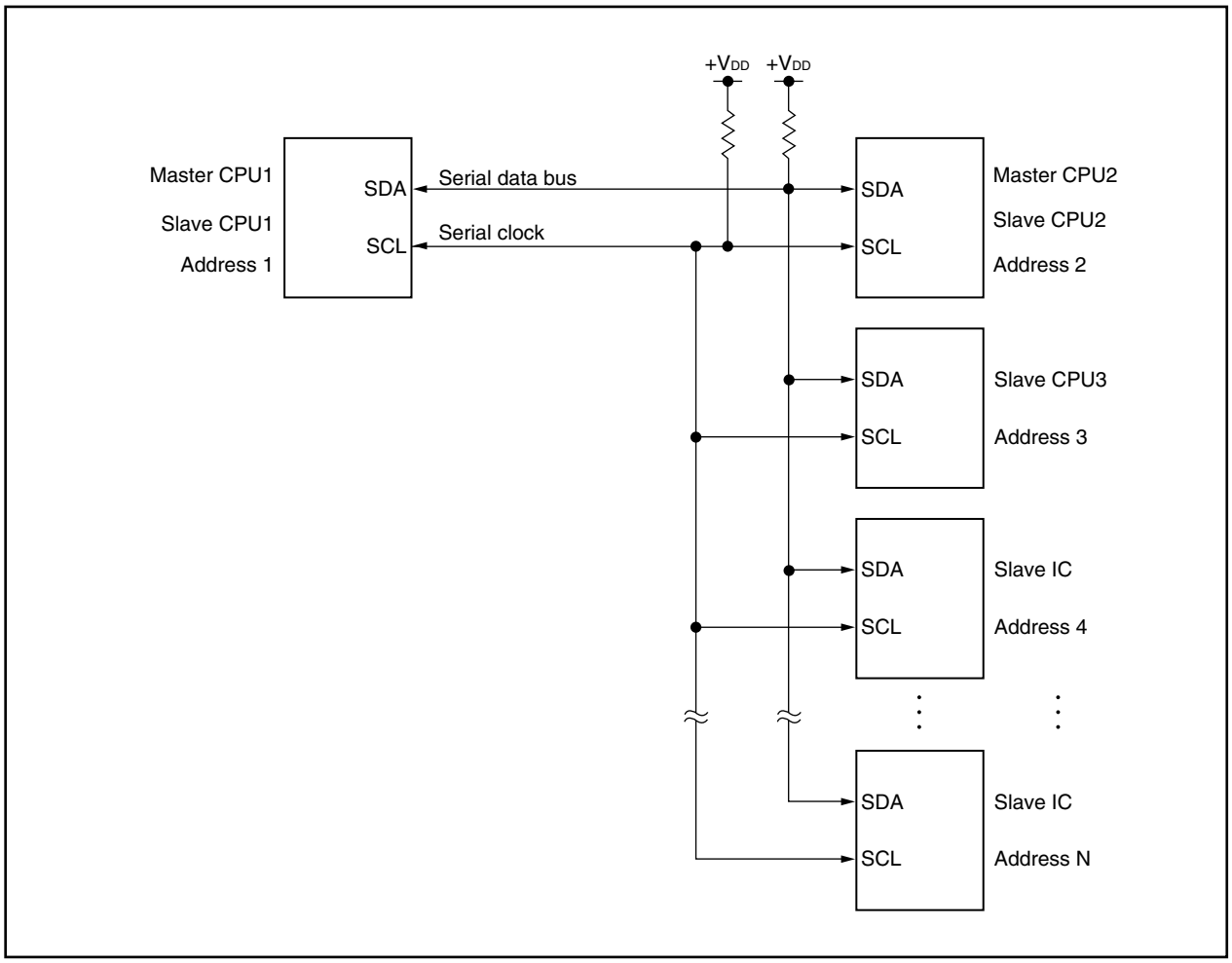
Since SCL and SDA are open-drain outputs, the I²C requires pull-up resistors for the serial clock line and the serial data bus line.

Figure 10-5. Block Diagram of I²C



The following shows a serial bus configuration example.

Figure 10-6. Serial Bus Configuration Example Using I^2C Bus



10.3.1 Configuration

I^2C consists of the following hardware.

Table 10-2. Configuration of I^2C

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC clock select register 0 (IICCL0) IIC function expansion register 0 (IICX0)

(1) IIC shift register 0 (IIC0)

IIC0 is used to convert 8-bit serial data into 8-bit parallel data and vice versa. IIC0 can be used for both transmission and reception.

Write and read operations to IIC0 are used to control the actual transmit and receive operations.

IIC0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IIC0 to 00H.

(2) Slave address register 0 (SVA0)

SVA0 sets local addresses when in slave mode.

SVA0 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets SVA0 to 00H.

(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 0 (SVA0) 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 transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signal (INTIIC0).

An I^2C 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 0 (IICC0)
SPIE bit: Bit 4 of IIC control register 0 (IICC0)

(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.

10.3.2 I^2C control registers

I^2C is controlled by four types of registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC function expansion register 0 (IICX0)
- IIC clock select register 0 (IICCL0)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

(1) IIC control register 0 (IICC0)

IICC0 is used to enable/disable I^2C operations, set wait timing, and set other I^2C operations.

IICC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IICC0 to 00H.

Caution In I^2C bus mode, set the port 1 mode register (PM1) as follows. In addition, set each output latch to 0.

- **Set P10 (SDA) to output mode (PM10 = 0)**
- **Set P12 (SCL) to output mode (PM12 = 0)**

(1/4)

After reset: 00H R/W Address: FFFFF340H

	7	6	5	4	3	2	1	0
IICC0	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I^2C Operation Enable/Disable Specification
0	Operation Stopped. IIC status register 0 (IICS0) preset. Internal operation stopped.
1	Operation enabled.

Condition for clearing (IICE = 0)	Condition for setting (IICE = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

LREL	Exit from Communications
0	Normal operation
1	This exits from the current communications 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 communications entry conditions are met.

- After a stop condition is detected, restart is in master mode.
- An address match or extension code reception occurs after the start condition.

Condition for clearing (LREL = 0)[Note]	Condition for setting (LREL = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICE = 0.

Remark STD: Bit 1 of IIC status register 0 (IICS0)
ACKD: Bit 2 of IIC status register 0 (IICS0)
TRC: Bit 3 of IIC status register 0 (IICS0)
COI: Bit 4 of IIC status register 0 (IICS0)
EXC: Bit 5 of IIC status register 0 (IICS0)
MSTS: Bit 7 of IIC status register 0 (IICS0)

(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 • When $\overline{\text{RESET}}$ is input	• 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 • When $\overline{\text{RESET}}$ is input	• Set by instruction

WTIM	Control of Wait and Interrupt Request Generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for the master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for the master device.

This bit's 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 • When $\overline{\text{RESET}}$ is input	• Set by instruction

ACKE	Acknowledge Control
0	Acknowledgement disabled.
1	Acknowledge enabled. During the ninth clock period, the SDA line is set to low level. However, the $\overline{\text{ACK}}$ 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 • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICE = 0.

(3/4)

STT	Start Condition Trigger
0	Start conditions not generated.
1	When bus is released (in STOP mode): Generate 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): Generate 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: Note that a start condition cannot be generated normally during the ACK period.
- Cannot be set at the same time as SPTn

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 • Cleared by LREL = 1 • When IICE = 0 • Cleared when $\overline{\text{RESET}}$ is input	• Set by instruction

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 device's 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 level 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 during the wait period when ACKE has been set to 0 and slave has been notified of final reception.
- For master transmission: Note that a stop condition cannot be generated normally during the ACK period. Set a stop condition 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 • Cleared by LREL = 1 • When IICE = 0 • Cleared when $\overline{\text{RESET}}$ is input	• 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 operation enable status. For details, see **10.3.13 Cautions**.

Caution When bit 3 (TRC) of IIC status register 0 (IICS0) 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 0 (IICS0)

IICS0 indicates the status of the I^2C bus.

IICS0 can be set by a 1-bit or 8-bit memory manipulation instruction. IICS0 is a read-only register.

$\overline{RESET}$ input sets IICS0 to 00H.

(1/3)

After reset: 00H R Address: FFFFF342H

	7	6	5	4	3	2	1	0
IICS0	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 • When $\overline{RESET}$ is input	• 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 • When $\overline{RESET}$ is input	• When the arbitration result is a “loss”.

Note This register is also cleared when a bit manipulation instruction is executed for bits other than IICS0.

Remark LREL: Bit 6 of IIC control register 0 (IICC0)
IICE: Bit 7 of IIC control register 0 (IICC0)

(2/3)

EXC	Detection of Extension Code Reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the higher four bits of the received address data are either “0000” or “1111” (set at the rising edge of the eighth clock).

COI	Detection of Matching Addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVA0) (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 rising 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 $\overline{\text{ALD}}$ changes from 0 to 1 • When $\overline{\text{RESET}}$ is input 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 When bit 3 (TRC) of IIC status register 0 (IICS0) is 1, if a wait is released by setting bit 5 (WREL) of IIC control register 0 (IICC0) at the 9th clock, the SDA line becomes high impedance after TRC is cleared.

Remark LREL: Bit 6 of IIC control register 0 (IICC0)
IICE: Bit 7 of IIC control register 0 (IICC0)

(3/3)

ACKD	Detection of $\overline{ACK}$
0	$\overline{ACK}$ was not detected.
1	$\overline{ACK}$ 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 • When $\overline{RESET}$ is input	• 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 • When $\overline{RESET}$ is input	• 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 • When $\overline{RESET}$ is input	• When a stop condition is detected

Remark LREL: Bit 6 of IIC control register 0 (IICC0)

IICE: Bit 7 of IIC control register 0 (IICC0)

(3) IIC clock select register 0 (IICCL0), IIC function expansion register 0 (IICX0)

The IICCL0 and IICX0 registers are used to set the transfer clock for the I^2C bus.

The IICCL0 and IICX0 registers can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF34AH

	7	6	5	4	3	2	1	0
IICX0	0	0	0	0	0	0	0	CLX

After reset: 00H R/W[Note] Address: FFFFF344H

	7	6	5	4	3	2	1	0
IICCL0	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 • When $\overline{\text{RESET}}$ is input	• When the SCL 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 • When $\overline{\text{RESET}}$ is input	• When the SDA line is at high level

SMC	Operation Mode Switching
0	Operated in standard mode
1	Operated in high-speed mode

DFC	Operation Control of Digital Filter
0	Digital filter off
1	Digital filter on
A digital filter can be used only in high-speed mode. The transfer clock does not change by setting DFC on/off in the high-speed mode.	

Note Bits 4 and 5 of the IICCL0 register are read-only bits.

Caution Be sure to set bits 6 and 7 of the IICCL0 register to 0.

Remark IICE: Bit 7 of IIC control register 0 (IICC0)

(2/2)

★

CLX	SMC	CL1	CL0	Selection Clock	Range of Settable Main Clock Frequency (f_{XX})	Operation Mode
0	0	0	0	f_{XX}/44	2.0 MHz to 4.19 MHz	Standard mode (SMC = 0)
0	0	0	1	f_{XX}/86	4.19 MHz to 8.38 MHz	
0	0	1	0	f_{XX}/172	8.38 MHz to 17 MHz	
0	0	1	1	TM2 output/66	TM2 setting	
0	1	0	0	f_{XX}/24	4.0 MHz to 8.38 MHz	High-speed mode (SMC = 1)
0	1	0	1			
0	1	1	0	f_{XX}/48	8.0 MHz to 17 MHz	
0	1	1	1	TM2 output/18	TM2 setting	
1	1	0	0	f_{XX}/12	4.0 MHz to 4.19 MHz	
1	1	0	1			
Other than above				Setting prohibited		

Remark When the selected clock is the timer output, it is not necessary to set the P26/TI2/TO2 pin to timer output mode.

(5) IIC shift register 0 (IIC0)

IIC0 is used for serial transmission/reception (shift operations) that are synchronized with the serial clock. It can be read from or written to in 8-bit units, but data should not be written to IIC0 during a data transfer.

After reset: 00H R/W Address: FFFFF348H

	7	6	5	4	3	2	1	0
IIC0								

(6) Slave address register 0 (SVA0)

SVA0 holds the I^2C bus's slave addresses.

It can be read from or written to in 8-bit units, but bit 0 should be fixed to 0.

After reset: 00H R/W Address: FFFFF346H

	7	6	5	4	3	2	1	0
SVA0								0

10.3.3 I^2C bus mode functions

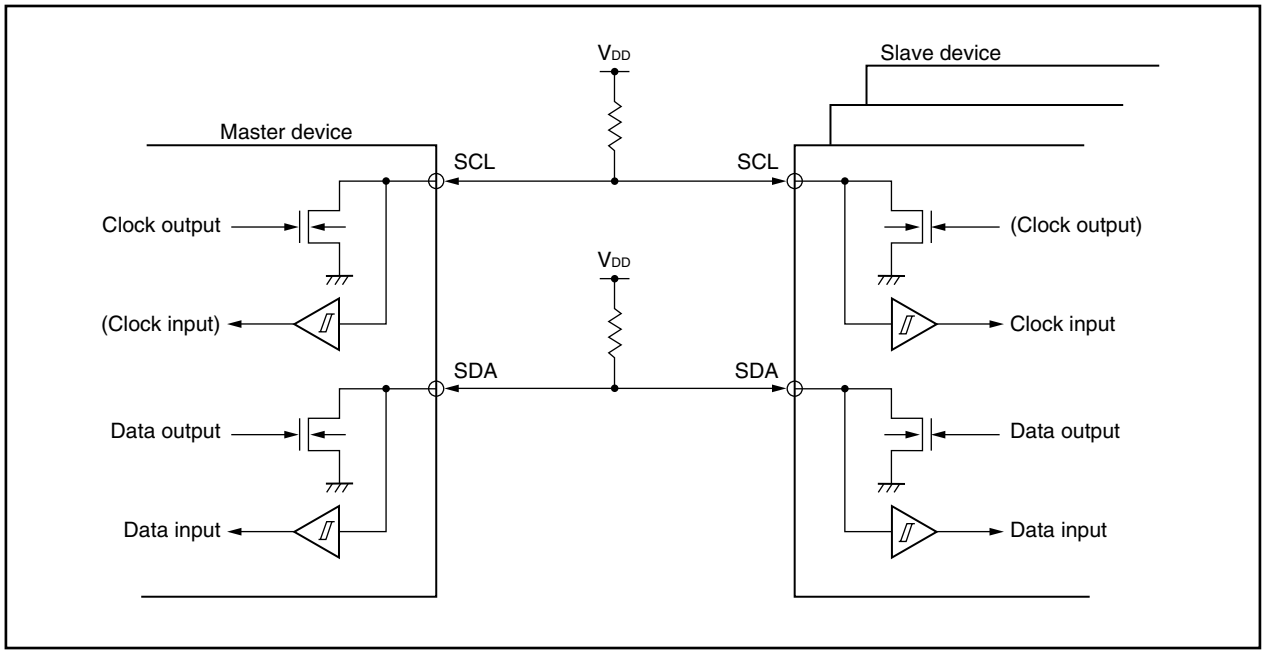
(1) Pin configuration

The serial clock pin (SCL) and serial data bus pin (SDA) are configured as follows.

SCL ..... This pin is used for serial clock input and output.
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 input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

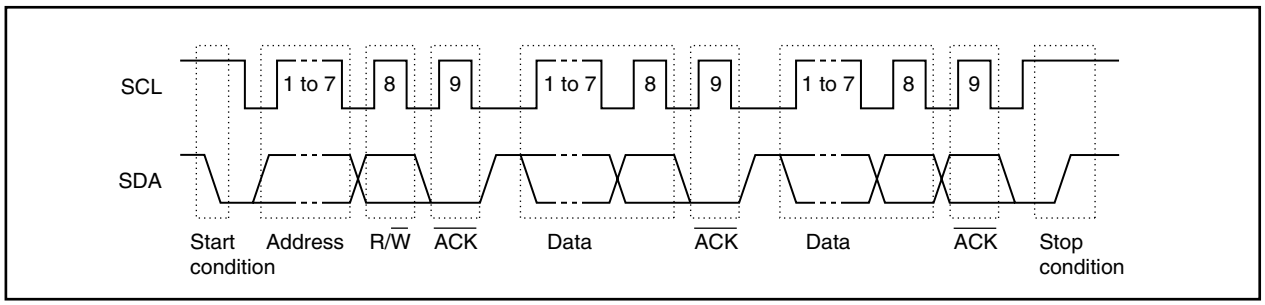
Figure 10-7. Pin Configuration Diagram



10.3.4 I^2C bus definitions and control methods

The following section describes the I^2C bus's serial data communication format and the signals used by the I^2C bus. Figure 10-8 shows the transfer timing for the "start condition", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 10-8. I^2C Bus's Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

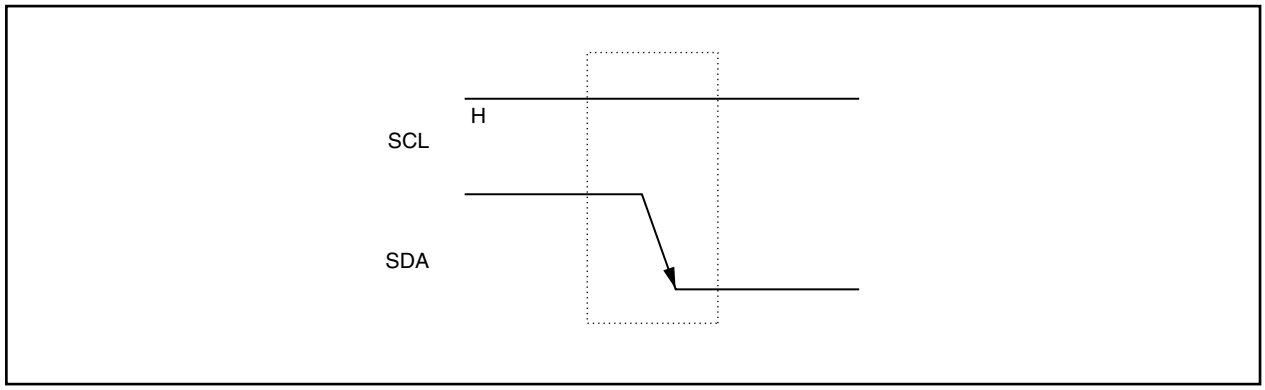
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL) is continuously output by the master device. However, in the slave device, the SCL's low-level period can be extended and a wait can be inserted.

(1) Start condition

The 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 10-9 Start Condition



A start condition is output when bit 1 (STT) of IIC control register 0 (IICC0) is set to 1 after a stop condition has been detected (SPD: Bit 0 = 1 in IIC status register 0 (IICS0)). When a start condition is detected, bit 1 (STD) of IICS0 is set to 1.

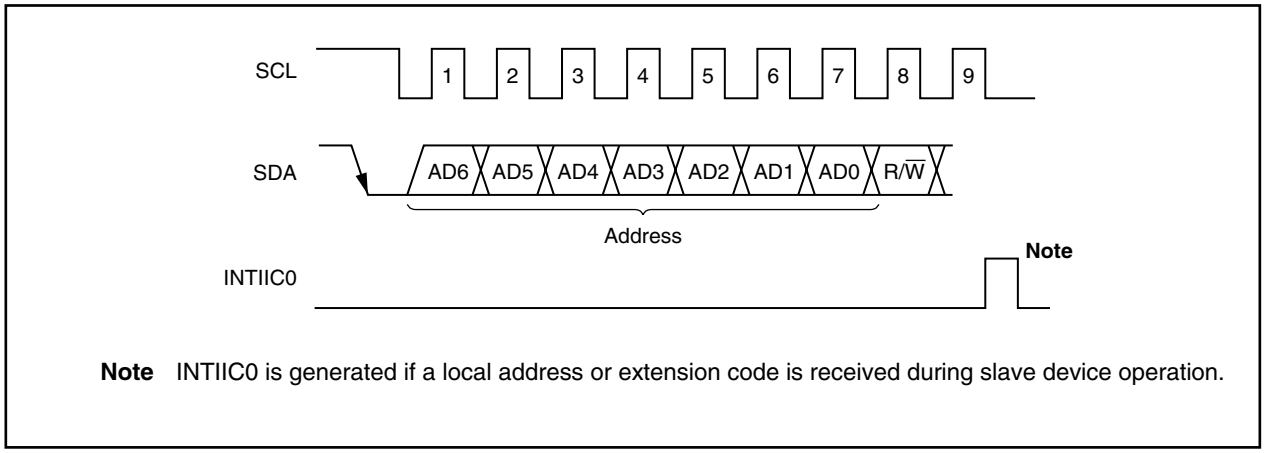
(2) Addresses

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 address data matches the data values stored in slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 10-10. Address



Note INTIIC0 is generated if a local address or extension code is received during slave device operation.

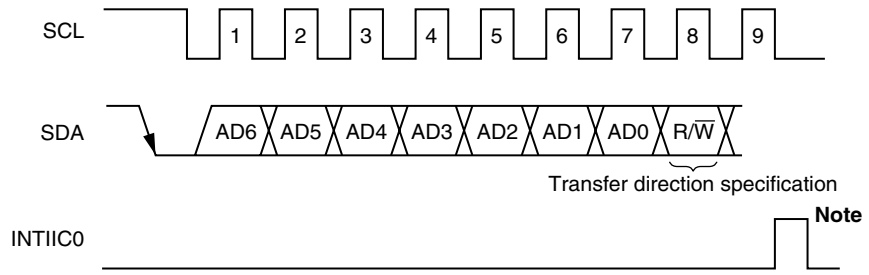
The slave address and the eighth bit, which specifies the transfer direction as described in **(3) Transfer direction specification** below, are written together to IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

The slave address is assigned to the higher 7 bits of IIC0.

(3) Transfer direction specification

In addition to the 7-bit address data, the master device transmits 1-bit data 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 10-11. Transfer Direction Specification



Note INTIIC0 is generated if 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 every 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.

(a) Reception was not performed normally.
(b) 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).

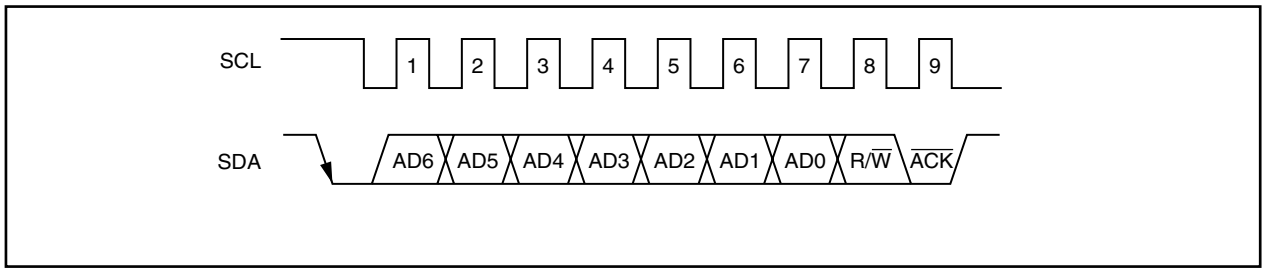
When bit 2 (ACKE) of IIC control register 0 (IICC0) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled.

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRC) of IIC status register 0 (IICS0) to be set. When this 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 10-12. $\overline{\text{ACK}}$ Signal



When the local address is received, an $\overline{ACK}$ signal is automatically output in synchronization with the falling edge of the SCL's eighth clock regardless of the ACKE value. No $\overline{ACK}$ signal is output if the received address is not a local address.

The $\overline{ACK}$ signal output method during data reception is based on the wait timing setting, as described below.

When 8-clock wait is selected: $\overline{ACK}$ signal is output at the falling edge of the SCL's eighth clock when ACKE is set to 1 before wait cancellation.

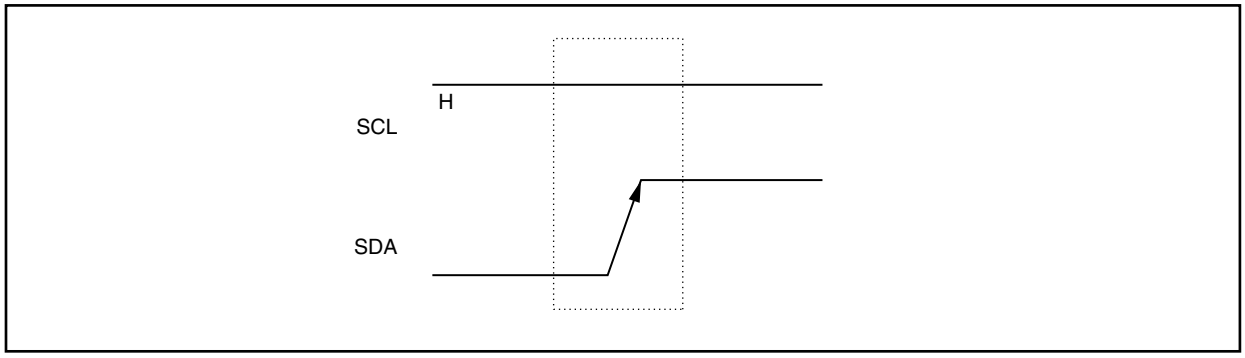
When 9-clock wait is selected: $\overline{ACK}$ signal is automatically output at the falling edge of the SCL's eighth clock if ACKE has already been set to 1.

(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 10-13. Stop Condition



A stop condition is generated when bit 0 (SPT) of IIC control register 0 (IICC0) is set to 1. When the stop condition is detected, bit 0 (SPD0) of IIC status register 0 (IICS0) is set to 1 and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 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 10-14. Wait Signal (1/2)

(a) When master device has a nine-clock wait and slave device has an eight-clock wait (master: transmission, slave: reception, and ACKE = 1)

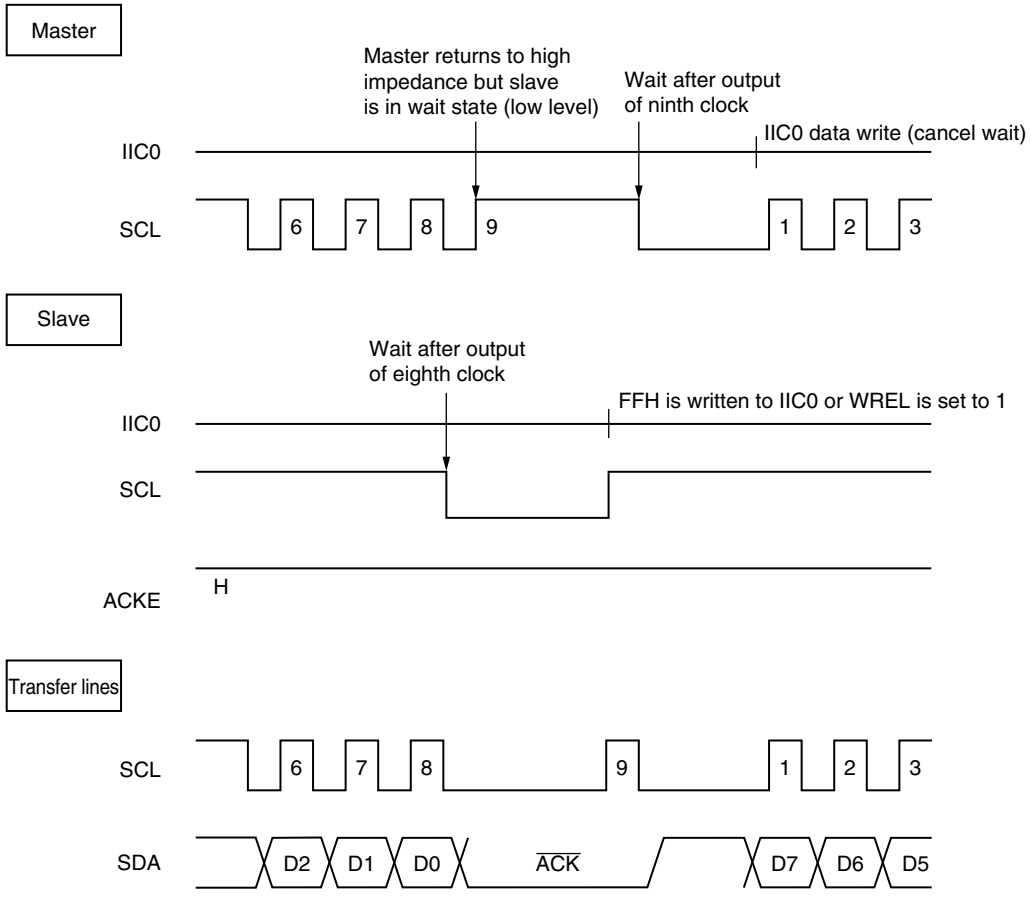
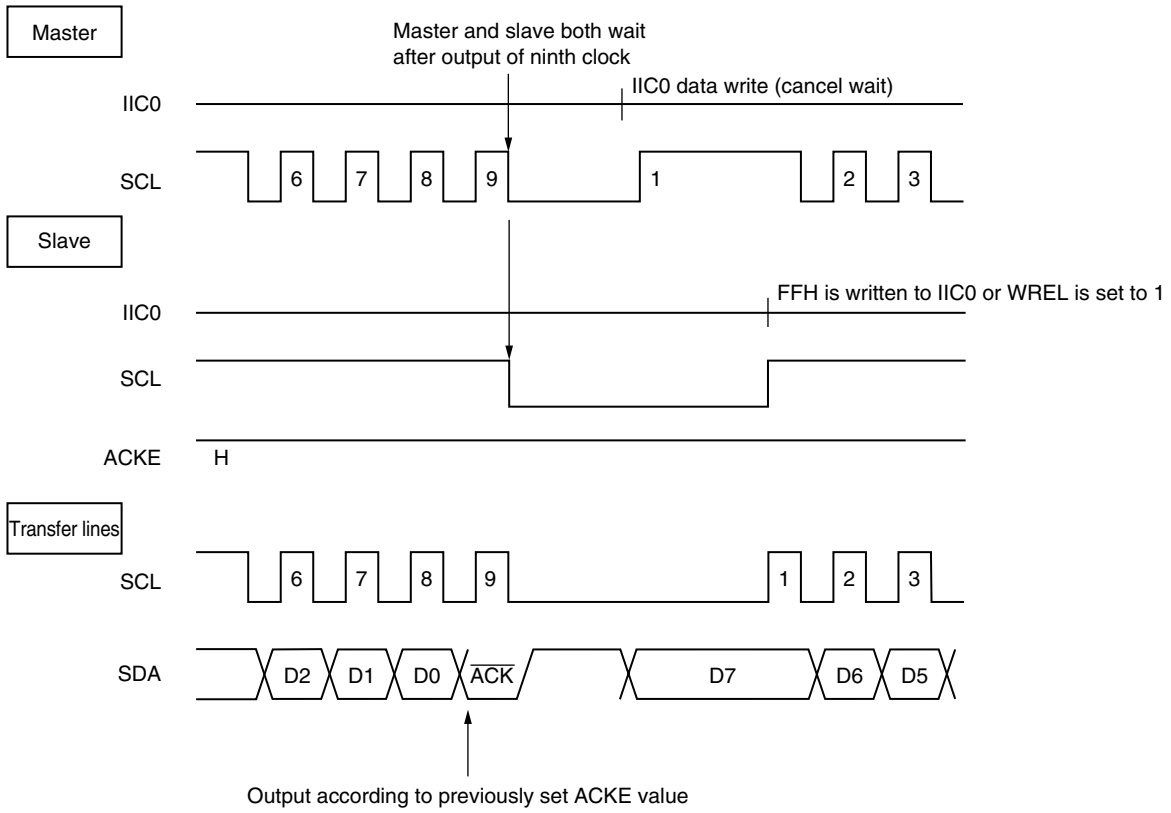


Figure 10-14. Wait Signal (2/2)

(b) When master and slave devices both have a nine-clock wait

(master: transmission, slave: reception, and ACKE = 1)



Remark ACKE: Bit 2 of IIC control register 0 (IICC0)
WREL: Bit 5 of IIC control register 0 (IICC0)

A wait may be automatically generated depending on the setting for bit 3 (WTIM) of IIC control register 0 (IICC0).

Normally, when bit 5 (WREL) of IICC0 is set to 1 or when FFH is written to IIC shift register 0 (IIC0), the wait status is canceled and the transmitting side writes data to IIC0 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 IICC0 to 1
- By setting bit 0 (SPT) of IICC0 to 1

10.3.5 I^2C interrupt request (INTIIC0)

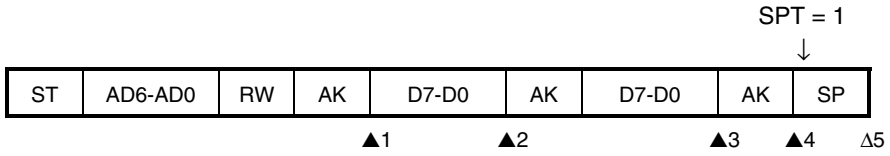
The following shows the value of IIC status register 0 (IICS0) at the INTIIC0 interrupt request generation timing and at the INTIIC0 interrupt timing.

Remark The interrupt control register of INTIIC0 is alternately used as the interrupt control register (CSIC0) of INTCSI0. An IICIC0 register does not exist.

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When WTIM = 0



▲1: IICS0 = 10XXX110B

▲2: IICS0 = 10XXX000B

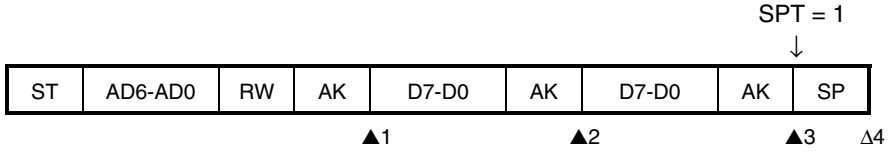
▲3: IICS0 = 10XXX000B (WTIM = 1)

▲4: IICS0 = 10XXXX00B

Δ 5: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 10XXX110B

▲2: IICS0 = 10XXX100B

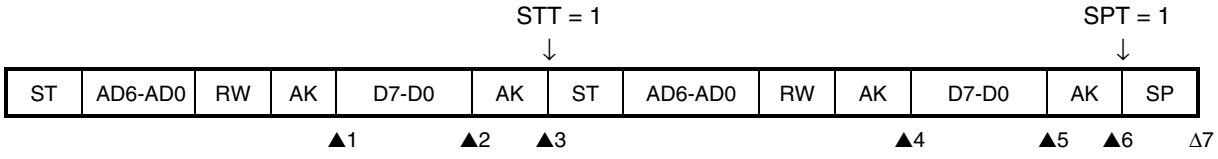
▲3: IICS0 = 10XXXX00B

Δ 4: IICS0 = 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



▲1: IICS0 = 10XXX110B

▲2: IICS0 = 10XXX000B (WTIM = 1)

▲3: IICS0 = 10XXXX00B (WTIM = 0)

▲4: IICS0 = 10XXX110B

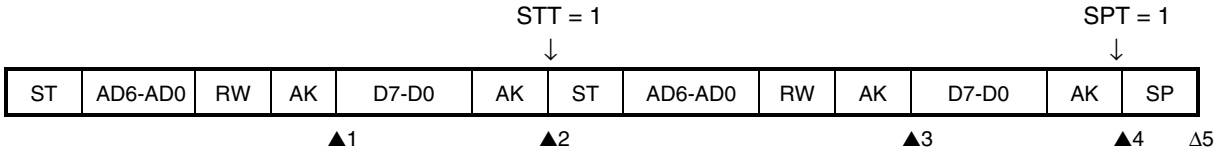
▲5: IICS0 = 10XXX000B (WTIM = 1)

▲6: IICS0 = 10XXXX00B

Δ 7: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 10XXX110B

▲2: IICS0 = 10XXXX00B

▲3: IICS0 = 10XXX110B

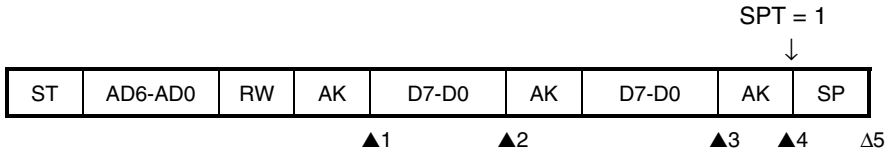
▲4: IICS0 = 10XXXX00B

Δ 5: IICS0 = 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: IICS0 = 1010X110B

▲2: IICS0 = 1010X000B

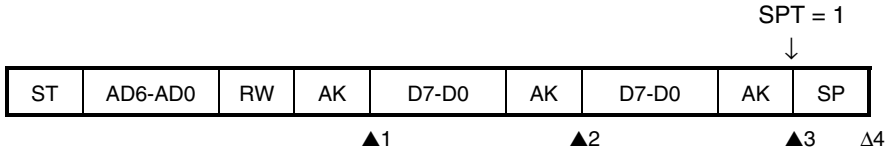
▲3: IICS0 = 1010X000B (WTIM = 1)

▲4: IICS0 = 1010XX00B

Δ 5: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 1010X110B

▲2: IICS0 = 1010X100B

▲3: IICS0 = 1010XX00B

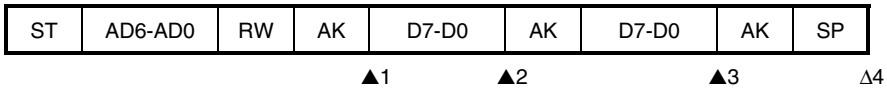
Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

(2) Slave device operation (when receiving slave address data (matches SVA0))

(a) Start ~ Address ~ Data ~ Data ~ Stop

<1> When WTIM = 0



▲1: IICS0 = 0001X110B

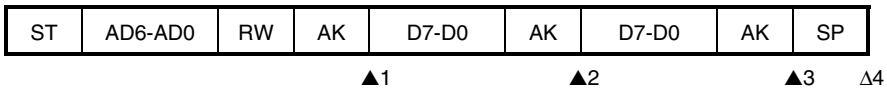
▲2: IICS0 = 0001X000B

▲3: IICS0 = 0001X000B

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 0001X110B

▲2: IICS0 = 0001X100B

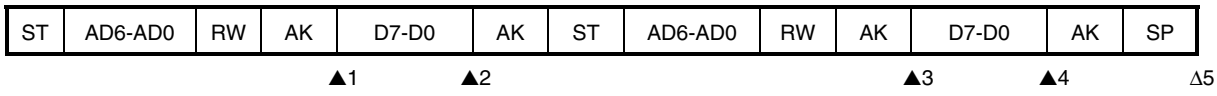
▲3: IICS0 = 0001XX00B

Δ 4: IICS0 = 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 SVA0)



▲1: IICS0 = 0001X110B

▲2: IICS0 = 0001X000B

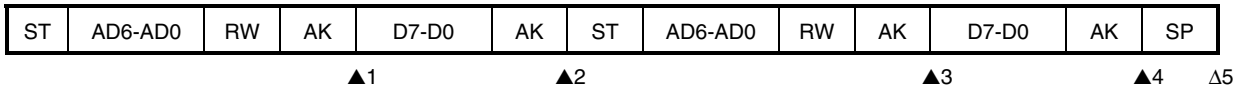
▲3: IICS0 = 0001X110B

▲4: IICS0 = 0001X000B

Δ 5: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1 (after restart, matches SVA0)



▲1: IICS0 = 0001X110B

▲2: IICS0 = 0001XX00B

▲3: IICS0 = 0001X110B

▲4: IICS0 = 0001XX00B

Δ 5: IICS0 = 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-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
				▲1	▲2				▲3		▲4	Δ5

▲1: IICS0 = 0001X110B

▲2: IICS0 = 0001X000B

▲3: IICS0 = 0010X010B

▲4: IICS0 = 0010X000B

Δ 5: IICS0 = 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-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
				▲1		▲2			▲3	▲4		▲5 Δ6

▲1: IICS0 = 0001X110B

▲2: IICS0 = 0001XX00B

▲3: IICS0 = 0010X010B

▲4: IICS0 = 0010X110B

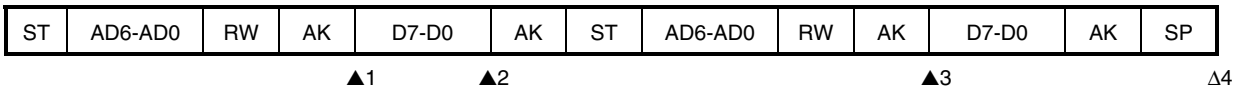
▲5: IICS0 = 0010XX00B

Δ 6: IICS0 = 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 address (= not extension code))



▲1: IICS0 = 0001X110B

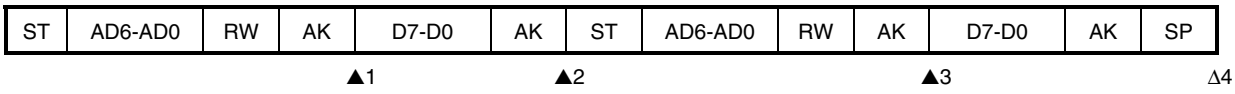
▲2: IICS0 = 0001X000B

▲3: IICS0 = 00000X10B

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0001X110B

▲2: IICS0 = 0001XX00B

▲3: IICS0 = 00000X10B

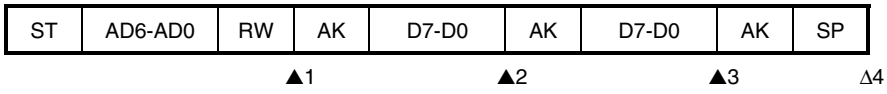
Δ 4: IICS0 = 00000001B

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



▲1: IICS0 = 0010X010B

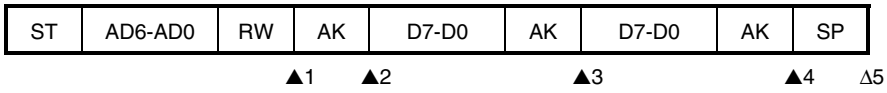
▲2: IICS0 = 0010X000B

▲3: IICS0 = 0010X000B

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 0010X010B

▲2: IICS0 = 0010X110B

▲3: IICS0 = 0010X100B

▲4: IICS0 = 0010XX00B

Δ 5: IICS0 = 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 SVA0)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1		▲2					▲3	▲4	Δ5

▲1: IICS0 = 0010X010B

▲2: IICS0 = 0010X000B

▲3: IICS0 = 0001X110B

▲4: IICS0 = 0001X000B

Δ 5: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1 (after restart, matches SVA0)

ST	AD6-AD0	RW	AK	D7-D0	AK	ST	AD6-AD0	RW	AK	D7-D0	AK	SP
			▲1	▲2		▲3				▲4	▲5	Δ6

▲1: IICS0 = 0010X010B

▲2: IICS0 = 0010X110B

▲3: IICS0 = 0010XX00B

▲4: IICS0 = 0001X110B

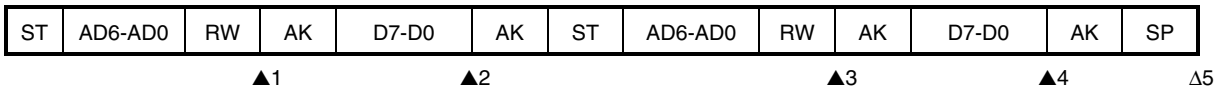
▲5: IICS0 = 0001XX00B

Δ 6: IICS0 = 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)



▲1: IICS0 = 0010X010B

▲2: IICS0 = 0010X000B

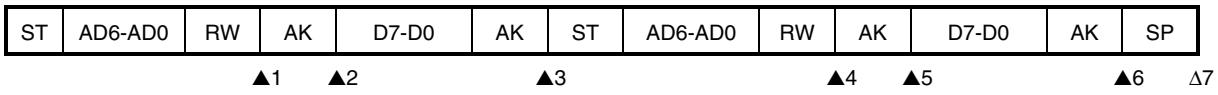
▲3: IICS0 = 0010X010B

▲4: IICS0 = 0010X000B

Δ 5: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1 (after restart, extension code reception)



▲1: IICS0 = 0010X010B

▲2: IICS0 = 0010X110B

▲3: IICS0 = 0010XX00B

▲4: IICS0 = 0010X010B

▲5: IICS0 = 0010X110B

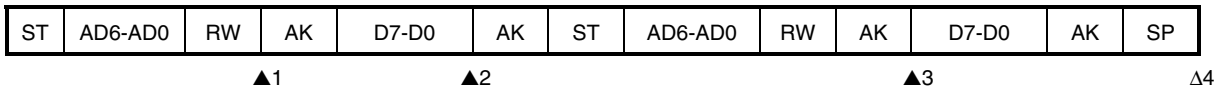
▲6: IICS0 = 0010XX00B

Δ 7: IICS0 = 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 address (= not extension code))



▲1: IICS0 = 0010X010B

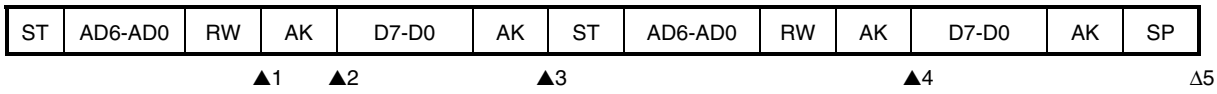
▲2: IICS0 = 0010X000B

▲3: IICS0 = 00000X10B

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0010X010B

▲2: IICS0 = 0010X110B

▲3: IICS0 = 0010XX00B

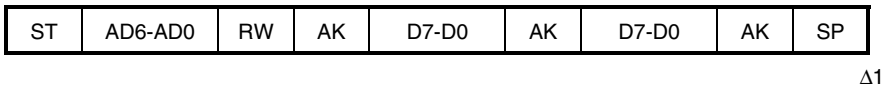
▲4: IICS0 = 00000X10B

Δ 5: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



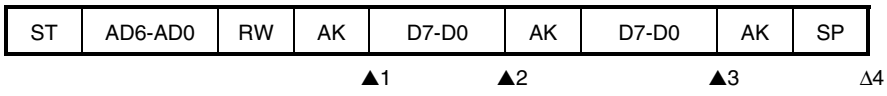
Δ 1: IICS0 = 00000001B

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



▲1: IICS0 = 0101X110B (Example: when ALD is read during interrupt servicing)

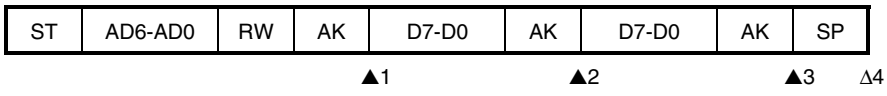
▲2: IICS0 = 0001X000B

▲3: IICS0 = 0001X000B

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 0101X110B (Example: when ALD is read during interrupt servicing)

▲2: IICS0 = 0001X100B

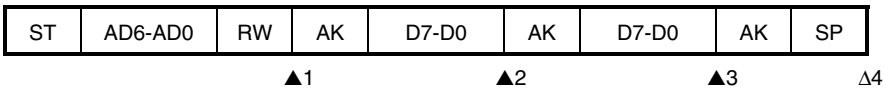
▲3: IICS0 = 0001XX00B

Δ 4: IICS0 = 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



▲1: IICS0 = 0110X010B (Example: when ALD is read during interrupt servicing)

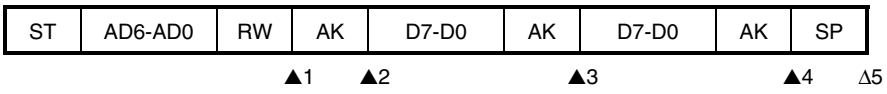
▲2: IICS0 = 0010X000B

▲3: IICS0 = 0010X000B

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

<2> When WTIM = 1



▲1: IICS0 = 0110X010B (Example: when ALD is read during interrupt servicing)

▲2: IICS0 = 0010X110B

▲3: IICS0 = 0010X100B

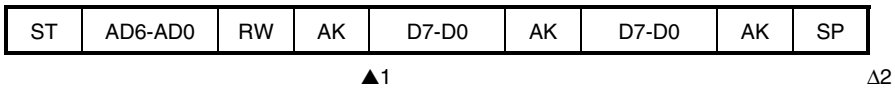
▲4: IICS0 = 0010XX00B

Δ 5: IICS0 = 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

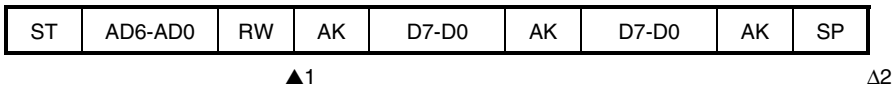


▲1: IICS0 = 01000110B (Example: when ALD is read during interrupt servicing)

Δ 2: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1

(b) When arbitration loss occurs during transmission of extension code



▲1: IICS0 = 0110X010B (Example: when ALD is read during interrupt servicing)

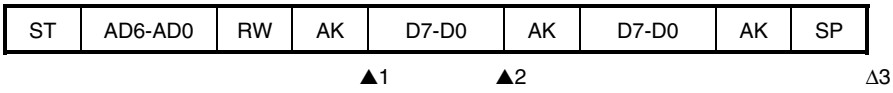
IICC0: LREL is set to 1 by software

Δ 2: IICS0 = 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



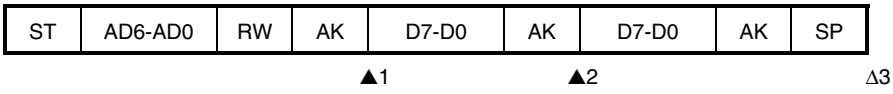
▲1: IICS0 = 10001110B

▲2: IICS0 = 01000000B (Example: when ALD is read during interrupt servicing)

Δ 3: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1

<2> When WTIM = 1



▲1: IICS0 = 10001110B

▲2: IICS0 = 01000100B (Example: when ALD is read during interrupt servicing)

Δ 3: IICS0 = 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: does not match SVA0)

ST	AD6-AD0	RW	AK	D7-Dn	ST	AD6-AD0	RW	AK	D7-D0	AK	SP

▲1 ▲2 Δ3

▲1: IICS0 = 1000X110B

▲2: IICS0 = 01000110B (Example: when ALD is read during interrupt servicing)

Δ 3: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care
Dn = D6 to D0

<2> Extension code

ST	AD6-AD0	RW	AK	D7-Dn	ST	AD6-AD0	RW	AK	D7-D0	AK	SP

▲1 ▲2 Δ3

▲1: IICS0 = 1000X110B

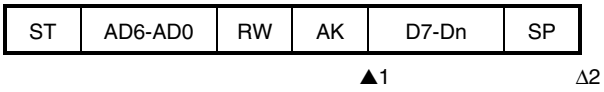
▲2: IICS0 = 0110X010B (Example: when ALD is read during interrupt servicing)

IICC0: LREL is set to 1 by software

Δ 3: IICS0 = 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



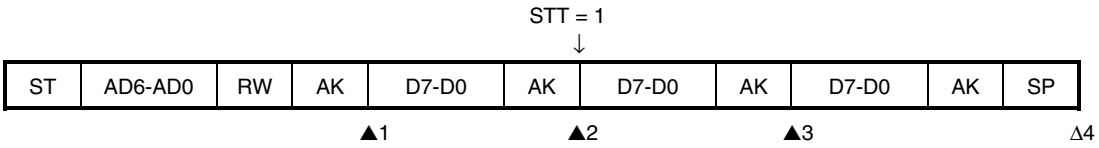
▲1: IICS0 = 1000X110B

Δ 2: IICS0 = 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



▲1: IICS0 = 1000X110B

▲2: IICS0 = 1000XX00B

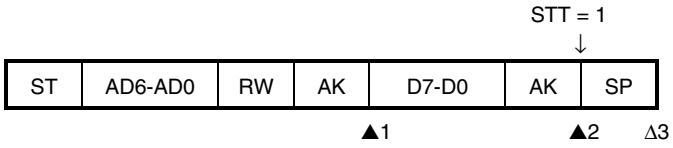
▲3: IICS0 = 01000100B (Example: when ALD is read during interrupt servicing)

Δ 4: IICS0 = 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: IICS0 = 1000X110B

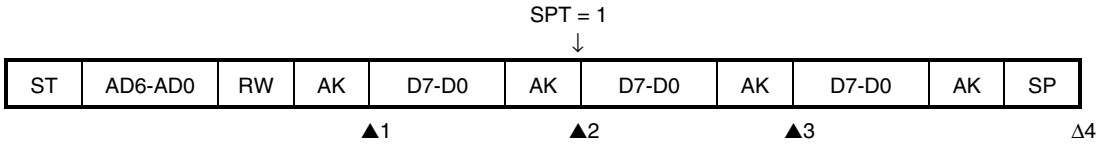
▲2: IICS0 = 1000XX00B

Δ 3: IICS0 = 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: IICS0 = 1000X110B

▲2: IICS0 = 1000XX00B

▲3: IICS0 = 01000000B (Example: when ALD is read during interrupt servicing)

Δ 4: IICS0 = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE = 1
X: don't care

10.3.6 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM) of IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown below.

Table 10-3. INTIICn 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 INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register (SVA0).
At this point, $\overline{ACK}$ is output regardless of the value set to bit 2 (ACKE) of IICC0. For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of the slave address register (SVA0), neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined regardless of the WTIM bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WREL) of IIC control register 0 (IICC0) to 1
- By writing to IIC shift register 0 (IIC0)
- By setting start condition (bit 1 (STT) of IIC control register 0 (IICC0) = 1)
- By setting stop condition (bit 0 (SPT) of IIC control register 0 (IICC0) = 1)

When an 8-clock wait has been selected (WTIM = 0), the output level of $\overline{ACK}$ must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC0 is generated when a stop condition is detected.

10.3.7 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An interrupt request (INTIIC0) occurs when a local address has been set to slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

10.3.8 Error detection

In I^2C bus mode, the status of the serial data bus (SDA) during data transmission is captured by IIC shift register 0 (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

10.3.9 Extension code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code reception flag (EXC) is set for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock.
The local address stored in slave address register 0 (SVA0) is not affected.

(2) If 11110xx0 is set to SVA0 by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that INTIIC0 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 0 (IICS0)
COI: Bit 4 of IIC status register 0 (IICS0)

(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 of IIC control register 0 (IICC0) to LREL = 1 and the CPU will enter the next communication wait state.

Table 10-4. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for a different bus format
1111 0xx	X	10-bit slave address specification

10.3.10 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 IIC status register 0 (IICS0) is set at 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 **10.3.5 I²C interrupt request (INTIIC0)**.

Note STD: Bit 1 of IIC status register 0 (IICS0)
STT: Bit 1 of IIC control register 0 (IICC0)

Figure 10-15. Arbitration Timing Example

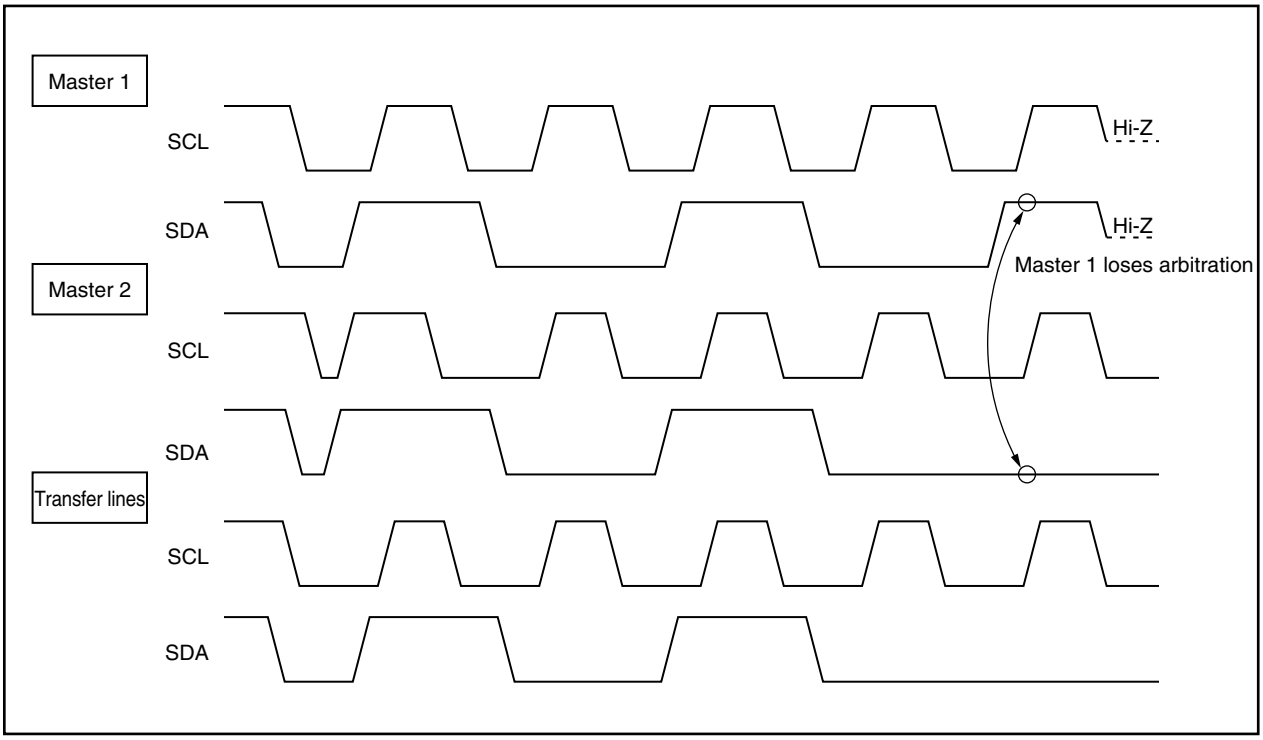


Table 10-5. Status During Arbitration and Interrupt Request Generation Timing

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 $\overline{ACK}$ 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 0 (IICC0)) = 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 0 (IICC0)

10.3.11 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request (INTIIC0) 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 IIC control register 0 (IICC0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

10.3.12 Communication reservation

To start master device communications when not currently using the 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 IIC control register 0 (IICC0) was set to 1).

If bit 1 (STT) of IICC0 is set while the bus is not used, a start condition is automatically generated and a 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 IIC shift register 0 (IIC0) causes the master's address transfer to start. At this point bit 4 (SPIE) of IICC0 should be set.

When STT 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 STT, set STT, wait for the wait period, then check MSTS (bit 7 of IIC status register 0 (IICS0)).

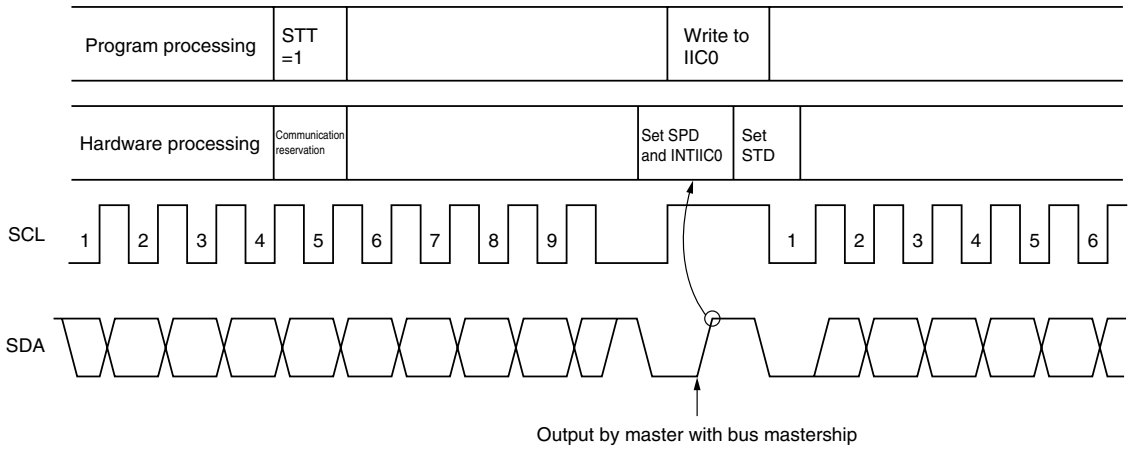
Wait periods, which should be set via software, are listed in Table 10-6. These wait periods can be set via the settings for bits 3, 1, and 0 (SMC, CL1, and CL0) of IIC clock select register 0 (IICCL0).

Table 10-6. 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.

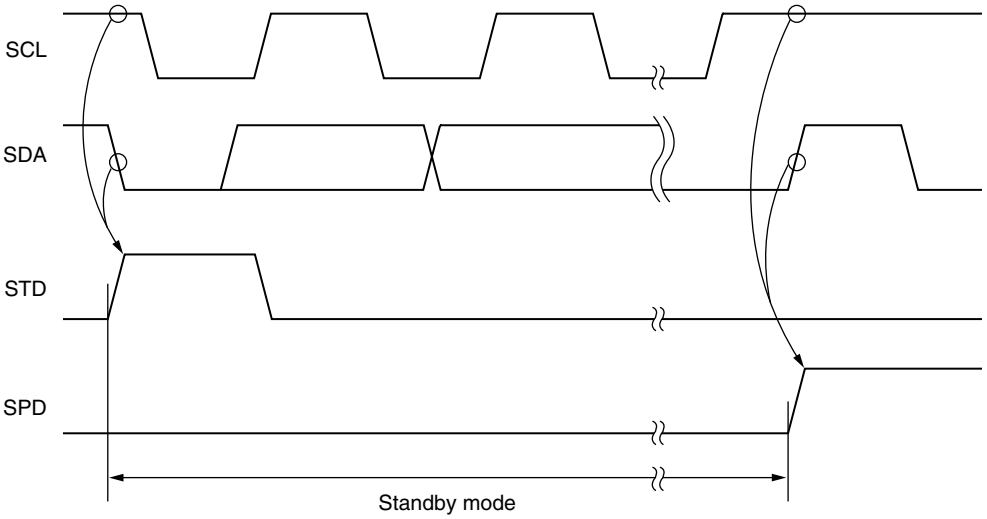
Figure 10-16. Communication Reservation Timing



IIC0: IIC shift register 0
STT0: Bit 1 of IIC control register 0 (IICC0)
STD0: Bit 1 of IIC status register 0 (IICS0)
SPD0: Bit 0 of IIC status register 0 (IICS0)

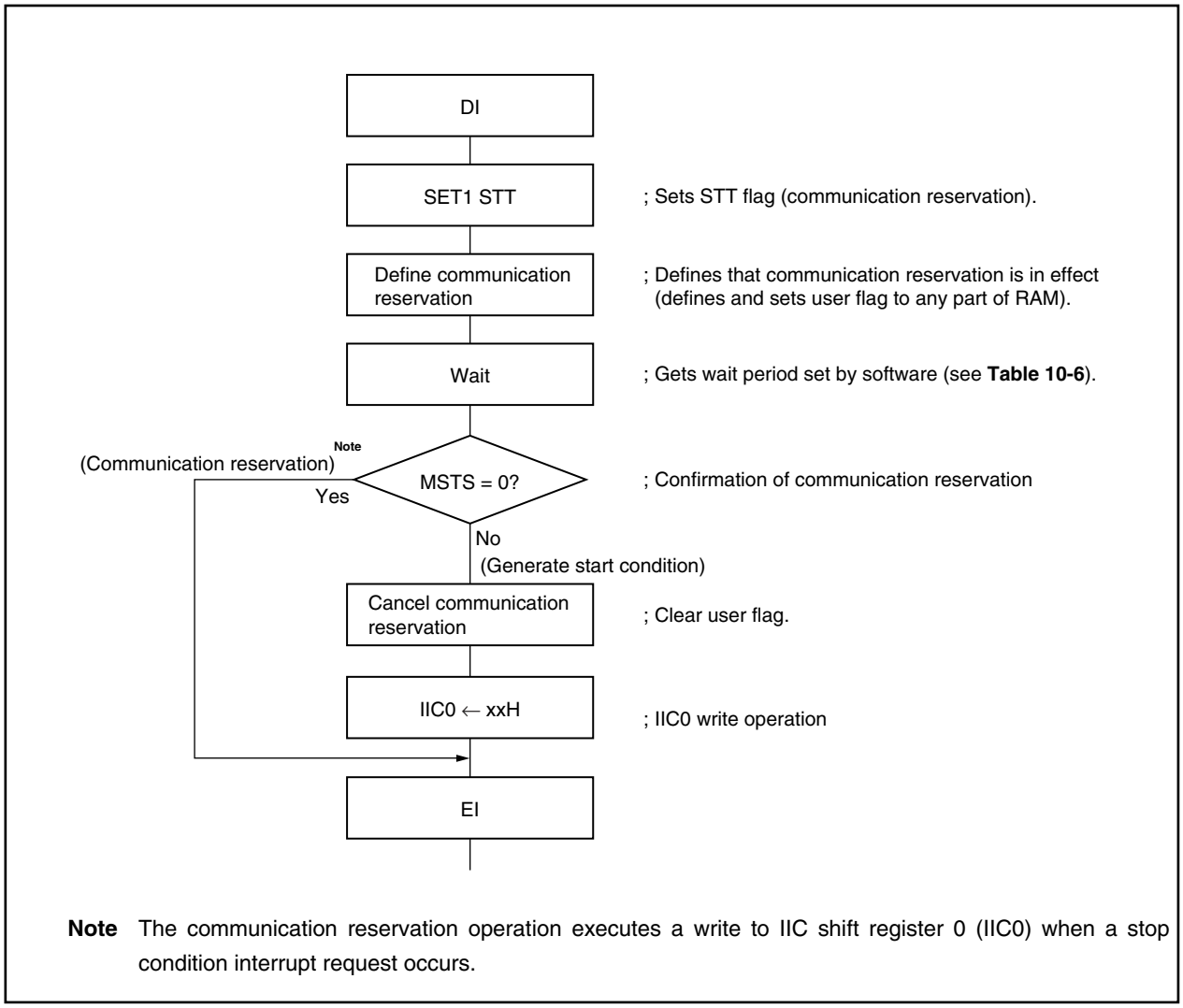
Communication reservations are acknowledged at the following timing. After bit 1 (STD) of IIC status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT) of IIC control register 0 (IICC0) to 1 before a stop condition is detected.

Figure 10-17. Timing for Acknowledging Communication Reservations



The communication reservation flow chart is illustrated below.

Figure 10-18. Communication Reservation Flow Chart



Note The communication reservation operation executes a write to IIC shift register 0 (IIC0) when a stop condition interrupt request occurs.

10.3.13 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.

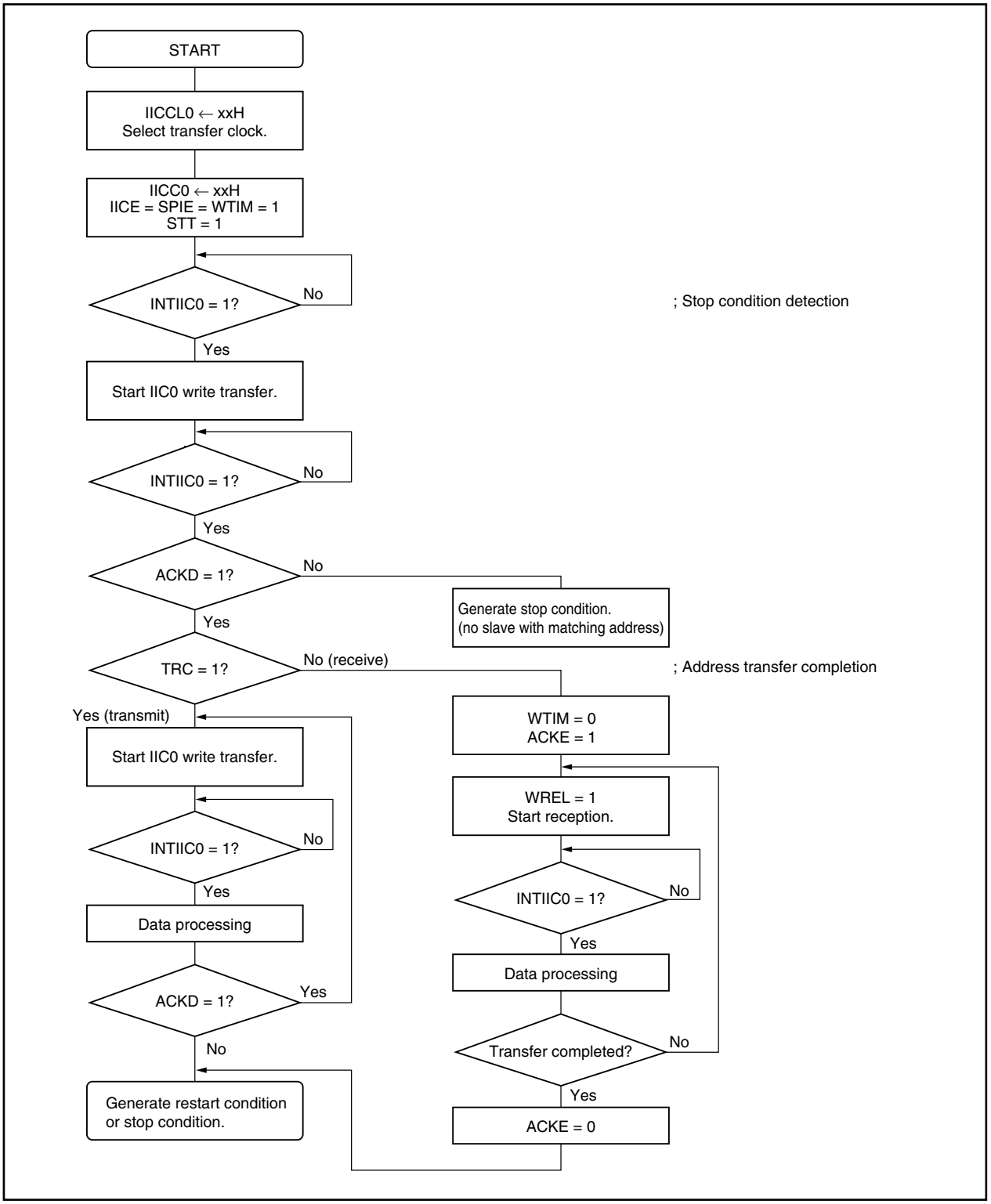
(a) Set IIC clock select register 0 (IICCL0).
(b) Set bit 7 (IICE) of IIC control register 0 (IICC0).
(c) Set bit 0 of IICC0.

10.3.14 Communication operations

(1) Master operations

The following is a flow chart of the master operations.

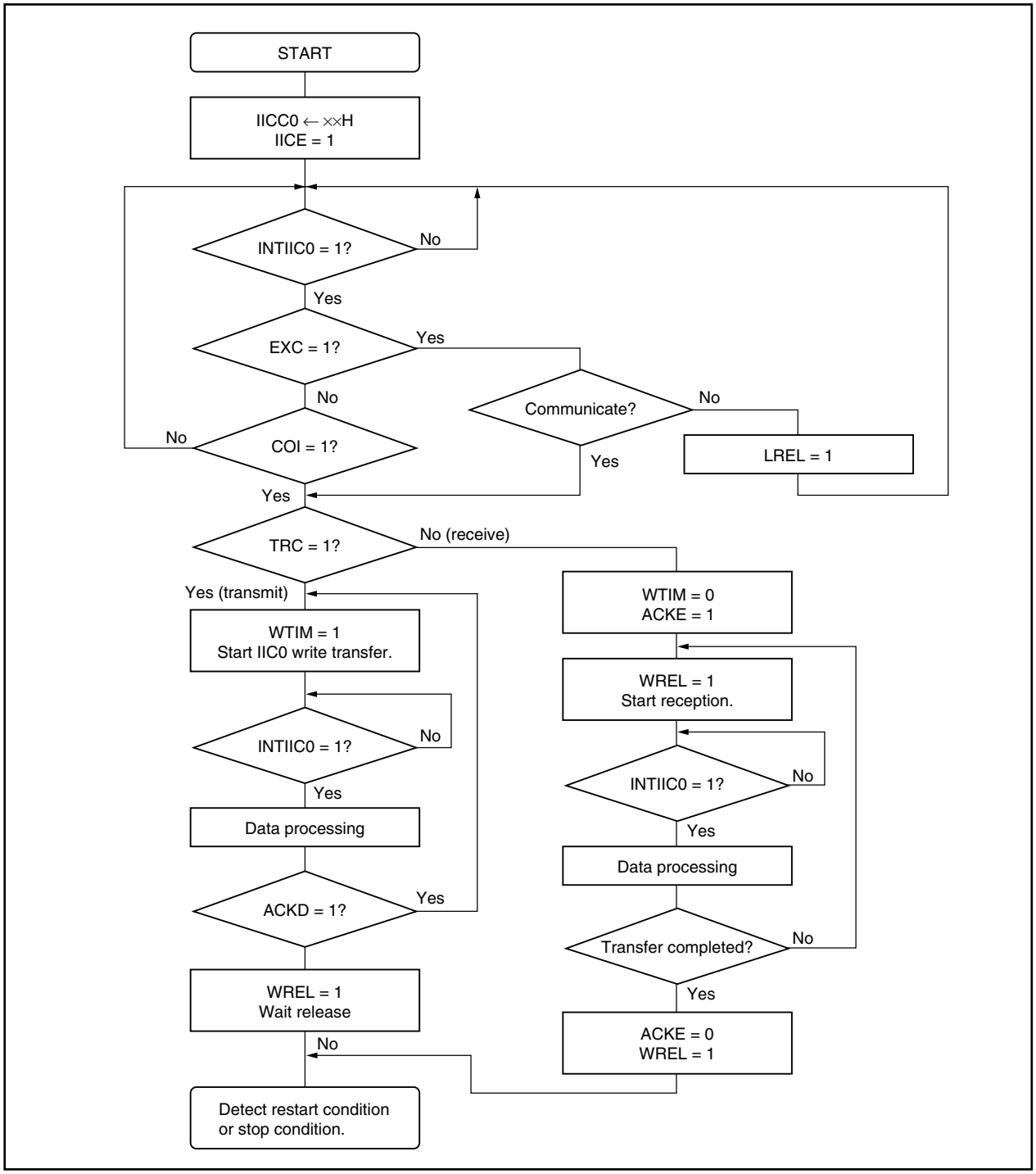
Figure 10-19. Master Operation Flow Chart



(2) Slave operation

The following is a flow chart of the slave operations.

Figure 10-20. Slave Operation Flow Chart



10.3.15 Timing of data communication

When using I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC bit (bit 3 of IIC status register 0 (IICS0)) that specifies the data transfer direction and then starts serial communication with the slave device.

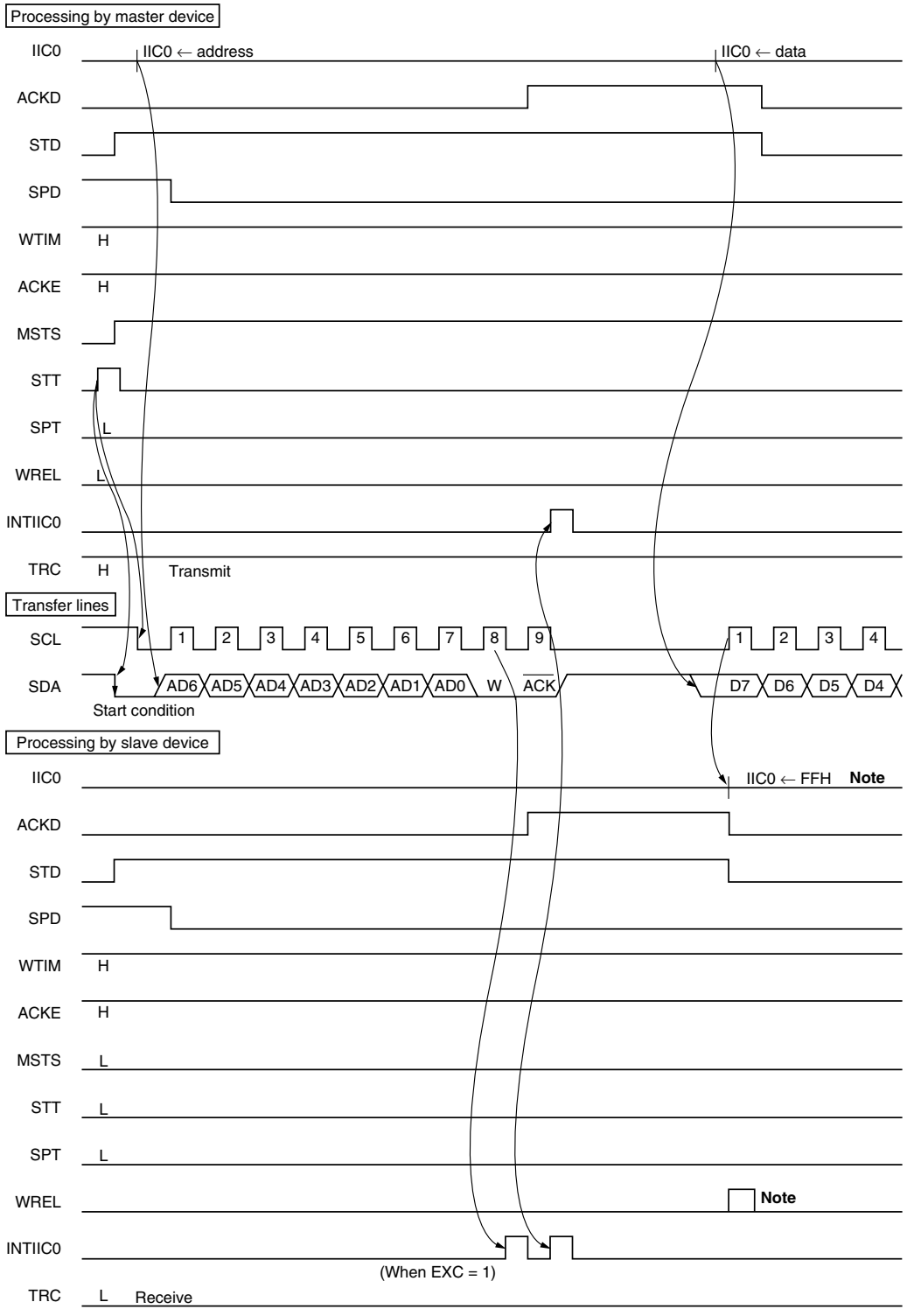
The shift operation of IIC bus shift register 0 (IIC0) is synchronized with the falling edge of the serial clock (SCL). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA pin.

Data input via the SDA pin is captured by IIC0 at the rising edge of SCL.

The following shows the timing charts of data communication.

Figure 10-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

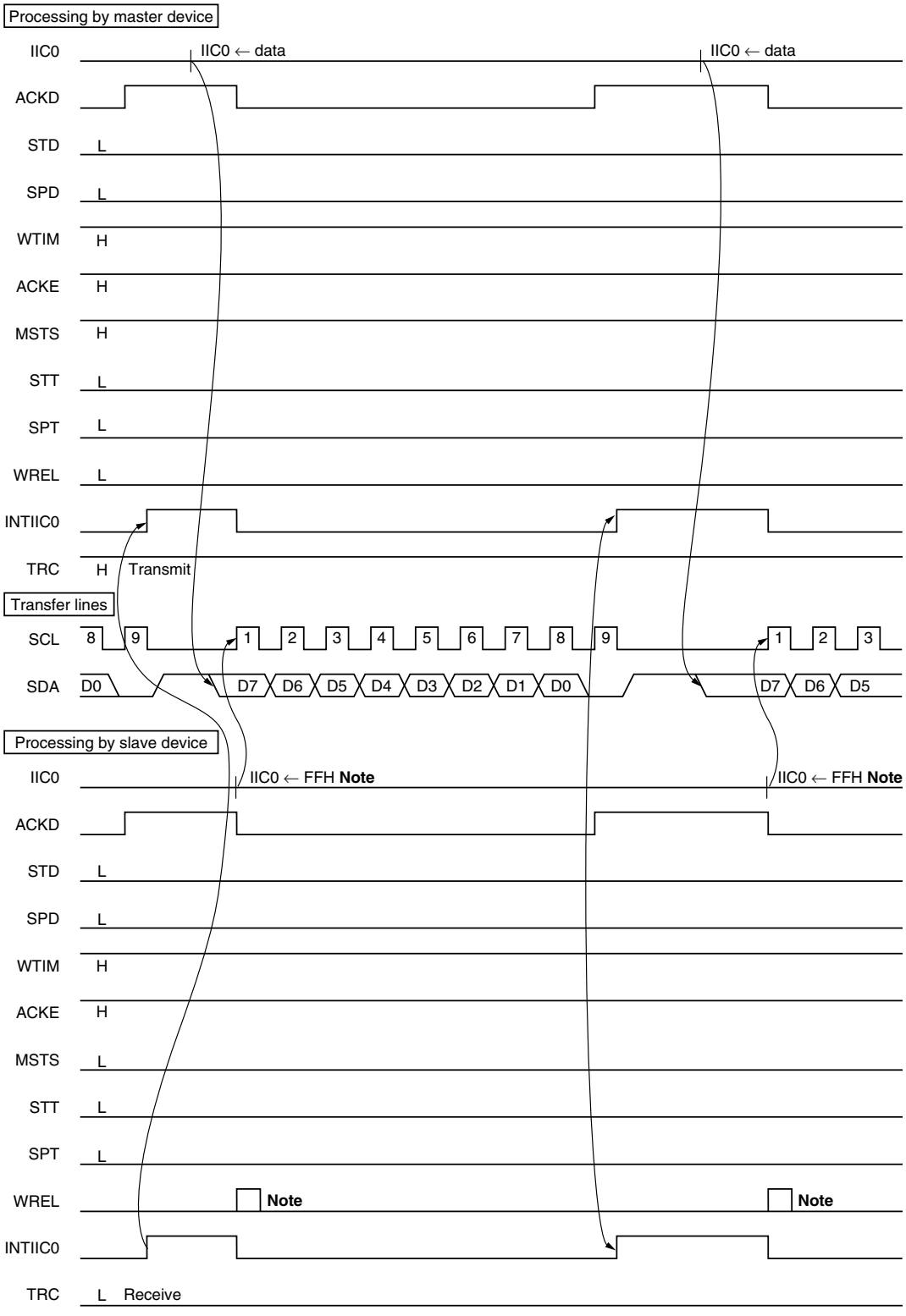
(a) Start condition ~ address



Note To cancel slave wait, write FFH to IIC0 or set WREL.

Figure 10-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

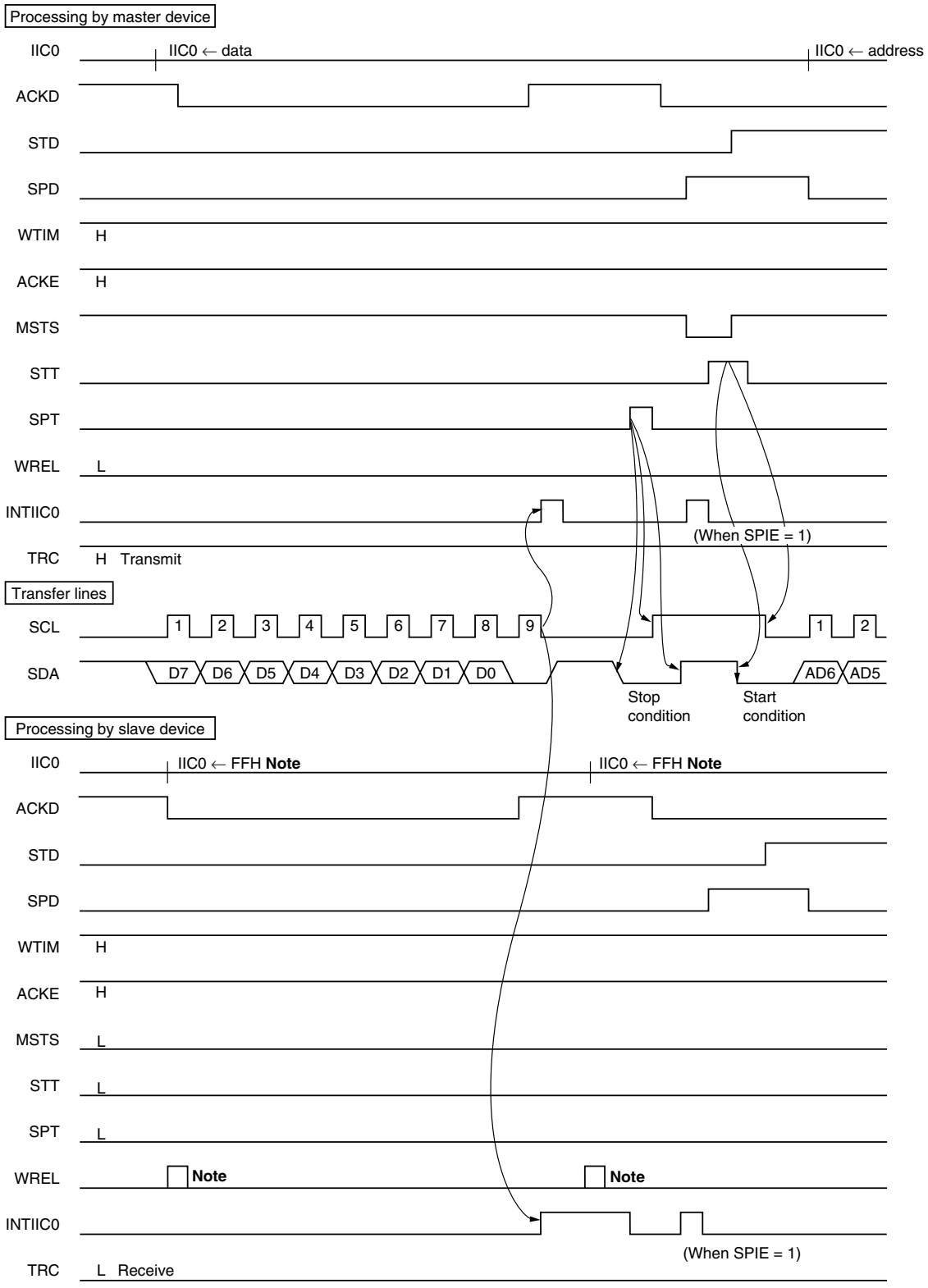
(b) Data



Note To cancel slave wait, write FFH to IIC0 or set WREL.

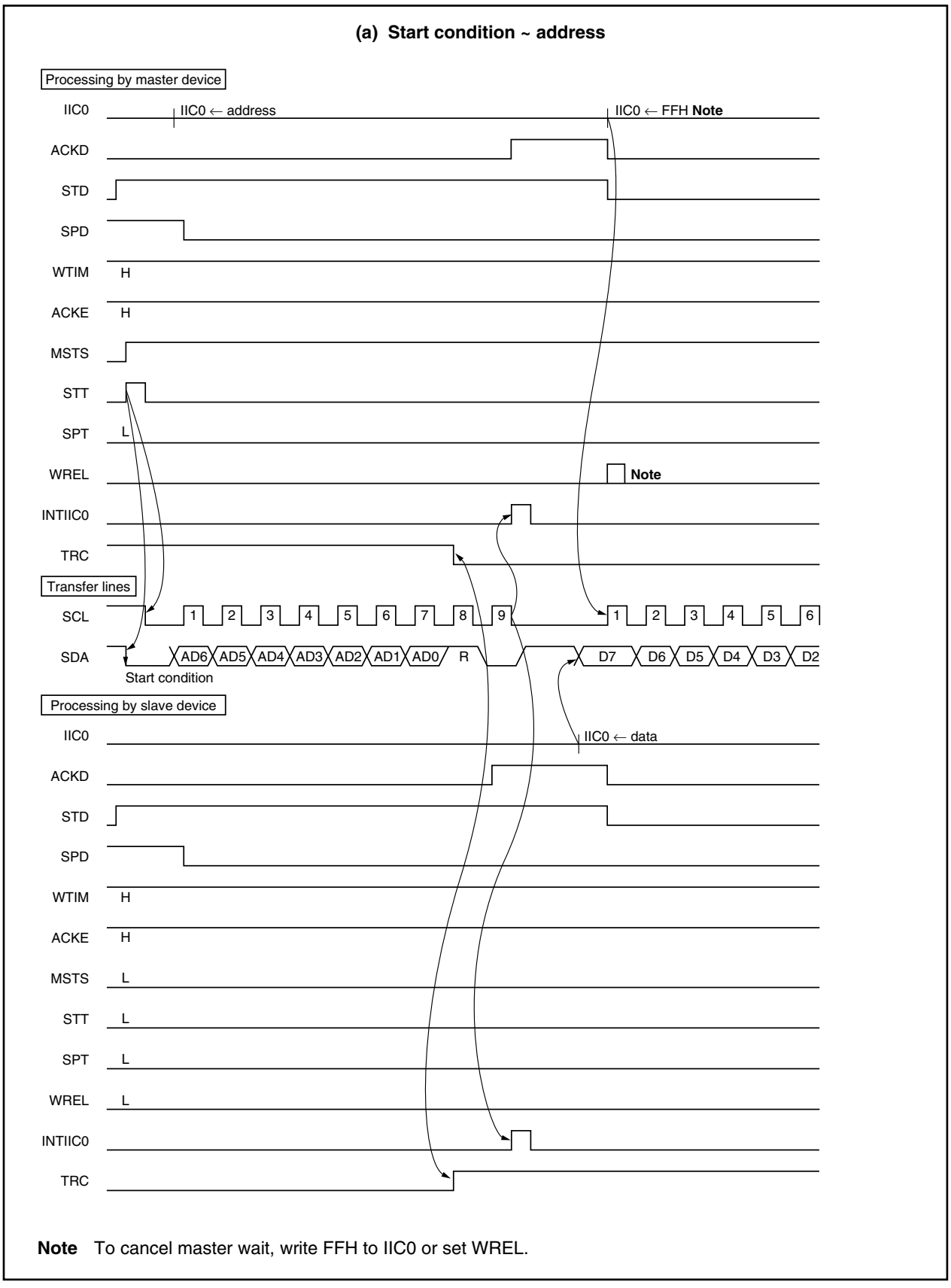
Figure 10-21. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition



Note To cancel slave wait, write FFH to IIC0 or set WREL.

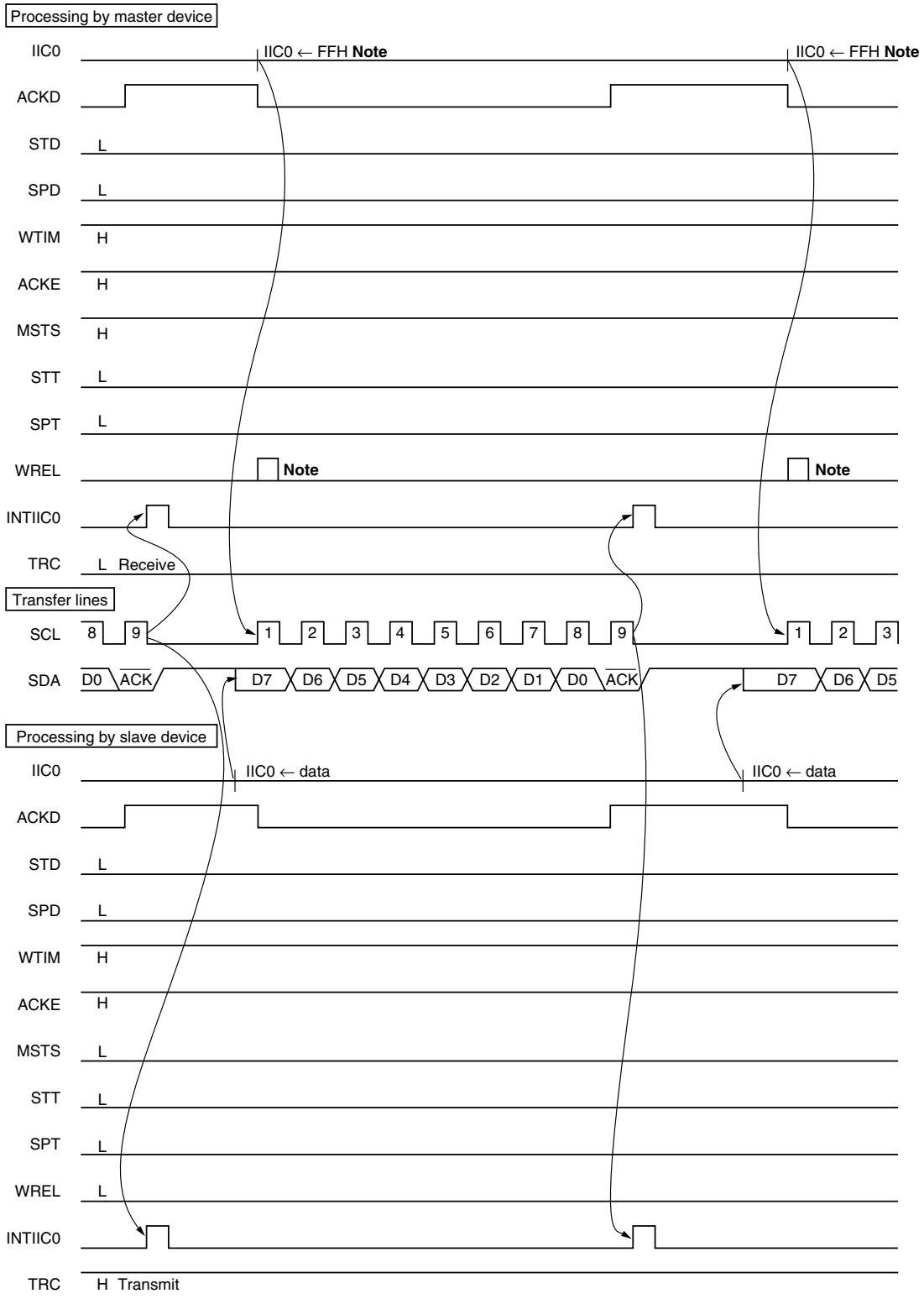
Figure 10-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)



Note To cancel master wait, write FFH to IIC0 or set WREL.

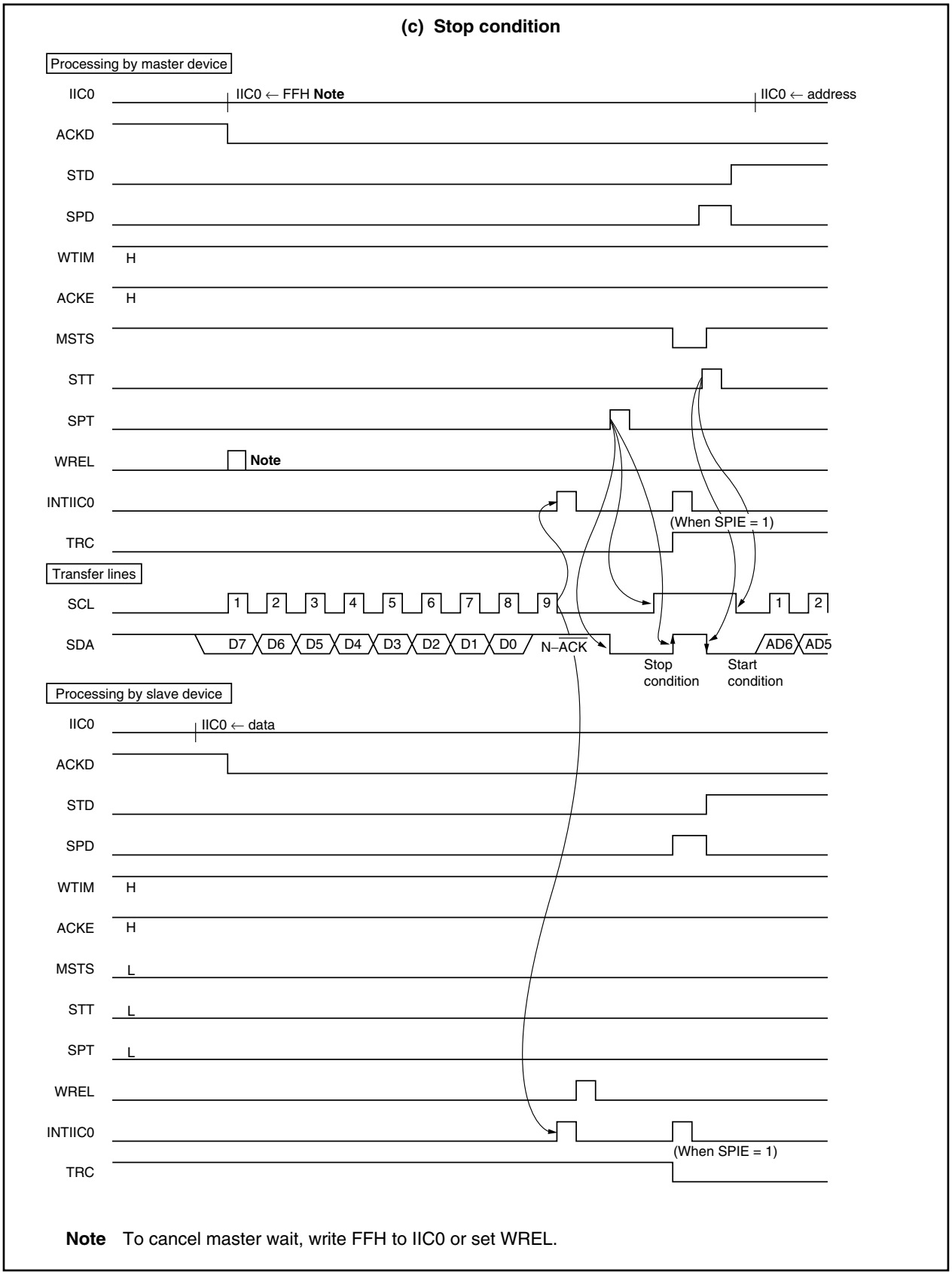
Figure 10-22. Example of Slave to Master Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data



Note To cancel master wait, write FFH to IIC0 or set WREL.

Figure 10-22. 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 IIC0 or set WREL.

10.4 Asynchronous Serial Interface (UART0, UART1)

UARTn (n = 0, 1) has the following two operation modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) Asynchronous serial interface (UART) mode

This mode enables full-duplex operation in which one byte of data is transmitted and received after the start bit.

The on-chip dedicated UARTn baud rate generator enables communications using a wide range of selectable baud rates. In addition, a baud rate based on divided clock input to the ASCKn pin can also be defined.

The UARTn baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

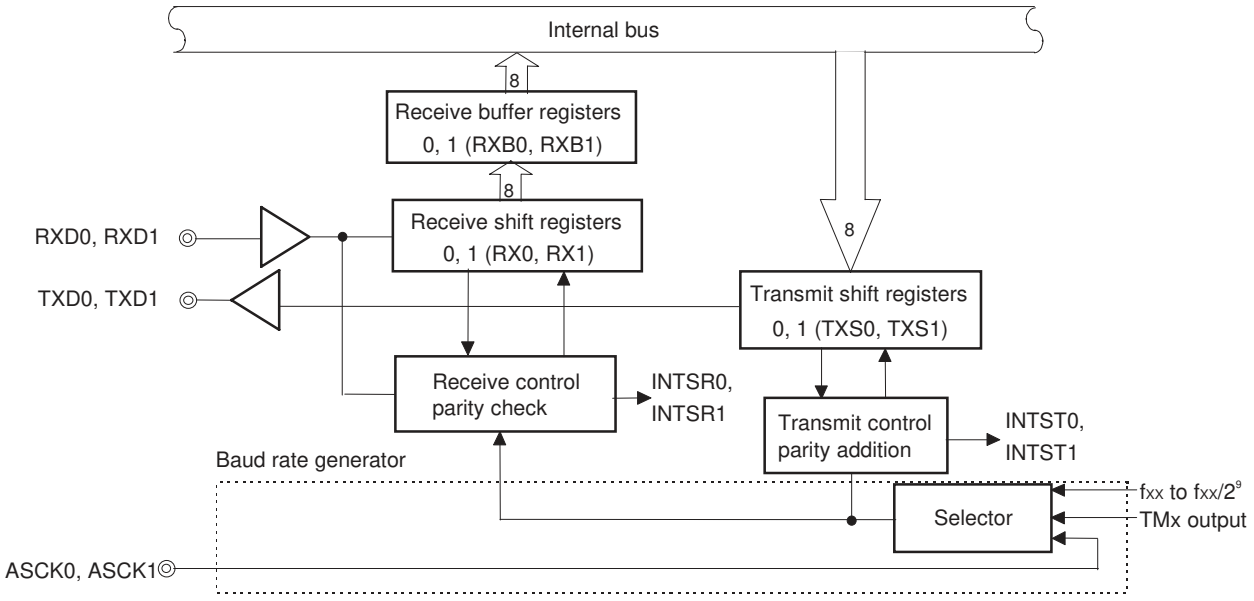
10.4.1 Configuration

UARTn consists of the following hardware.

Table 10-7. Configuration of UARTn

Item	Configuration
Registers	Transmit shift registers 0, 1 (TXS0, TXS1) Receive buffer registers 0, 1 (RXB0, RXB1)
Control registers	Asynchronous serial interface mode registers 0, 1 (ASIM0, ASIM1) Asynchronous serial interface status registers 0, 1 (ASIS0, ASIS1) Baud rate generator control registers 0, 1 (BRGC0, BRGC1) Baud rate generator mode control registers 0, 1 (BRGMC0, BRGMC1) Baud rate generator mode control register 01 (BRGMC01)

Figure 10-23. Block Diagram of UARTn



Remark TMx output is as follows:
UART0: TM3
UART1: TM2

(1) Transmit shift registers 0, 1 (TXS0, TXS1)

TXSn is the register for setting transmit data. Data written to TXSn is transmitted as serial data.

When the data length is set to 7 bits, bit 0 to bit 6 of the data written to TXSn is transmitted as serial data. Writing data to TXSn starts the transmit operation.

TXSn can be written to by an 8-bit memory manipulation instruction. It cannot be read from.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Caution Do not write to TXSn during a transmit operation.

(2) Receive shift registers 0, 1 (RX0, RX1)

The RXn register converts serial data input via the RXD0 and RXD1 pins into parallel data. When one byte of data is received at RXn, the received data is transferred to receive buffer registers 0 and 1 (RXB0, RXB1).

RX0 and RX1 cannot be manipulated directly by a program.

(3) Receive buffer registers 0, 1 (RXB0, RXB1)

RXBn is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred.

When the data length is set to 7 bits, received data is sent to bit 0 to bit 6 of RXBn. In RXBn, the MSB must be set to 0.

RXBn can be read by an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input sets RXBn to FFH.

(4) Transmission controller

The transmission controller controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to transmit shift register n (TXSn), based on the values set to asynchronous serial interface mode register n (ASIMn).

(5) Reception controller

The reception controller controls receive operations based on the values set to asynchronous serial interface mode register n (ASIMn). During a receive operation, it performs error checking, such as for parity errors, and sets various values to asynchronous serial interface status register n (ASISn) according to the type of error that is detected.

10.4.2 UARTn control registers

UARTn uses the following four types of registers for control functions (n = 0, 1).

- Asynchronous serial interface mode register n (ASIMn)
- Asynchronous serial interface status register n (ASISn)
- Baud rate generator control register n (BRGCn)
- Baud rate generator mode control registers n and 01 (BRGMCn, BRGMC01)

(1) Asynchronous serial interface mode registers 0, 1 (ASIM0, ASIM1)

ASIMn is an 8-bit register that controls the serial transfer operations of UARTn.

ASIMn can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

After reset: 00H R/W Address: FFFFF300H, FFFFF310H

	7	6	5	4	3	2	1	0
ASIMn	TXEn	RXEn	PS1n	PS0n	UCLn	SLn	ISRMn	0

(n = 0, 1)

TXEn	RXEn	Operation Mode	RXDn/Pxx Pin Function	TXDn/Pxx Pin Function
0	0	Operation stop	Port function	Port function
0	1	UART mode (receive only)	Serial function	Port function
1	0	UART mode (transmit only)	Port function	Serial function
1	1	UART mode (transmit and receive)	Serial function	Serial function

PS1n	PS0n	Parity Bit Specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

UCLn	Character Length Specification
0	7 bits
1	8 bits

SLn	Stop Bit Length Specification for Transmit Data
0	1 bit
1	2 bits

ISRMn	Receive Completion Interrupt Control when Error Occurs
0	Receive completion interrupt is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

Cautions 1. Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

2. Be sure to set bit 0 to 0.

(2) Asynchronous serial interface status registers 0, 1 (ASIS0, ASIS1)

When a receive error occurs in UART mode, these registers indicate the type of error.

ASISn can be read using an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

After reset: 00H R Address: FFFFF302H, FFFFF312H

	7	6	5	4	3	2	1	0
ASISn	0	0	0	0	0	PEn	FEn	OVEn

(n = 0, 1)

PEn	Parity Error Flag
0	No parity error
1	Parity error (Transmit data parity does not match)

FEn	Framing Error Flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVEn	Overrun Error Flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if the stop bit length has been set to two bits by setting bit 2 (SLn) of asynchronous serial interface mode register n (ASIMn), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of receive buffer register n (RXBn) when an overrun error has occurred. Until the contents of RXBn are read, further overrun errors will occur when receiving data.

(3) Baud rate generator control registers 0, 1 (BRGC0, BRGC1)

These registers set the serial clock for UARTn.

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

$\overline{\text{RESET}}$ input sets these registers to 00H.

After reset: 00H R/W Address: FFFFF304H, FFFFF314H

	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	Selection of Input Clock	k
0	0	0	0	0	×	×	×	Setting prohibited	–
0	0	0	0	1	0	0	0	$f_{SCK}/8$	8
0	0	0	0	1	0	0	1	$f_{SCK}/9$	9
0	0	0	0	1	0	1	0	$f_{SCK}/10$	10
0	0	0	0	1	0	1	1	$f_{SCK}/11$	11
0	0	0	0	1	1	0	0	$f_{SCK}/12$	12
0	0	0	0	1	1	0	1	$f_{SCK}/13$	13
0	0	0	0	1	1	1	0	$f_{SCK}/14$	14
0	0	0	0	1	1	1	1	$f_{SCK}/15$	15
0	0	0	1	0	0	0	0	$f_{SCK}/16$	16
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	$f_{SCK}/255$	255

Cautions 1. The value of BRGCn becomes 00H after reset. Before starting operation, select a setting other than "Setting prohibited". Selecting the "Setting prohibited" setting in stop mode does not cause any problems.

2. If write is performed to BRGCn during communication processing, the output of the baud rate generator will be disturbed and communication will not be performed normally. Therefore, do not write to BRGCn during communication processing.

Remark f_{SCK}: Source clock of 8-bit counter

(4) Baud rate generator mode control registers 0, 01 (BRGMC0, BRGMC01)

These registers set the UARTn source clock.

BRGMC0 and BRGMC01 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

After reset: 00H R/W Address: FFFFF320H

	7	6	5	4	3	2	1	0
BRGMC01	0	0	0	0	0	0	0	TPS03

After reset: 00H R/W Address: FFFFF30EH

	7	6	5	4	3	2	1	0
BRGMC0	0	0	0	0	0	TPS02	TPS01	TPS00

TPS03	TPS02	TPS01	TPS00	8-Bit Counter Source Clock Selection	m
0	0	0	0	External clock (ASCK0)	–
0	0	0	1	f_{XX}	0
0	0	1	0	$f_{XX}/2$	1
0	0	1	1	$f_{XX}/4$	2
0	1	0	0	$f_{XX}/8$	3
0	1	0	1	$f_{XX}/16$	4
0	1	1	0	$f_{XX}/32$	5
0	1	1	1	TM3 output	–
1	0	0	0	$f_{XX}/64$	6
1	0	0	1	$f_{XX}/128$	7
1	0	1	0	$f_{XX}/256$	8
1	0	1	1	$f_{XX}/512$	9
1	1	0	0	Setting prohibited	–
1	1	0	1		–
1	1	1	0		–
1	1	1	1		–

Cautions 1. If write is performed to BRGMC0, BRGMC01 during communication processing, the output of the baud rate generator will be disturbed and communication will not be performed normally. Therefore, do not write to BRGMC0, BRGMC01 during communication processing.

2. Be sure to set bits 3 to 7 of the BRGMC0 register to 0.

Remarks 1. Source clock of 8-bit counter: f_{SCK}

2. When the selected clock is the timer output, it is not necessary to set the P27/TI3/TO3 pin to timer output mode.

(5) Baud rate generator mode control register 1 (BRGMC1)

This register sets the UART1 source clock.

BRGMC1 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 00H.

After reset: 00H R/W Address: FFFFF31EH

	7	6	5	4	3	2	1	0
BRGMC1	0	0	0	0	0	TPS12	TPS11	TPS10

TPS12	TPS11	TPS10	8-Bit Counter Source Clock Selection	m
0	0	0	External clock (ASCK1)	–
0	0	1	f_{XX}	0
0	1	0	$f_{XX}/2$	1
0	1	1	$f_{XX}/4$	2
1	0	0	$f_{XX}/8$	3
1	0	1	$f_{XX}/16$	4
1	1	0	$f_{XX}/32$	5
1	1	1	TM2 output	–

Cautions 1. If write is performed to BRGMC1 during communication processing, the output of the baud rate generator will be disturbed and communication will not be performed normally. Therefore, do not write to BRGMC1 during communication processing.

2. Be sure to set bits 3 to 7 to 0.

Remarks 1. Source clock of 8-bit counter: f_{SCK}

2. When the selected clock is timer the output, it is not necessary to set the P26/TI2/TO2 pin to timer output mode.

10.4.3 Operations

UARTn has the following two operation modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

In operation stop mode, pins can be used as normal ports.

(a) Register settings

Operation stop mode settings are made via the TXEn and RXEn bits of asynchronous serial interface mode register n (ASIMn).

Figure 10-24. Settings of ASIMn (Operation Stop Mode)

After reset: 00H R/W Address: FFFFF300H, FFFFF310H

	7	6	5	4	3	2	1	0
ASIMn (n = 0, 1)	TXEn	RXEn	PS1n	PS0n	CLn	SLn	ISRMn	0

TXEn	RXEn	Operation mode	RXDn/Pxx pin function	TXDn/Pxx pin function
0	0	Operation stop	Port function	Port function

Cautions 1. Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

2. Be sure to set bit 0 to 0.

(2) Asynchronous serial interface (UART) mode

This mode enables full-duplex operation in which one byte of data is transmitted and received after the start bit.

The on-chip dedicated UARTn baud rate generator enables communications using a wide range of selectable baud rates.

The UARTn baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(a) Register settings

UART mode settings are made via asynchronous serial interface mode register n (ASIMn), asynchronous serial interface status register n (ASISn), baud rate generator control register n (BRGCn), baud rate generator mode control registers n and 01 (BRGMCn, BRGMC01) (n = 0, 1).

★ **Figure 10-25. ASIMn Setting (UART Mode)**

After reset: 00H R/W Address: FFFFF300H, FFFFF310H

	7	6	5	4	3	2	1	0
ASIMn (n = 0, 1)	TXEn	RXEn	PS1n	PS0n	CLn	SLn	ISRMn	0

TXEn	RXEn	Operation mode	RXDn/Pxx pin function	TXDn/Pxx pin function
0	1	UART mode (receive only)	Serial function	Port function
1	0	UART mode (transmit only)	Port function	Serial function
1	1	UART mode (transmit and receive)	Serial function	Serial function

PS1n	PS0n	Parity bit specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

CLn	Character length specification
0	7 bits
1	8 bits

SLn	Stop bit length specification for transmit data
0	1 bit
1	2 bits

ISRMn	Receive completion interrupt control when error occurs
0	Receive completion interrupt is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

Cautions 1. Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

2. Be sure to set bit 0 to 0.

★ **Figure 10-26. ASISn Setting (UART Mode)**

After reset: 00H R Address: FFFFF302H, FFFFF312H

	7	6	5	4	3	2	1	0
ASISn (n = 0, 1)	0	0	0	0	0	PEn	FEn	OVEn

PEn	Parity error flag
0	No parity error
1	Parity error (Transmit data parity does not match)

FEn	Framing error flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVEn	Overrun error flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if the stop bit length has been set to two bits by setting bit 2 (SLn) of asynchronous serial interface mode register n (ASIMn), stop bit detection during a receive operation only applies to a stop bit length of 1 bit.

2. Be sure to read the contents of receive buffer register n (RXBn) when an overrun error has occurred.

Until the contents of the RXBn register are read, further overrun errors will occur when receiving data.

★ **Figure 10-27. BRGCn Setting (UART Mode)**

After reset: 00H R/W Address: FFFFF304H, FFFFF314H

	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	Input clock selection	k
0	0	0	0	0	×	×	×	Setting prohibited	–
0	0	0	0	1	0	0	0	f_{SCK}/8	8
0	0	0	0	1	0	0	1	f_{SCK}/9	9
0	0	0	0	1	0	1	0	f_{SCK}/10	10
0	0	0	0	1	0	1	1	f_{SCK}/11	11
0	0	0	0	1	1	0	0	f_{SCK}/12	12
0	0	0	0	1	1	0	1	f_{SCK}/13	13
0	0	0	0	1	1	1	0	f_{SCK}/14	14
0	0	0	0	1	1	1	1	f_{SCK}/15	15
0	0	0	1	0	0	0	0	f_{SCK}/16	16
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	f_{SCK}/255	255

Cautions 1. Reset input sets the BRGCn register to 00H.
Before starting operation, select a setting other than "Setting prohibited". Selecting a "Setting prohibited" setting in stop mode does not cause any problems.

2. If write is performed to the BRGCn register during communication processing, the output of the baud rate generator is disturbed and communication will not be performed normally. Therefore, do not write to BRGCn during communication processing.

Remark f_{SCK}: Source clock of 8-bit counter

★ **Figure 10-28. BRGMC0 and BRGMC01 Settings (UART Mode)**

After reset: 00H R/W Address: FFFFF30EH

	7	6	5	4	3	2	1	0
BRGMC0	0	0	0	0	0	TPS02	TPS01	TPS00

After reset: 00H R/W Address: FFFFF320H

	7	6	5	4	3	2	1	0
BRGMC01	0	0	0	0	0	0	0	TPS03

TPS03	TPS02	TPS01	TPS00	8-bit counter source clock selection	m
0	0	0	0	External clock (ASCK0)	–
0	0	0	1	f_{XX}	0
0	0	1	0	$f_{XX}/2$	1
0	0	1	1	$f_{XX}/4$	2
0	1	0	0	$f_{XX}/8$	3
0	1	0	1	$f_{XX}/16$	4
0	1	1	0	$f_{XX}/32$	5
0	1	1	1	TM3 output	–
1	0	0	0	$f_{XX}/64$	6
1	0	0	1	$f_{XX}/128$	7
1	0	1	0	$f_{XX}/256$	8
1	0	1	1	$f_{XX}/512$	9
1	1	0	0	Setting prohibited	–
1	1	0	1		–
1	1	1	0		–
1	1	1	1		–

Cautions 1. If write is performed to the BRGMC0 and BRGMC01 registers during communication processing, the output of the baud rate generator is disturbed and communication will not be performed normally. Therefore, do not write to the BRGMC0 and BRGMC01 registers during communication processing.

2. Be sure to set bits 3 to7 of the BRGMC0 register to 0.

Remarks 1. f_{XX}: Main clock oscillation frequency

2. When the timer output is selected as the clock, it is not necessary to set the P27/TO3/TI3 pin to timer output mode.

★ **Figure 10-29. BRGMC1 Settings (UART Mode)**

After reset: 00H R/W Address: FFFFF31EH

	7	6	5	4	3	2	1	0
BRGMC1	0	0	0	0	0	TPS12	TPS11	TPS10

TPS12	TPS11	TPS10	8-bit counter source clock selection	m
0	0	0	External clock (ASCK1)	–
0	0	1	f_{XX}	0
0	1	0	$f_{XX}/2$	1
0	1	1	$f_{XX}/4$	2
1	0	0	$f_{XX}/8$	3
1	0	1	$f_{XX}/16$	4
1	1	0	$f_{XX}/32$	5
1	1	1	TM2 output	–

Cautions 1. If write is performed to the BRGMC1 register during communication processing, the output of the baud rate generator is disturbed and communication will not be performed normally. Therefore, do not write to the BRGMC1 register during communication processing.

2. Be sure to set bits 3 to 7 of to 0.

Remarks 1. f_{XX}: Main clock oscillation frequency

2. When the timer output is selected as the clock, it is not necessary to set the P26/TO2/TI2 pin to timer output mode.

(b) Baud rate

The transmit/receive clock for the baud rate to be generated is a signal generated by dividing the main clock.

- **Generation of baud rate transmit/receive clock using main clock**

  The transmit/receive clock is obtained by dividing the main clock. The following equation is used to obtain the baud rate from the main clock.

  <When 8 ≤ k ≤ 255>

  $$[\text{Baud rate}] = \frac{f_{XX}}{2^{m+1} \times k} \quad [\text{Hz}]$$

  f_{XX}: Main clock oscillation frequency
  m: Value set by TPS03 to TPS00 ($0 \leq m \leq 9$) – when UART0
  Value set by TPS12 to TPS10 ($0 \leq m \leq 5$) – when UART1
  k: Value set by MDLn7 to MDLn0 ($8 \leq k \leq 255$)

- **Baud rate error tolerance**

  The baud rate error tolerance depends on the number of bits in a frame and the counter division ratio [1/(16+k)].

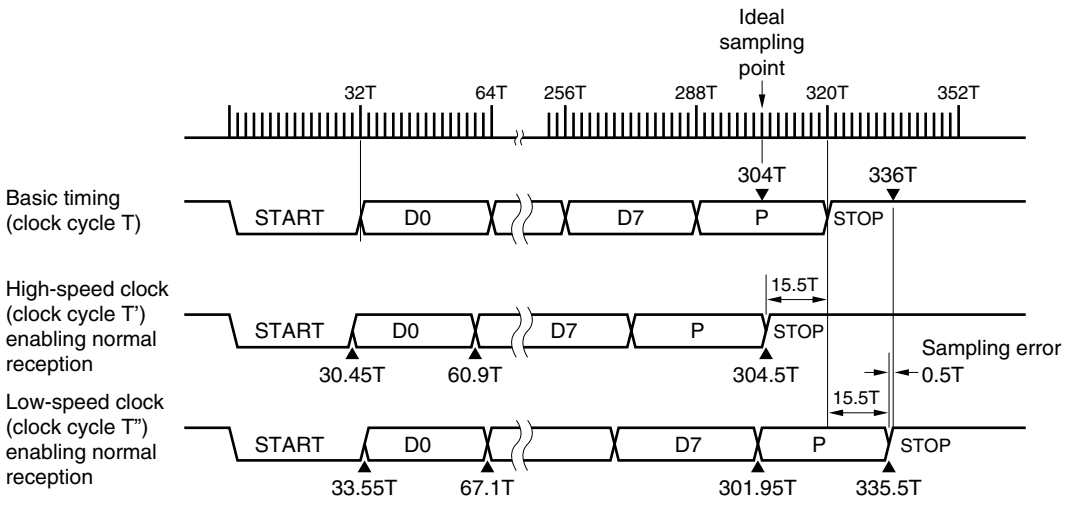
  Table 10-8 shows the relationship between the main clock and the baud rate, and Figure 10-30 shows an example of the baud rate error tolerance.

Table 10-8. Relationship Between Main Clock and Baud Rate

Baud Rate (bps)	fxx = 2 MHz			fxx = 4.194 MHz			fxx = 8.388 MHz			fxx = 17 MHz			fxx = 20 MHz		
	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)
	255	9	–4.26	—	—	—	—	—	—	—	—	—	—	—	—
16	244	8	0.06	255	9	0.39	—	—	—	—	—	—	—	—	—
32	244	7	0.06	255	8	0.38	255	9	0.38	—	—	—	—	—	—
75	208	6	0.16	218	7	0.20	218	8	0.20	221	9	0.16	—	—	—
76	206	6	–0.20	216	7	–0.20	216	8	–0.20	218	9	0.20	255	9	0.78
256	244	4	0.06	128	6	0.01	128	7	0.01	130	8	–0.23	152	8	0.39
1200	208	2	0.16	217	3	0.20	217	4	0.20	221	5	0.16	130	6	0.16
2400	208	1	0.16	218	2	0.20	218	3	0.20	221	4	0.16	130	5	0.16
4800	208	0	0.16	218	1	0.20	218	2	0.20	221	3	0.16	130	4	0.16
9600	104	0	0.16	218	0	0.20	218	1	0.20	221	2	0.16	130	3	0.16
19200	52	0	0.16	109	0	0.20	218	0	0.20	221	1	0.16	130	2	0.16
31250	32	0	0.00	67	0	0.16	134	0	0.16	136	1	0.00	160	1	0.00
38400	26	0	0.16	55	0	–0.71	110	0	–0.71	221	0	0.16	130	1	0.16
76800	13	0	0.16	27	0	1.13	54	0	1.13	111	0	–0.29	130	0	0.16
125000	8	0	0.00	17	0	–1.32	34	0	–1.32	68	0	0.00	80	0	0.00
150000	—	—	—	14	0	0.14	28	0	0.14	55	0	0.62	67	0	–0.50
262000	—	—	—	8	0	0.05	16	0	0.05	32	0	1.38	38	0	0.44
300000	—	—	—	—	—	—	14	0	–0.14	28	0	–1.18	33	0	1.01
524000	—	—	—	—	—	—	8	0	0.05	16	0	1.38	19	0	0.44
1250000	—	—	—	—	—	—	—	—	—	7	0	–2.86	8	0	0.00

Remark fxx: Main clock oscillation frequency

Figure 10-30. Error Tolerance (When k = 16), Including Sampling Errors



Remark T: 8-bit counter's source clock cycle

$$\text{Baud rate error tolerance (when k = 16)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

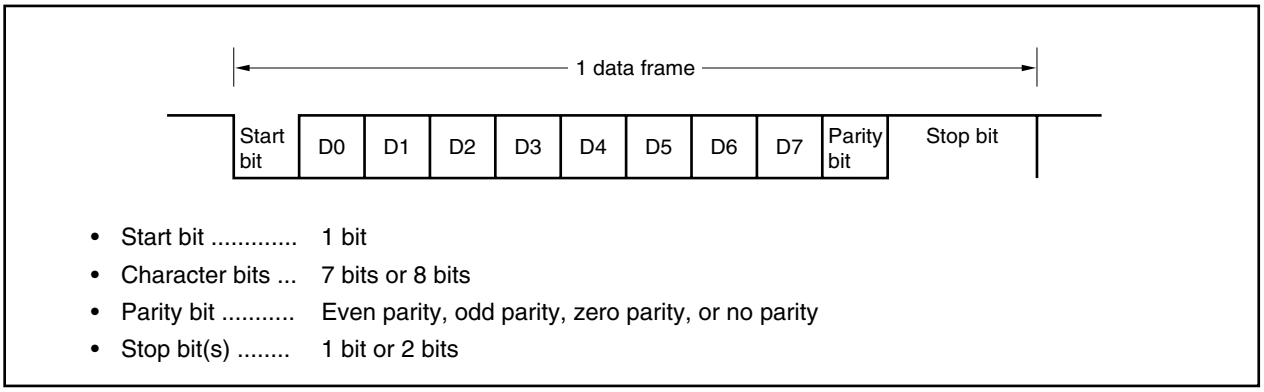
(c) Communication operations

(i) Data format

As shown in Figure 10-31, the format of the transmit/receive data consists of a start bit, character bits, a parity bit, and one or more stop bits.

Asynchronous serial interface mode register n (ASIMn) is used to set the character bit length, parity selection, and stop bit length within each data frame (n = 0, 1).

Figure 10-31. Format of Transmit/Receive Data in Asynchronous Serial Interface



- Start bit ..... 1 bit
- Character bits ... 7 bits or 8 bits
- Parity bit ..... Even parity, odd parity, zero parity, or no parity
- Stop bit(s) ..... 1 bit or 2 bits

When 7 bits is selected as the number of character bits, only the lower 7 bits (from bit 0 to bit 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to 0.

Asynchronous serial interface mode register n (ASIMn) and baud rate generator control register n (BRGCn) are used to set the serial transfer rate (n = 0, 1).

If a receive error occurs, information about the receive error can be ascertained by reading asynchronous serial interface status register n (ASISn) (n = 0, 1).

(ii) Parity types and operations

The parity bit is used to detect bit errors in transfer data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

<1> Even parity

- **During transmission**

  The number of bits in transmit data including the parity bit is controlled so that an even number of "1" bits is set. The value of the parity bit is as follows.

  If the transmit data contains an odd number of "1" bits: The parity bit value is "1"
  If the transmit data contains an even number of "1" bits: The parity bit value is "0"

- **During reception**

  The number of "1" bits is counted among the receive data including a parity bit, and a parity error occurs when the result is an odd number.

<2> Odd parity

- **During transmission**

  The number of bits in transmit data including a parity bit is controlled so that an odd number of "1" bits is set. The value of the parity bit is as follows.

  If the transmit data contains an odd number of "1" bits: The parity bit value is "0"
  If the transmit data contains an even number of "1" bits: The parity bit value is "1"

- **During reception**

  The number of "1" bits is counted among the receive data including a parity bit, and a parity error occurs when the result is an even number.

<3> Zero parity

During transmission, the parity bit is set to "0" regardless of the transmit data.
During reception, the parity bit is not checked. Therefore, no parity errors will occur regardless of whether the parity bit is a "0" or a "1".

<4> No parity

No parity bit is added to the transmit data.
During reception, receive data is regarded as having no parity bit. Since there is no parity bit, no parity errors will occur.

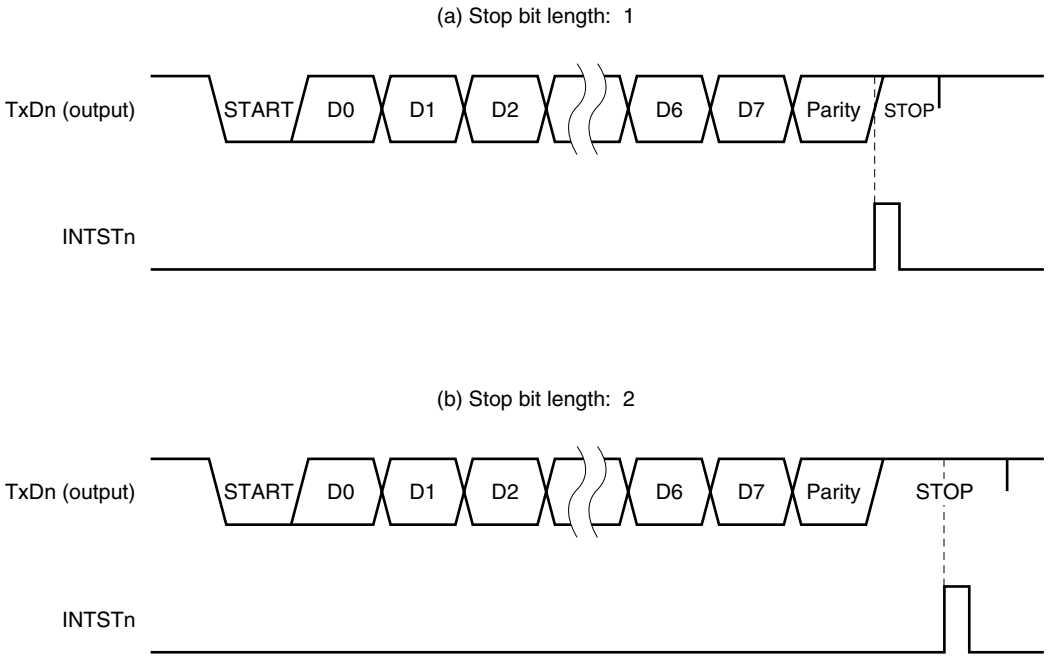
(iii) Transmission

The transmit operation is started when transmit data is written to transmit shift register n (TXSn). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXSn, thereby emptying TXSn, after which a transmit completion interrupt (INTSTn) is issued.

The timing of the transmit completion interrupt is shown below.

Figure 10-32. Timing of Asynchronous Serial Interface Transmit Completion Interrupt



Caution Do not write to asynchronous serial interface mode register n (ASIMn) during a transmit operation. Writing to ASIMn during a transmit operation may disable further transmit operations (in such cases, input $\overline{\text{RESET}}$ to restore normal operation).

Whether or not a transmit operation is in progress can be determined via software using the transmit completion interrupt (INTSTn) or the interrupt request flag (STIFn) that is set by INTSTn.

Remark n = 0, 1

(iv) Reception

The receive operation is enabled when "1" is set to bit 6 (RXEn) of asynchronous serial interface mode register n (ASIMn), and the input via the RXDn pin is sampled.

The serial clock specified by baud rate generator control register n (BRGCn) is used when sampling the RXDn pin.

When the RXDn pin goes low, the 5-bit counter begins counting and the start timing signal for data sampling is output when half of the specified baud rate time has elapsed. If sampling the RXDn pin input with this start timing signal yields a low-level result, a start bit is recognized, after which the 5-bit counter is initialized and starts counting and data sampling begins. After the start bit is recognized, the character data, parity bit, and one-bit stop bit are detected, at which point reception of one data frame is completed.

Once reception of one data frame is completed, the receive data in the shift register is transferred to receive buffer register n (RXBn) and a receive completion interrupt (INTSRn) occurs.

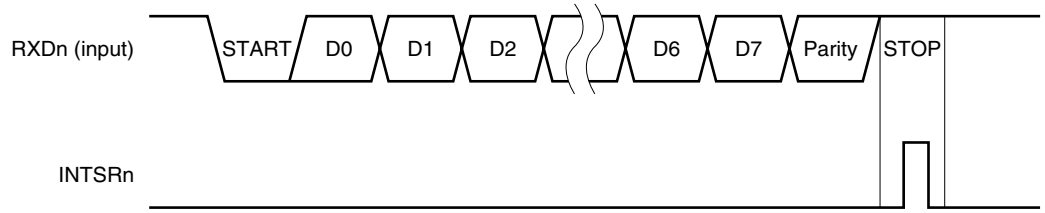
Even if an error has occurred, the receive data in which the error occurred is still transferred to RXBn.

When an error occurs, INSTRn is generated if bit 1 (ISRMn) of ASIMn is cleared (0). On the other hand, INTSRn is not generated if the ISRMn bit is set (1) (see **10.4.2 (1) Asynchronous serial interface mode registers 0 and 1 (ASIM0, ASIM1**).

If the RXEn bit is reset to 0 during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXBn and ASISn do not change, nor does INTSRn or INTSERn occur.

The timing of the asynchronous serial interface receive completion interrupt is shown below.

Figure 10-33. Timing of Asynchronous Serial Interface Receive Completion Interrupt



Caution Be sure to read the contents of receive buffer register n (RXBn) even when a receive error has occurred. If the contents of RXBn are not read, an overrun error will occur during the next data receive operation and the receive error status will remain.

Remarks 1. n = 0, 1

2. The interrupt control register of INTSR0 is alternately used as the interrupt control register (CSIC1) of INTCSI1. An SRIC0 register does not exist.

(v) Receive error

Three types of errors can occur during a receive operation: a parity error, framing error, and overrun error. When, as the result of data reception, an error flag is set in asynchronous serial interface status register n (ASISn), the receive error interrupt request (INTSERn) is generated. The receive error interrupt request is generated prior to the receive completion interrupt request (INTSRn). Table 10-9 shows receive error causes.

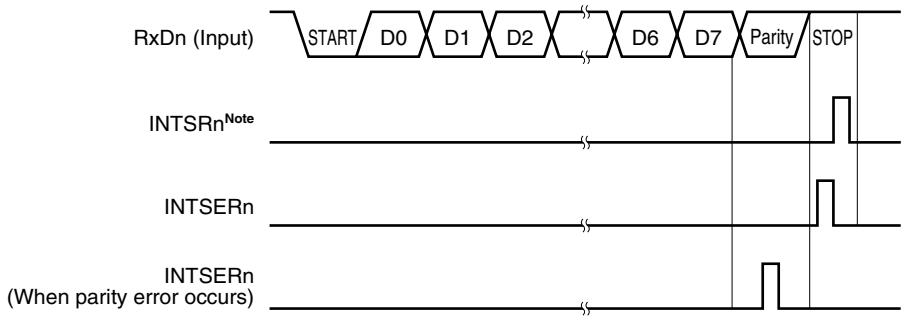
By reading the contents of ASISn during receive error interrupt servicing (INTSERn), it is possible to ascertain which error has occurred during reception (see **Table 10-9** and **10.4.2 (2) Asynchronous serial interface status registers 0 and 1 (ASIS0, ASIS1)**

The contents of ASISn are reset (0) by reading the receive buffer register (RXBn) or receiving subsequent data (if there is an error in the subsequent data, the error flag is set).

Table 10-9. Receive Error Causes

Receive error	Cause	ASISn value
Parity error	Parity specification at transmission and receive data parity do not match.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of subsequent data was completed before data was read from the receive buffer register.	01H

Figure 10-34. Receive Error Timing



Note Even if a receive error occurs when the ISRMn bit of ASIMn is set (1), INTSRn is not generated.

Cautions 1. The contents of asynchronous serial interface status register n (ASISn) are reset (0) by reading receive buffer register N (RXBn) or receiving subsequent data. To check the contents of an error, always read ASISn before reading RXBn.

2. Be sure to read receive buffer register n (RXBn) even when a receive error occurs. If RXBn is not read out, an overrun error will occur during subsequent data reception and as a result receive errors will continue to occur.

Remark n = 0, 1

10.4.4 Standby function

(1) Operation in HALT mode

Only serial transfer operations are performed normally.

(2) Operation in IDLE and software STOP modes

(a) When internal clock is selected as serial clock

The operations of asynchronous serial interface mode register n (ASIMn), asynchronous serial status register n (ASISn), baud rate generator control register n (BRGCn), baud rage generator mode control registers n and 01 (BRGMCn, BRGMC01), transmit shift register n (TXSn), and receive buffer register n (RXBn) are stopped and their values immediately before the clock stopped are held.

The TXDn pin output holds the data immediately before the clock was stopped (in software STOP mode) during transmission. When the clock is stopped during reception, the receive data until the clock stopped is stored and subsequent receive operations are stopped. Reception resumes upon clock restart.

(b) When external clock is selected as serial clock

Only serial transfer operations are performed normally.

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, and can handle 12 channels of analog input signals (ANI0 to ANI11).

(1) Hardware start

Conversion is started by trigger input (ADTRG) (rising edge, falling edge, or both rising and falling edges can be specified).

(2) Software start

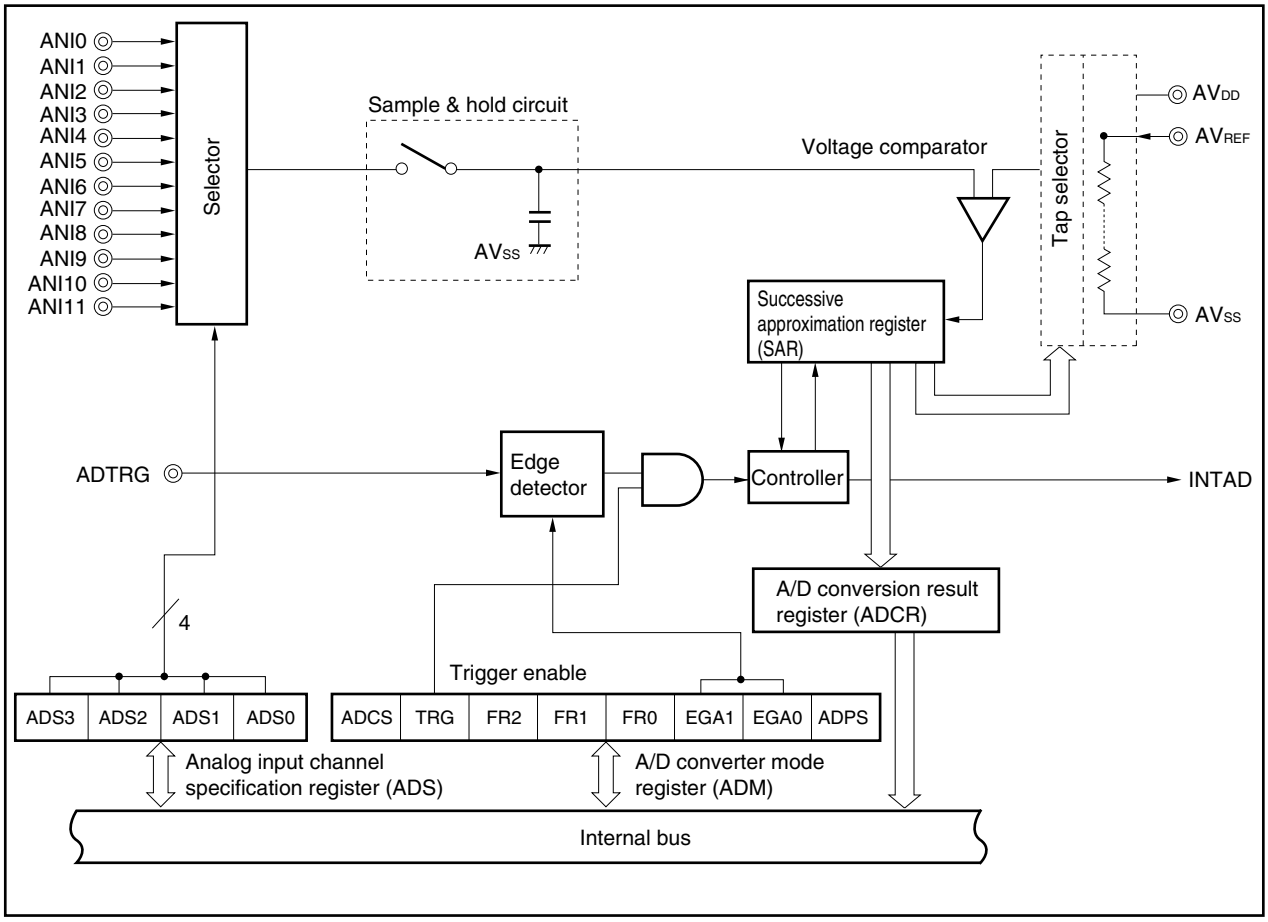
Conversion is started by setting the A/D converter mode register (ADM).

One analog input channel is selected from ANI0 to ANI11, and A/D conversion is performed. If A/D conversion has been started by means of hardware start, conversion stops once it has been completed, and an interrupt request (INTAD) is generated. If conversion has been started by means of software start, conversion is performed repeatedly. Each time conversion has been completed, INTAD is generated.

Operation of the A/D converter continues in HALT mode.

The block diagram is shown below.

Figure 11-1. Block Diagram of A/D Converter



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)
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
Control registers	A/D converter mode register (ADM) Analog input channel specification register (ADS)

(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 result of the comparison starting from the most significant bit (MSB). When the comparison result has been stored down to the least significant bit (LSB) (i.e., when the A/D conversion has been completed), 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 H (ADCRH)

Each time A/D conversion is completed, the result of the conversion is loaded to this register from the successive approximation register. The higher 10 bits of this register hold the result of the A/D conversion (the lower 6 bits are fixed to 0). This register is read using a 16-bit memory manipulation instruction. $\overline{\text{RESET}}$ input sets ADCR to 0000H.

When using only the higher 8 bits of the result of the A/D conversion, ADCRH is read using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADCRH to 00H.

(3) 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 also holds the sampled analog input signal voltage during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input signal with the output voltage of the series resistor string.

(5) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(6) ANI0 to ANI11 pins

These are analog input pins for the 12 channels of the A/D converter, and are used to input analog signals to be converted into digital signals. Pins other than ones selected as the analog input by the analog input channel specification register (ADS) can be used as input ports.

Caution Make sure that the voltages input to ANI0 through ANI11 do not exceed the rated values. If a voltage higher than or equal to AV_{REF} or lower than or equal to AV_{SS} (even within the range of 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.

(7) AV_{REF} pin

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

The signals input to the ANI0 through ANI11 pins are converted into digital signals based on the voltage applied across AV_{REF} and AV_{SS}.

(8) AV_{SS} pin

This is the ground pin of the A/D converter. Always keep the potential at this pin the same as that at the V_{SS} pin even when the A/D converter is not in use.

(9) AV_{DD} pin

This is the analog power supply pin of the A/D converter. Always keep the potential at this pin the same as that at the V_{DD} pin even when the A/D converter is not in use.

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)

(1) A/D converter mode register (ADM)

This register specifies the conversion time of the input analog signal to be converted into a digital signal, starting or stopping the conversion, and an external trigger.

ADM is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADM to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF3C0H

	7	6	5	4	3	2	1	0
ADM	ADCS	TRG	FR2	FR1	FR0	EGA1	EGA0	ADPS

ADCS	A/D Conversion Control
0	Conversion stopped
1	Conversion enabled

TRG	Software Start or Hardware Start Selection
0	Software start
1	Hardware start

(2/2)

★

ADPS	FR2	FR1	FR0	Selection of Conversion Time[Note 1] + Stabilization time[Note 2]
0	0	0	0	288/fxx
0	0	0	1	216/fxx
0	0	1	0	168/fxx
0	0	1	1	120/fxx
0	1	0	0	96/fxx
0	1	0	1	72/fxx
0	1	1	0	60/fxx
0	1	1	1	48/fxx
1	0	0	0	288/fxx + 144/fxx
1	0	0	1	216/fxx + 108/fxx
1	0	1	0	168/fxx + 84/fxx
1	0	1	1	120/fxx + 60/fxx
1	1	0	0	96/fxx + 48/fxx
1	1	0	1	72/fxx + 36/fxx
1	1	1	0	60/fxx + 30/fxx
1	1	1	1	48/fxx + 24/fxx

EGA1	EGA0	Edge Specification for External Trigger Signal
0	0	No edge detection
0	1	Detection at falling edge
1	0	Detection at rising edge
1	1	Detection at both rising and falling edges

ADPS	A/D Conversion Time Mode Selection
0	Comparator on
1	Comparator off

Notes 1. Conversion time (actual A/D conversion time)
Always set the time to 5 μs ≤ Conversion time ≤ 100 μs. However, when ADPS bit = 1, the oscillation stabilization time is not included.

2. Stabilization time (setup time of A/D converter)
Each A/D conversion requires “conversion time + stabilization time”. There is no stabilization time when ADPS = 0.

Remark Turning off the internal comparator cuts the current flowing through the AV_{DD} pin.

★ **Table 11-2. A/D Conversion Time Selection**

ADPS	FR2	FR1	FR0	Conversion Time Selection					
				Conversion Time + Stabilization Time	f_{XX}				
					20 MHz	17 MHz	13.5 MHz	8 MHz	2 MHz
0	0	0	0	288/f_{XX}	14.4 μs	16.9 μs	21.3 μs	36.0 μs	Setting prohibited
0	0	0	1	216/f_{XX}	10.8 μs	12.7 μs	16.0 μs	27.0 μs	Setting prohibited
0	0	1	0	168/f_{XX}	8.4 μs	9.9 μs	12.4 μs	21.0 μs	84.0 μs
0	0	1	1	120/f_{XX}	6.0 μs	7.1 μs	8.9 μs	15.0 μs	60.0 μs
0	1	0	0	96/f_{XX}	Setting prohibited	5.6 μs	7.1 μs	12.0 μs	48.0 μs
0	1	0	1	72/f_{XX}	Setting prohibited	Setting prohibited	5.3 μs	9.0 μs	36.0 μs
0	1	1	0	60/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	7.5 μs	30.0 μs
0	1	1	1	48/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	6.0 μs	24.0 μs
1	0	0	0	288/f_{XX}+144/f_{XX}	21.6 μs	25.4 μs	32.0 μs	54.0 μs	Setting prohibited
1	0	0	1	216/f_{XX}+108/f_{XX}	16.2 μs	19.1 μs	24.0 μs	40.5 μs	Setting prohibited
1	0	1	0	168/f_{XX}+84/f_{XX}	12.6 μs	14.9 μs	18.7 μs	31.5 μs	Setting prohibited
1	0	1	1	120/f_{XX}+60/f_{XX}	9.0 μs	10.7 μs	13.3 μs	22.5 μs	90.0 μs
1	1	0	0	96/f_{XX}+48/f_{XX}	Setting prohibited	8.4 μs	10.7 μs	18.0 μs	72.0 μs
1	1	0	1	72/f_{XX}+36/f_{XX}	Setting prohibited	Setting prohibited	8.0 μs	13.5 μs	54.0 μs
1	1	1	0	60/f_{XX}+30/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	11.3 μs	45.0 μs
1	1	1	1	48/f_{XX}+24/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	9.0 μs	36.0 μs

(2) Analog input channel specification register (ADS)

ADS specifies the port for inputting the analog voltage to be converted into a digital signal.

ADS is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ADS to 00H.

After reset: 00H R/W Address: FFFFF3C2H

	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Analog Input Channel Specification
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
Other than above				Setting prohibited

Caution Be sure to set bits 4 to 7 to 0.

11.4 Operation

11.4.1 Basic operation

<1> Select one channel whose analog signal is to be converted into a digital signal by using the analog input channel specification register (ADS).

<2> The sample & hold circuit samples the voltage input to the selected analog input channel.

<3> After sampling for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until it has been converted into a digital signal.

<4> Set bit 9 of the successive approximation register (SAR). The tap selector sets the voltage tap of the series resistor string to (1/2) AV_{REF}.

<5> The voltage difference between the voltage tap of the series resistor string and the analog input voltage is compared by the voltage comparator. If the analog input voltage is greater than (1/2) AV_{REF}, the MSB of the SAR remains set. If the analog input voltage is less than (1/2) AV_{REF}, the MSB is reset.

<6> Next, bit 8 of the SAR is automatically set, and the analog input voltage is compared again. Depending on the value of bit 9 to which the result of the preceding comparison has been set, the voltage tap of the series resistor string is selected as follows:

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The analog input voltage is compared with one of these voltage taps, and bit 8 of the SAR is manipulated as follows depending on the result of the comparison.

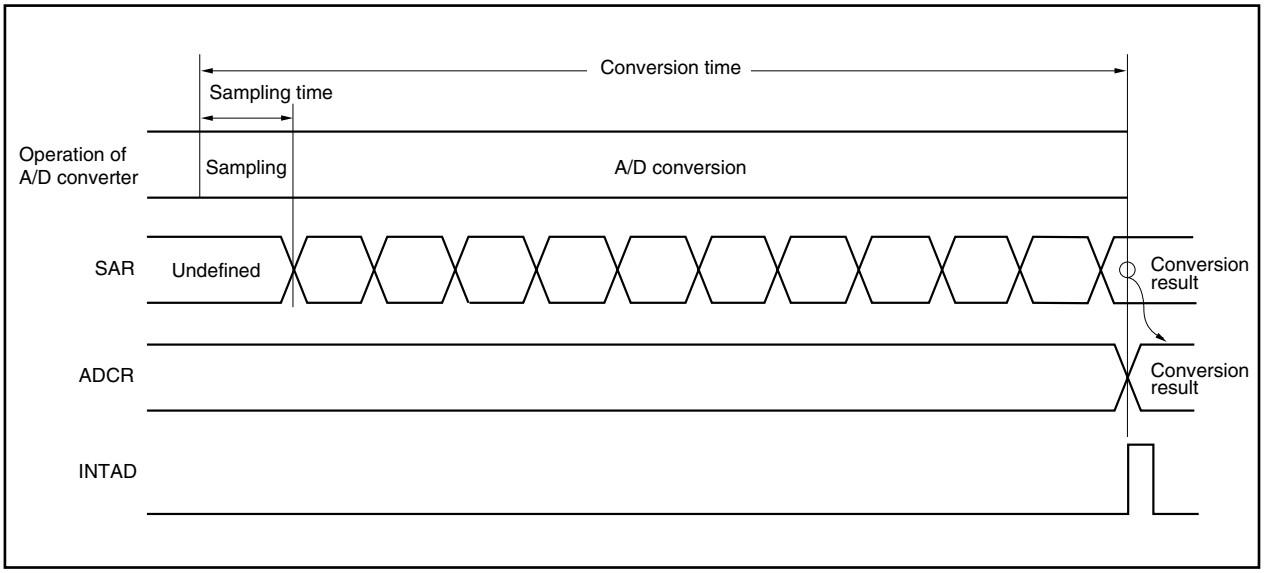
- Analog input voltage ≥ voltage tap: Bit 8 = 1
- Analog input voltage ≤ voltage tap: Bit 8 = 0

<7> The above steps are repeated until bit 0 of the SAR has been manipulated.

<8> When comparison of all 10 bits of the SAR has been completed, the valid digital value remains in the SAR, and the value of the SAR is transferred and latched to the A/D conversion result register (ADCR). At the same time, an A/D conversion end interrupt request (INTAD) can be generated.

Caution The first conversion value immediately after setting ADCS = 0 → 1 may not satisfy the ratings.

Figure 11-2. Basic Operation of A/D Converter



A/D conversion is successively executed until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset to 0 by software.

If ADM and the analog input channel specification register (ADS) are written during A/D conversion, the conversion is initialized. If ADCS is set to 1 at this time, conversion is started from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register (ADCR) to 0000H.

11.4.2 Input voltage and conversion result

The analog voltages input to the analog input pins (ANI0 to ANI11) and the result of the A/D conversion (contents of the A/D conversion result register (ADCR)) are related as follows.

$$ADCR = INT(\frac{V_{IN}}{AV_{REF}} \times 1024 + 0.5)$$

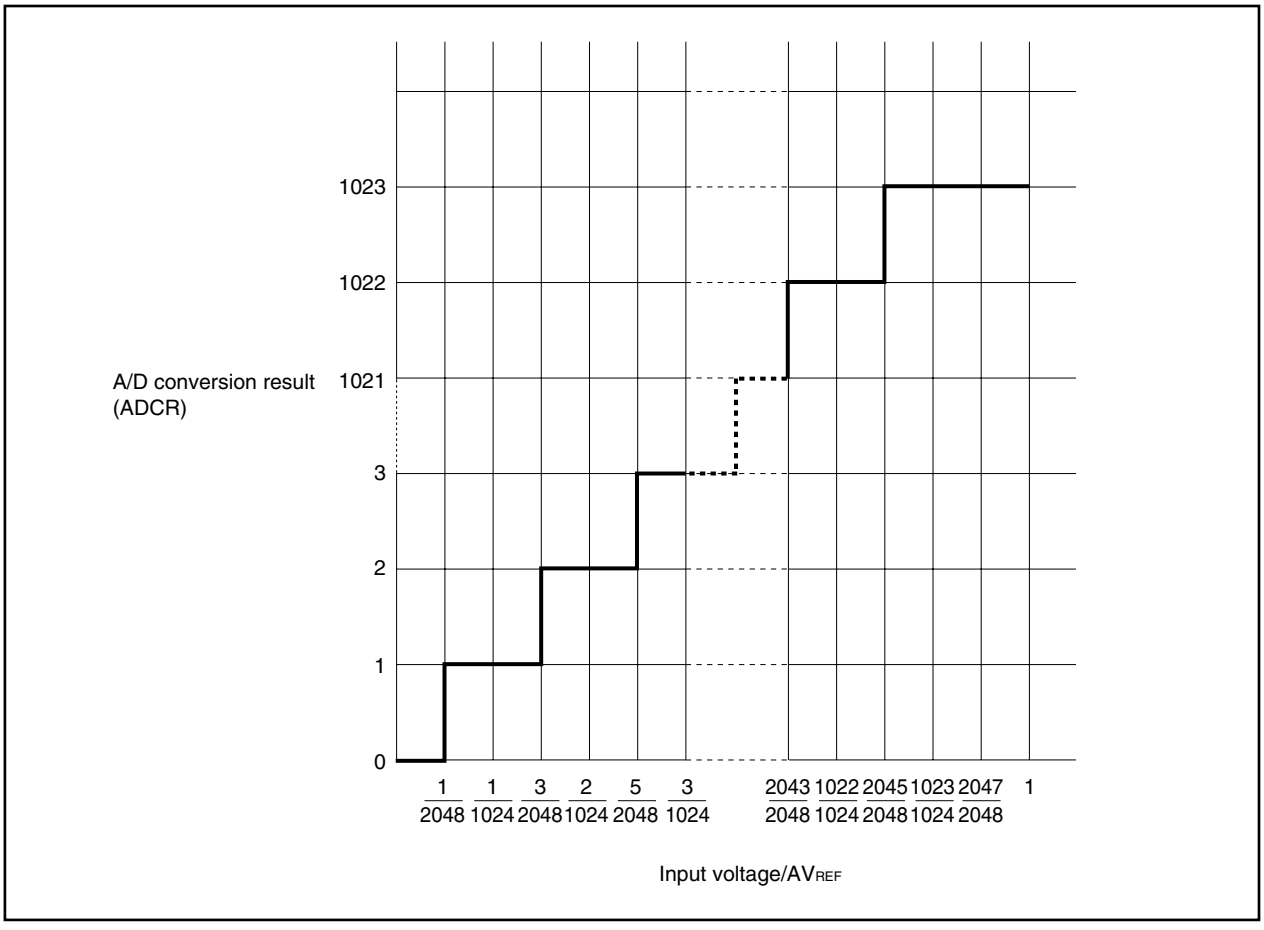
Or,

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

INT ( ): Function that returns integer of value in ( )
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: Value of the A/D conversion result register (ADCR)

The relationship between the analog input voltage and A/D conversion result is shown below.

Figure 11-3. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 A/D converter operation mode

In this mode one of the analog input channels ANI0 to ANI11 is selected by the analog input channel specification register (ADS) and A/D conversion is executed.

The A/D conversion can be started in the following two ways.

- Hardware start: Started by trigger input (ADTRG) (rising edge, falling edge, or both rising and falling edges can be specified)
- Software start: Started by setting A/D converter mode register (ADM)

The result of the A/D conversion is stored in the A/D conversion result register (ADCR) and an interrupt request signal (INTAD) is generated at the same time.

(1) A/D conversion by hardware start

A/D conversion is on standby if bit 6 (TRG) and bit 7 (ADCS) of the A/D converter mode register (ADM) are set to 1. When an external trigger signal is input, the A/D converter starts converting the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) into a digital signal.

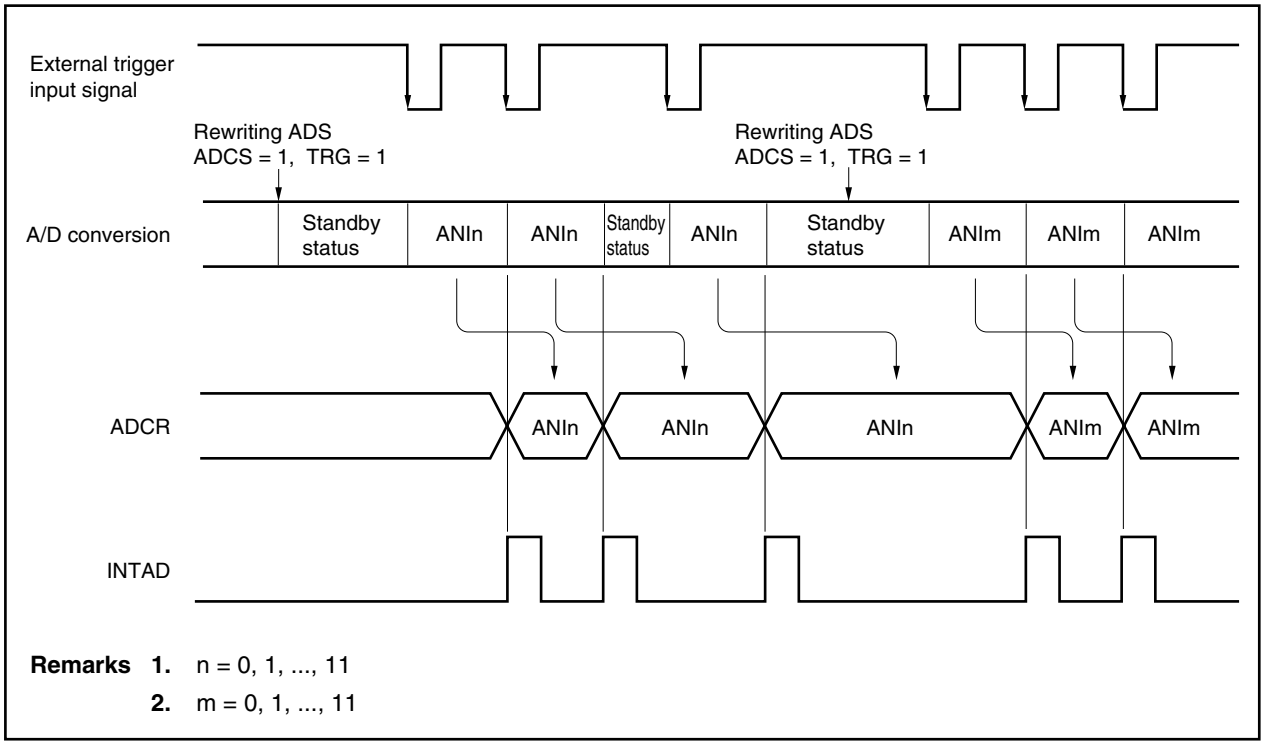
When A/D conversion has been completed, the result of the conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once in A/D conversion has been started and completed, conversion is not started again unless a new external trigger signal is input.

If data with ADCS set to 1 is written to ADM during A/D conversion, the conversion under execution is stopped, and the A/D converter stands by until a new external trigger signal is input. If the external trigger signal is input, A/D conversion is executed again from the beginning.

If data with ADCS set to 0 is written to ADM during A/D conversion, the conversion is immediately stopped.

Caution Be sure to make the input interval of the external trigger signal higher than the conversion time specified by the FR2 to FR0 bits of the ADM register + 6 CPU clocks.

Figure 11-4. A/D Conversion by Hardware Start (with Falling Edge Specified)



Remarks 1. n = 0, 1, ..., 11

2. m = 0, 1, ..., 11

(2) A/D conversion by software start

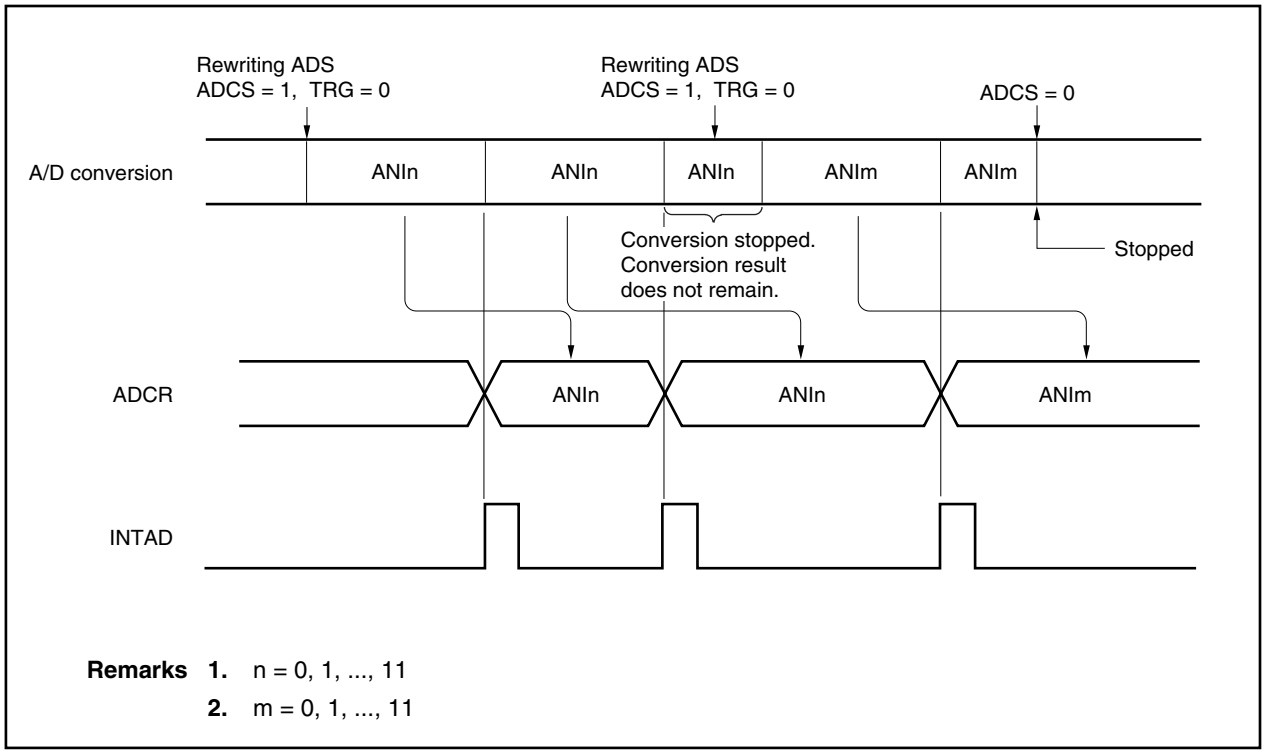
If bit 6 (TRG) of A/D converter mode register 1 (ADM1) is set to 0 and bit 7 (ADCS) is set to 1, the A/D converter starts converting the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) into a digital signal.

When A/D conversion has been completed, the result of the conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once A/D conversion has been started and completed, the next conversion is started immediately. A/D conversion is repeated until new data is written to ADS.

If ADS is rewritten during A/D conversion, the conversion under execution is stopped, and conversion of the newly selected analog input channel is started.

If data with ADCS set to 0 is written to ADM during A/D conversion, the conversion is immediately stopped.

Figure 11-5. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 11

2. m = 0, 1, ..., 11

11.5 Notes on Using A/D Converter

(1) Current consumption in standby mode

The A/D converter stops operation in the IDLE/software STOP mode (operable in the HALT mode). At this time, the current consumption of the A/D converter can be reduced by stopping the conversion (by resetting the bit 7 (ADCS) of the A/D converter mode register (ADM) to 0).

To reduce the current consumption in the IDLE/software STOP mode, set the AV_{REF} potential in the user circuit to the same value (0 V) as the AV_{SS} potential.

(2) Input range of ANI0 to ANI11

Keep the input voltage of the ANI0 through ANI11 pins to within the rated range. If a voltage greater than or equal to AV_{REF} or lower than or equal to AV_{SS} (even within the range of the absolute maximum ratings) is input to a channel, the converted value of the channel becomes undefined. Moreover, the values of the other channels may also be affected.

(3) Conflict

<1> Conflict between writing A/D conversion result register (ADCR) and reading ADCR at end of conversion

Reading ADCR takes precedence. After ADCR has been read, a new conversion result is written to ADCR.

<2> Conflict between writing ADCR and external trigger signal input at end of conversion

The external trigger signal is not input during A/D conversion. Therefore, the external trigger signal is not acknowledged during writing of ADCR.

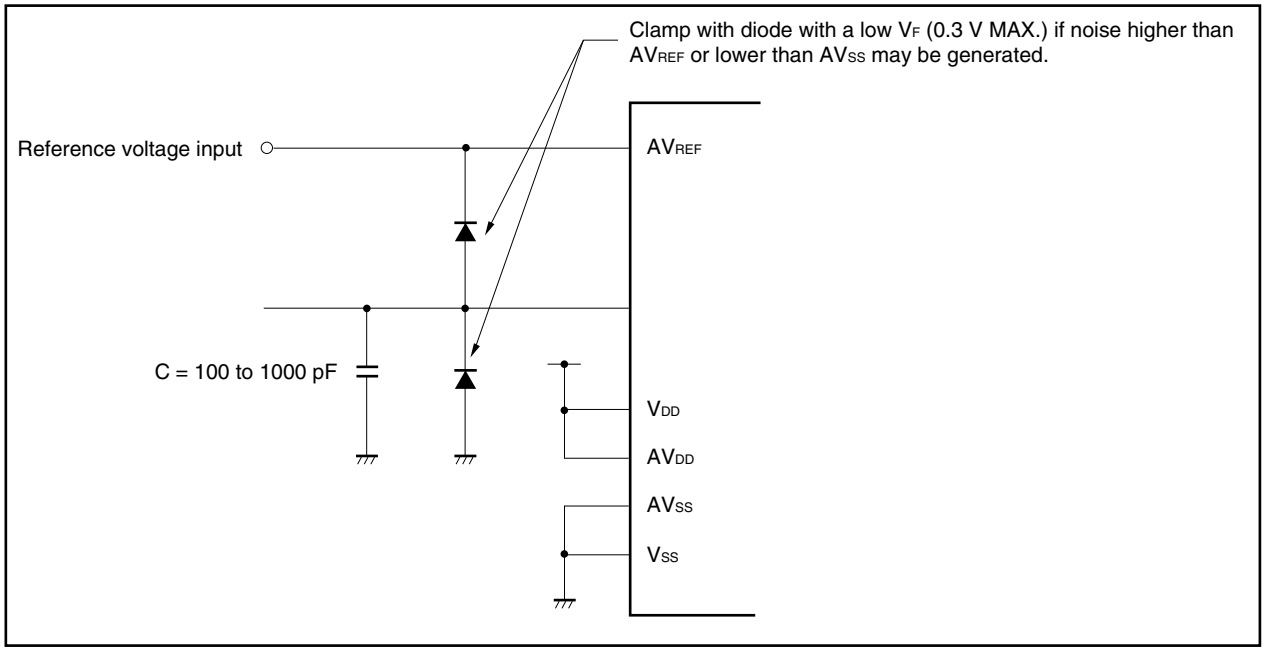
<3> Conflict between writing of ADCR and writing A/D converter mode register (ADM) or analog input channel specification register (ADS)

When ADM or ADS write is performed immediately after ADCR write following the end or A/D conversion, the conversion result is not written to the ADCR register, and INTAD is not generated.

(4) Countermeasures against noise

To keep the resolution of 10 bits, prevent noise from being superimposed on the AV_{REF} and ANI0 to ANI11 pins. The higher the output impedance of the analog input source, the heavier the influence of noise. To lower noise, connecting an external capacitor as shown in Figure 11-6 is recommended.

Figure 11-6. Handling of Analog Input Pin



(5) ANI0 to ANI11

The analog input (ANI0 to ANI11) pins function alternately as port pins.

To execute A/D conversion with any of ANI0 to ANI11 selected, do not execute an instruction that inputs data to the port during conversion; otherwise, the resolution may drop.

If a digital pulse is applied to pins adjacent to the pin whose input signal is converted into a digital signal, the expected A/D conversion result may not be obtained because of the influence of coupling noise. Therefore, do not apply a pulse to the adjacent pins.

(6) Input impedance of AV_{REF} pin

A series resistor string is connected between the AV_{REF} and AV_{SS} pins.

If the output impedance of the reference voltage source is too high, the series resistor string between the AV_{REF} and AV_{SS} pins is connected in series, increasing the error of the reference voltage.

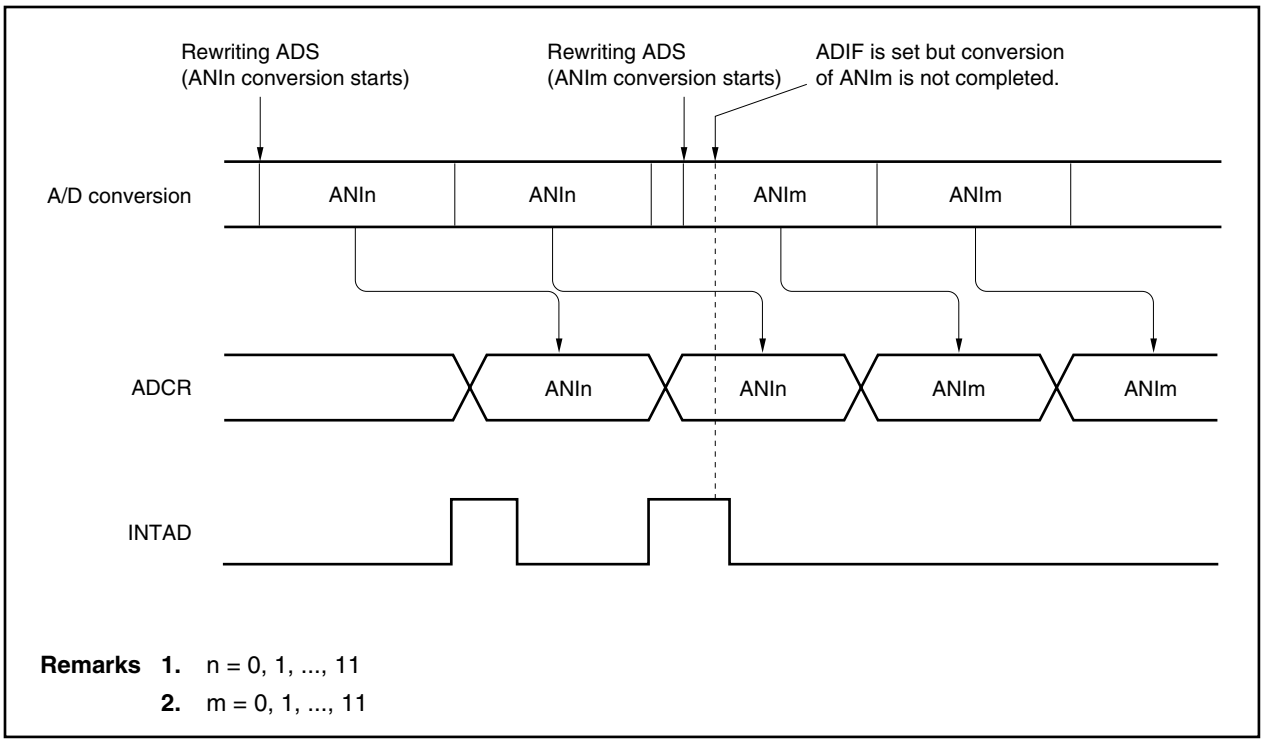
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the contents of the analog input channel specification register (ADS) are changed.

If the analog input pin is changed during conversion, therefore, the result of the A/D conversion of the preceding analog input signal and the conversion end interrupt request flag may be set immediately before ADS is rewritten. If ADIF is read immediately after ADS has been rewritten, it may be set despite the fact that conversion of the newly selected analog input signal has not been completed yet.

When stopping A/D conversion and then resuming, clear ADIF before resuming conversion.

Figure 11-7. A/D Conversion End Interrupt Request Generation Timing



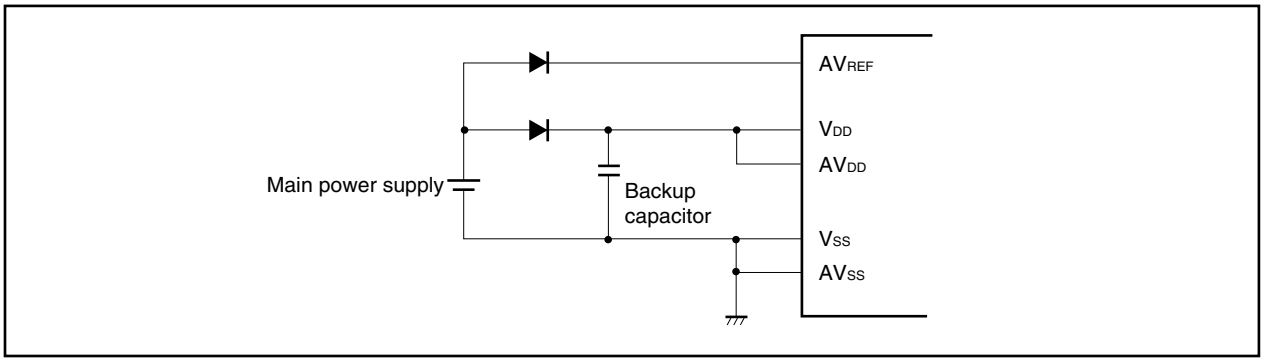
Remarks 1. n = 0, 1, ..., 11

2. m = 0, 1, ..., 11

(8) AV_{DD} pin

The AV_{DD} pin is the power supply pin of the analog circuit, and also supplies power to the input circuit of ANI0 to ANI11. Even in an application where a backup power supply is used, therefore, be sure to apply the same voltage as the V_{DD} pin to the AV_{DD} pin as shown in Figure 11-8.

Figure 11-8. Handling of AV_{DD} Pin



(9) Reading out A/D conversion result register (ADCR)

A write operation to the A/D converter mode register (ADM) and analog input channel specification register (ADS) may cause the ADCR contents to be undefined. Therefore, read ADCR during A/D conversion (ADCS bit = 1). Incorrect conversion results may be read out at a timing other than the above.

★ 11.6 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range). %FSR indicates the ratio of analog input voltage that can be converted as a percentage, and is always represented by the following formula regardless of the resolution.

1%FSR = (Max. value of analog input voltage that can be converted − Min. value of analog input voltage that can be converted)/100
= (AV_{REF} – 0)/100
= AV_{REF}/100

1LSB is as follows when the resolution is 10 bits.

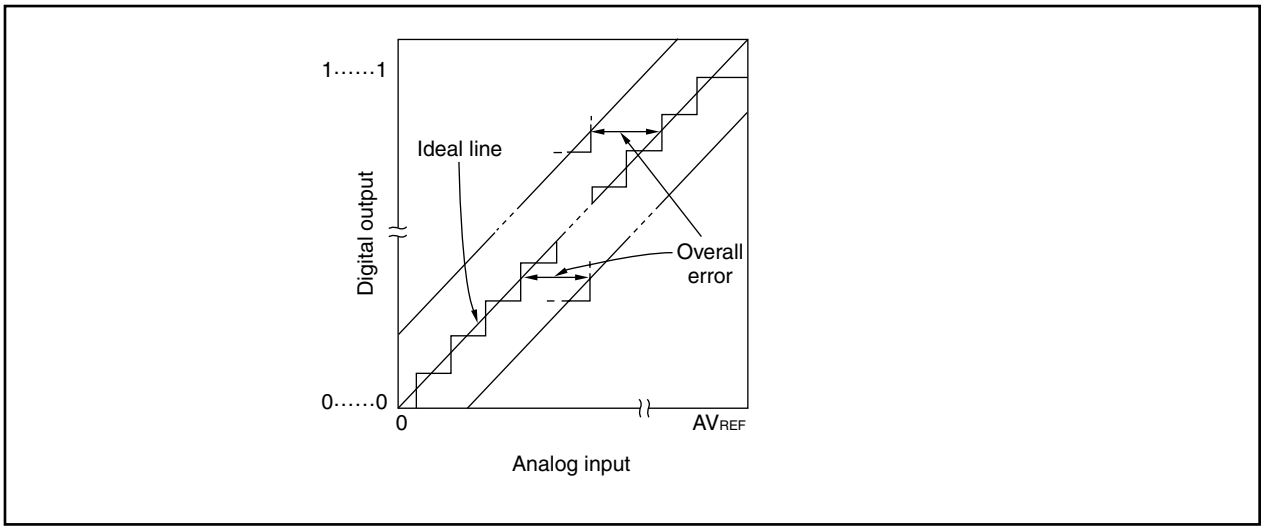
1LSB = $1/2^{10}$ = 1/1024
= 0.098%FSR

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, linearity error and errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

Figure 11-9. Overall Error

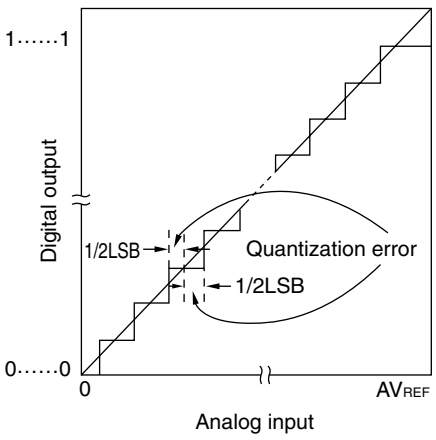


(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

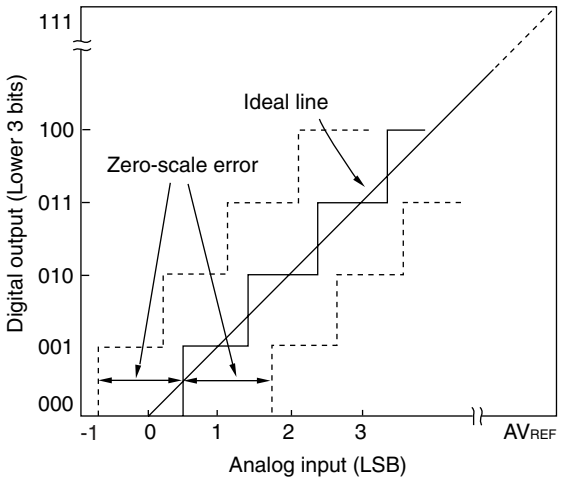
Figure 11-10. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001.

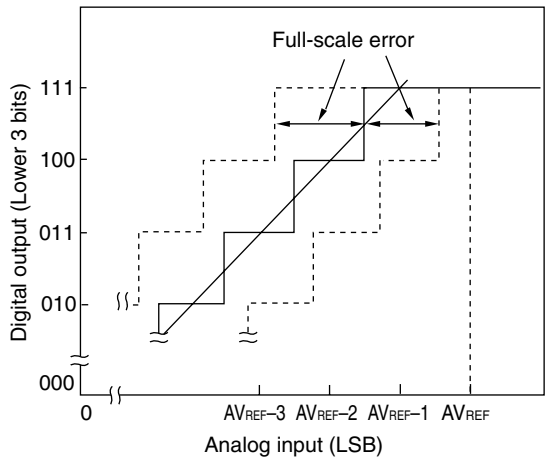
Figure 11-11. Zero-Scale Error



(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 1……110 to 1……111.

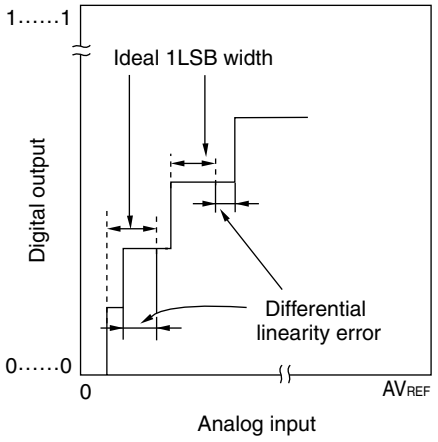
Figure 11-12. Full-Scale Error



(6) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

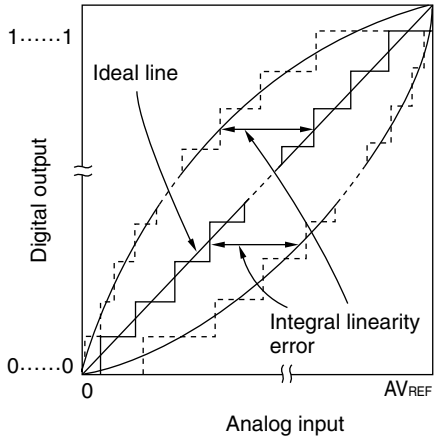
Figure 11-13. Differential Linearity Error



(7) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

Figure 11-14. Integral Linearity Error



(8) Conversion time

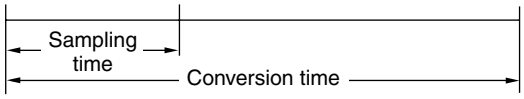
This expresses the time from when the analog input voltage was applied to the time when the digital output was obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

Figure 11-15. Sampling Time



CHAPTER 12 DMA FUNCTIONS

★ 12.1 Functions

The V850/SA1 incorporates a three channel DMA (Direct Memory Access) controller (DMAC) that controls and executes DMA transfer.

The DMAC transfers data between internal RAM and on-chip peripheral I/O based on a trigger from the on-chip peripheral I/O (serial interface, timer/counter, or A/D converter).

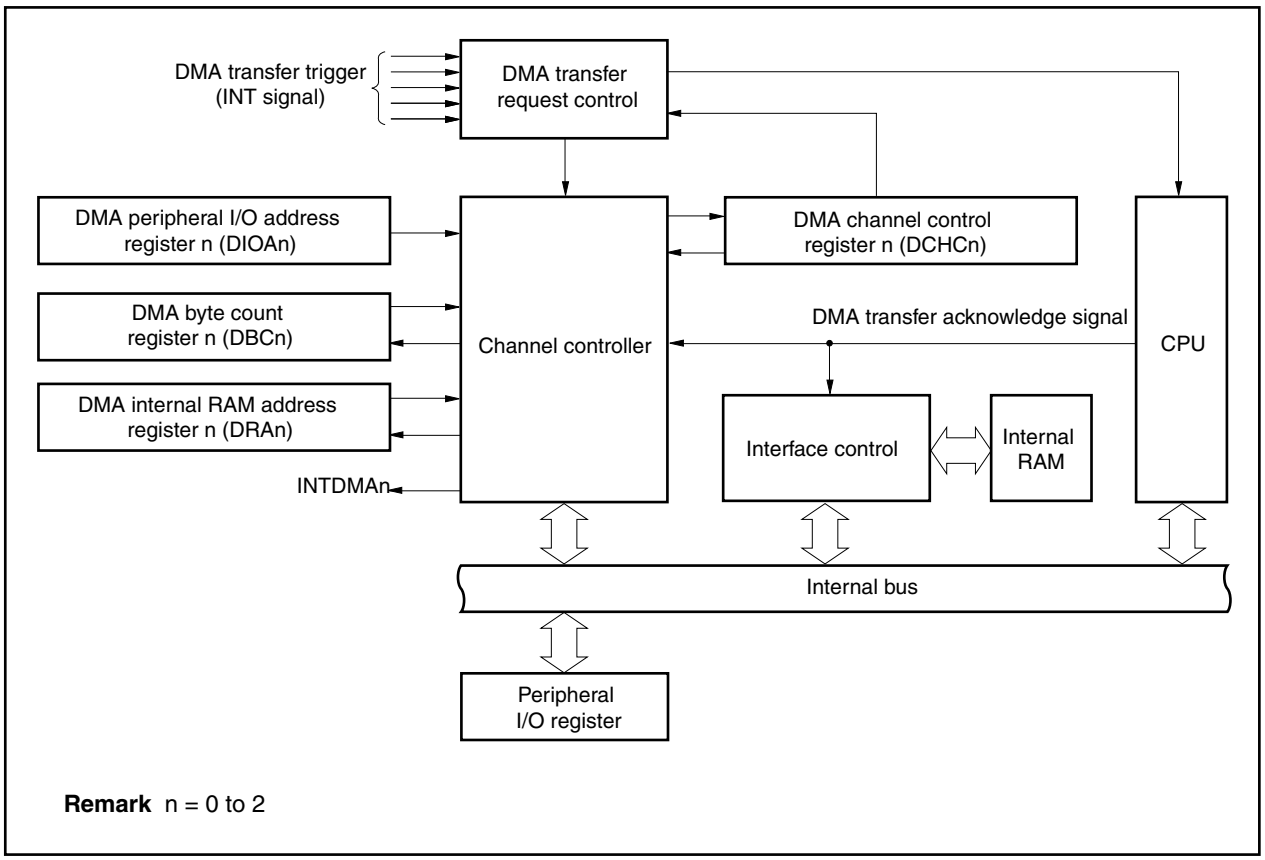
★ 12.2 Features

- DMA channels: 3
- Transfer unit: 8/16 bits
- Maximum transfer count: 256 times (8-bit transfer)
- Transfer mode: Single transfer
- Transfer clock: 4 clocks (min.) ($4 \times f_{CPU}$)
- Transfer request: Request by interrupt from on-chip peripheral I/O (serial interface, timer/counter, A/D converter)
- Transfer target: Internal RAM $\Longleftrightarrow$ peripheral I/O

Remark f_{CPU}: CPU operation clock

⋆ 12.3 Configuration

Figure 12-1. Block Diagram of DMAC



Remark n = 0 to 2

(1) DMA transfer request control block

The DMA transfer request control block generates a DMA transfer request signal for the CPU when the DMA transfer trigger (INT signal) specified by DMA channel control register n (DCHCn) is input.

When the DMA transfer request signal is acknowledged, the CPU generates a DMA transfer acknowledge signal for the channel control block and interface control block after the current CPU processing has finished.

(2) Channel control block

The channel control block distinguishes the DMA transfer channel n (DMA0 to DMA2) to be transferred and controls the internal RAM, peripheral I/O addresses, and access cycles (internal RAM: 1 clock, peripheral I/O register: 3 clocks) set by the peripheral I/O registers of the channel to be transferred, the transfer direction, and the transfer count. In addition, it also controls the priority order when two or more DMAn transfer triggers (INT signals) are generated simultaneously.

12.4 Control Registers

(1) DMA peripheral I/O address registers 0 to 2 (DIOA0 to DIOA2)

These registers are used to set the peripheral I/O register address for DMA channel n.

These registers are can be read/written in 16-bit units.

After reset: Undefined R/W Address: DIOA0 FFFFF180H
DIOA1 FFFFF190H
DIOA2 FFFFF1A0H

	15	14	13	12	11	10	9	1	0
DIOAn	0	0	0	0	0	0	IOAn9 to IOAn1		0

(n = 0 to 2)

Caution The following peripheral I/O registers must not be set in DIOAn.
P4, P5, P6, P9, P11, P12, PM4, PM5, PM6, PM9, PM11, PM12, PMC12, MM, DWC, BCC, SYC, PSC, PCC, SYS, PRCMD, DIOAn, DRAn, DBCn, DCHCn, interrupt control register (xxICn), ISPR

(2) DMA internal RAM address registers 0 to 2 (DRA0 to DRA2)

These registers are used to set the internal RAM address for DMA channel n. An address is incremented after each transfer is completed, when the DADn bit of the DCHDn register is 0. The incrementation value is “1” during 8-bit transfers and “2” during 16-bit transfers (n = 0 to 2).

These registers are can be read/written in 16-bit units.

After reset: Undefined R/W Address: DRA0 FFFFF182H
DRA1 FFFFF192H
DRA2 FFFFF1A2H

	15	14	13	12	0
DRAn	0	0	0	RAn12 to RAn0	

(n = 0 to 2)

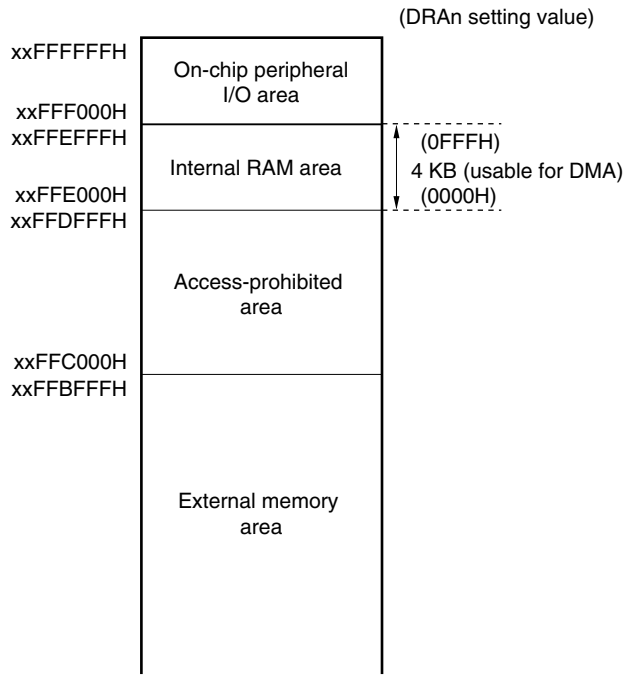
★ **Caution Do not set the RAn12 bit to 1 in the μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 70F3015B, and 70F3015BY.**

The correspondence between DRAn setting value and internal RAM area is shown below.

(a) μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 70F3015B, 70F3015BY

Set the DRAn register to a value in the range of 0000H to 0FFFH (n = 0 to 2).
Setting is prohibited for values between 1000H and 1FFFH.

★ **Figure 12-2. Correspondence Between DRAn Setting Value and Internal RAM (4 KB)**



Cautions 1. Do not set odd addresses for 16-bit transfer (DCHCn register DSn = 1).

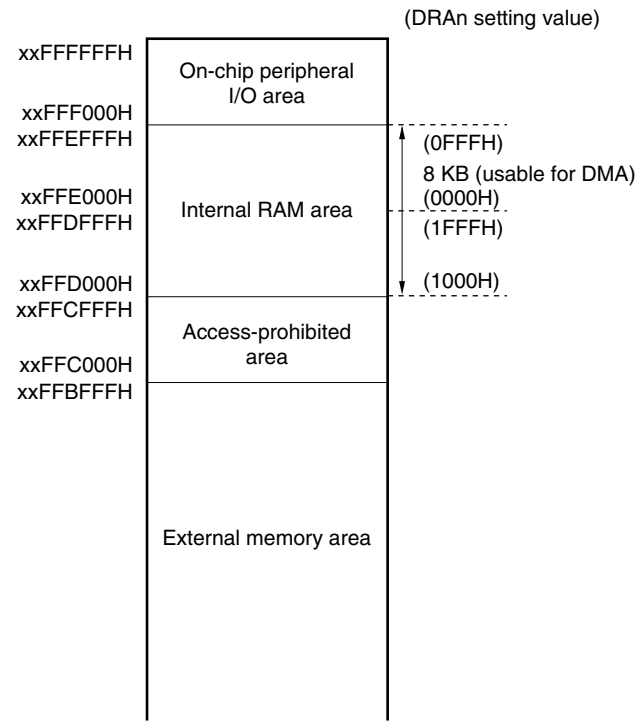
2. While the increment function is being used (DCHCn register DDADn = 0), if the DRAn register value is set to 0FFFH, it will be incremented to 1000H, and will thus become a setting-prohibited value.

Remark The DRAn register setting values are in the parentheses.

(b) μPD703017A, 703017AY, 70F3017A, 70F3017AY

Set the DRAn register to a value in the range of 0000H to 0FFFH or 1000H to 1FFFH (n = 0 to 2).

★ **Figure 12-3. Correspondence Between DRAn Setting Value and Internal RAM (8 KB)**



Caution Do not set odd addresses for 16-bit transfer (DCHCn register DSn =1).

Remark The DRAn register setting values are in the parentheses.

(3) DMA byte count registers 0 to 2 (DBC0 to DBC2)

These are 8-bit registers that are used to set the number of transfers for DMA channel n.

The remaining number of transfers is retained during the DMA transfers.

A value of 1 is decremented once per transfer if the transfer is a byte (8-bit) transfer, and a value of 2 is decremented once per transfer if the transfer is a 16-bit transfer. The transfers are terminated when a borrow operation occurs. Accordingly, "number of transfers − 1" should be set for byte (8-bit) transfers and "(number of transfers − 1) × 2" should be set for 16-bit transfers. During 16-bit transfers, values set to bit 0 are ignored, and 0 is set to bit 0 after decrementation.

These registers are can be read/written in 8-bit units.

After reset: Undefined R/W Address: DBC0 FFFFF184H
DBC1 FFFFF194H
DBC2 FFFFF1A4H

	7	6	5	4	3	2	1	0
DBCn	BCn7	BCn6	BCn5	BCn4	BCn3	BCn2	BCn1	BCn0

(n = 0 to 2)

Caution Values set to bit 0 are ignored during 16-bit transfers.

(4) DMA channel control registers 0 to 2 (DCHC0 to DCHC2)

These registers are used to control the DMA transfer operation mode for DMA channel n.

These registers are can be read/written in 1-bit or 8-bit units.

(1/2)

After reset: 00H R/W Address: DCHC0: FFFFF186H, DCHC1: FFFFF196H, DCHC2: FFFFF1A6H

	7	6	5	4	3	2	1	0
DCHCn	TCn	0	DADn	TTYPn1	TTYPn0	TDIRn	DSn	ENn

(n = 0 to 2)

TCn	DMA Transfer Completed/Not Completed[Note 1]
0	Not completed
1	Completed

DADn	Internal RAM Address Count Direction Control
0	Increment
1	Address is fixed

Channel n	TTYPn1	TTYPn0	Setting of Trigger for DMA Transfer
0	0	0	INTCSI0/INTIIC0[Note 2]
	0	1	INTTM00
	1	0	INTAD
	1	1	INTTM4
1	0	0	INTCSI1/INTSR0
	0	1	INTST1
	1	0	INTPCSI0/INTIIC0[Note 2]
	1	1	INTTM4
2	0	0	INTSR1
	0	1	INTST0
	1	0	INTAD
	1	1	INTTM5

Notes 1. TCn (n = 0 to 2) is set to 1 when a specified number of transfers are completed, and is cleared to 0 when a write instruction is executed.

★ **2.** INTIIC0 is available only in the μPD703014Y, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY.

★ **Cautions 1. DMA transfer is started using an interrupt request signal (INTxxn above) generated from an on-chip peripheral I/O. DMA transfer is not started even if the xxIFn bit of the interrupt control register (xxICn) that is the target of the INTxxn signal is set 1.**

2. If the INTxxn signal is generated in synchronization with the external clock, do not set the INTxxn signal as multiple DMA transfer triggers at the same time. For example, do not set INTCSI0 as the trigger of both DMA channel 0 and DMA channel 1.

The trigger of the DMA transfer generated in synchronization with the external clock is shown below.

- **INTCSI0 when $\overline{SCK0}$ pin input is selected as serial clock**
- **INTCSI0 when timer 2 output (TO2) that operates with TI2 pin input is selected as serial clock**
- **INTTM4 when TI4 pin input is selected as count clock.**

(2/2)

TDIRn	Transfer Direction Control Between Peripheral I/O and Internal RAM[Note 1]
0	From internal RAM to peripheral I/O
1	From peripheral I/O to internal RAM

DSn	Control of Transfer Data Size for DMA Transfer[Note 1]
0	8-bit transfer
1	16-bit transfer

ENn	Control of DMA Transfer Enable/Disable Status[Note 2]
0	Disabled
1	Enabled (reset to 0 after DMA transfer is completed)

Notes 1. Make sure that the transfer format conforms to the peripheral I/O register specifications (access-enabled data size, read/write, etc.) for the DMA peripheral I/O address register (DIOAn).

2. After the specified number of transfer is completed, this bit is cleared to 0.

⋆ 12.5 Operation

The DMA controller of the V850/SA1 supports only the single transfer mode.

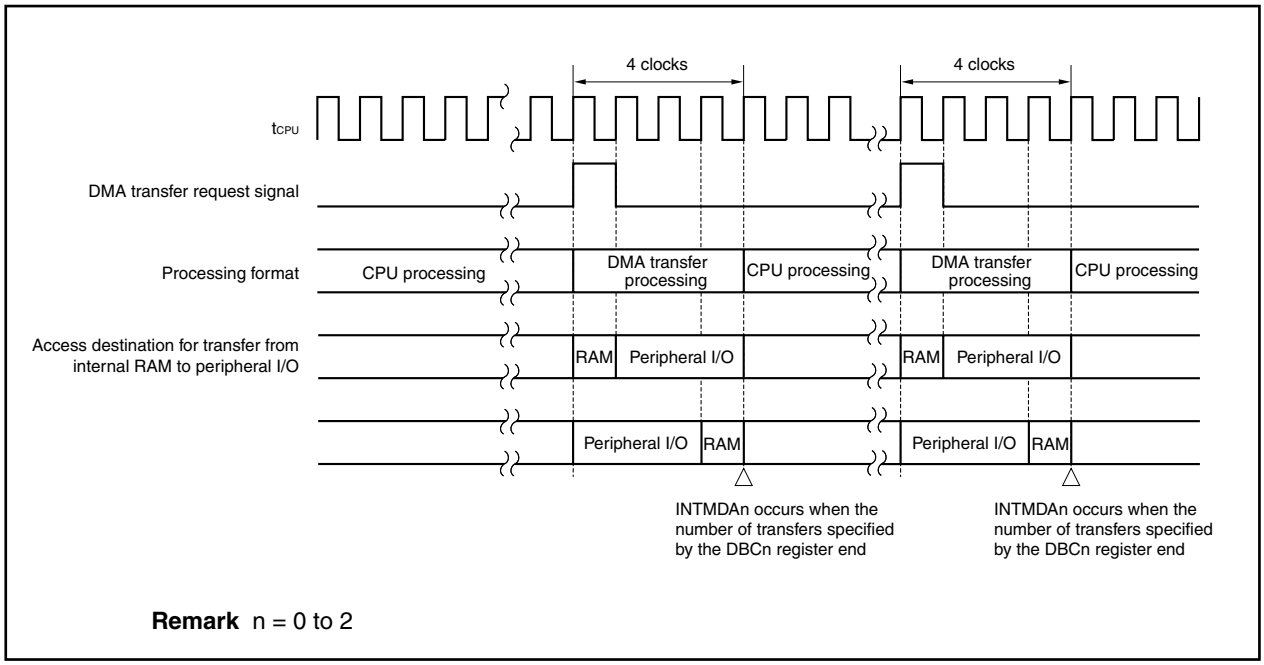
When a DMA transfer request (INTxxx: refer to **12.4 (4) DMA channel control registers 0 to 2 (DCHC0 to DCHC2**)) is generated during CPU processing, a single DMA transfer is started after the current CPU processing has finished. Regardless of the transfer direction, 4 CPU clocks (f_{CPU}) are required for one DMA transfer. The 4 CPU clocks are divided as follows.

- Internal RAM access: 1 clock
- Peripheral I/O access: 3 clocks

After one DMA transfer (8/16 bits) ends, control always shifts to the CPU processing and waits for the generation of the next DMA transfer request (INTxxx). After the specified number of data transfers ends, the DMA transfer end interrupt requests (INTDMA0 to INTDMA2) are generated for each channel of the interrupt controller if the TCn bit of the DOCHn register becomes 1.

The DMA transfer operation timing chart is shown below.

Figure 12-4. DMA Transfer Operation Timing

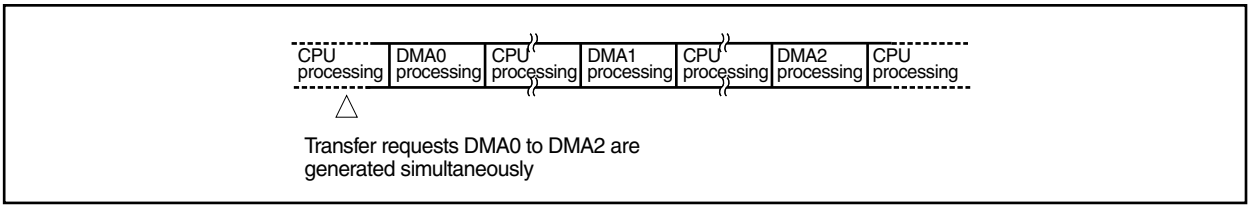


Remark n = 0 to 2

If two or more DMA transfer requests are generated simultaneously, the DMA transfer requests are executed in a priority order of DMA0 > DMA1 > DMA2. While a higher priority DMA transfer request is being executed, the lower priority DMA transfer requests are held pending. After the higher priority DMA transfer ends, control always shifts to the CPU processing once, and then the lower priority DMA transfer request is executed after the CPU processing ends.

The processing when the transfer requests DMA0 to DMA2 are generated simultaneously is shown below.

Figure 12-5. Processing When Transfer Requests DMA0 to DMA2 Are Generated Simultaneously



DMA operation stops only in the IDLE/software STOP mode. In the HALT mode, DMA operation continues, DMA also operates during the bus hold period and after access to external memory.

★ 12.6 Cautions

- To manipulate the bits of the interrupt control register (xxICn) in the EI state when using the DMA function, execute the DI instruction before manipulation and EI instruction after manipulation. Alternatively, clear (0) the xxIFn bit at the start of the interrupt servicing routine (when not using the DMA function, these manipulations are not required).

- If an interrupt request signal is generated in synchronization with the external clock, setting the interrupt request signal as multiple DMA transfer triggers is prohibited. If set, the priority of the DMA may be reversed.

Remark xx: Peripheral unit identification name (see **Table 5-2**)
N: Peripheral unit number (see **Table 5-2**)

CHAPTER 13 REAL-TIME OUTPUT FUNCTION (RTO)

13.1 Function

The V850/SA1 incorporates a real-time output function that transfers preset data to real-time output buffer registers (RTBL, RTBH), and then transfers this data with hardware to an external device via the output latches, upon the occurrence of an external interrupt or external trigger.

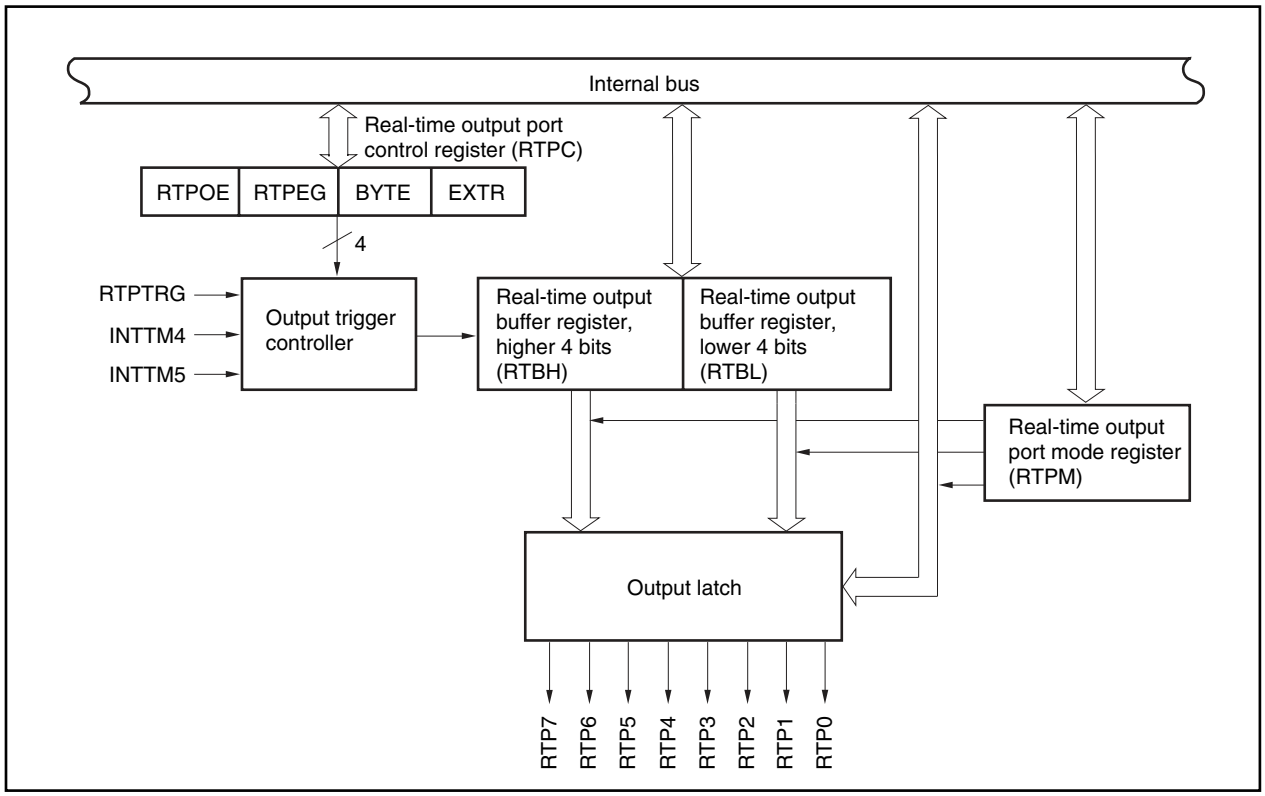
Because RTO can output signals without jitter, it is suitable for controlling a stepper motor.

★ 13.2 Features

- 8-bit real-time output unit
- Port mode and real-time output mode can be selected in 1-bit units
- 8 bits × 1 channel or 4 bits × 2 channels can be selected
- Trigger signal: Selectable from the following three.
  External interrupt: RTPTRG
  Internal interrupt: INTTM4, INTTM5

13.3 Configuration

Figure 13-1. Block Diagram of RTO



RTO consists of the following hardware.

Table 13-1. Configuration of RTO

Item	Configuration
Registers	Real-time output buffer registers (RTBL, RTBH)
Control registers	Real-time output port mode register (RTPM) Real-time output port control register (RTPC)

(1) Real-time output buffer registers (RTBL, RTBH)

RTBL and RTBH are 4-bit registers that hold output data in advance.

These registers are mapped to independent addresses in the special function register (SFR) area as shown in Figure 13-2.

If an operation mode of 4 bits × 2 channels is specified, data can be individually set to RTBL and RTBH. The data of both the registers can be read all at once by specifying the address of either of the registers.

If an operation mode of 8 bits × 1 channel is specified, 8-bit data can be set to both RTBL and RTBH respectively by writing the data to either of the registers. The data of both the registers can be read all at once by specifying the address of either of the registers.

These registers are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 00H.

Figure 13-2 shows the configuration of RTBL and RTBH, and Table 13-2 shows the operation to be performed when RTBL and RTBH are manipulated.

Figure 13-2. Configuration of Real-Time Output Buffer Registers

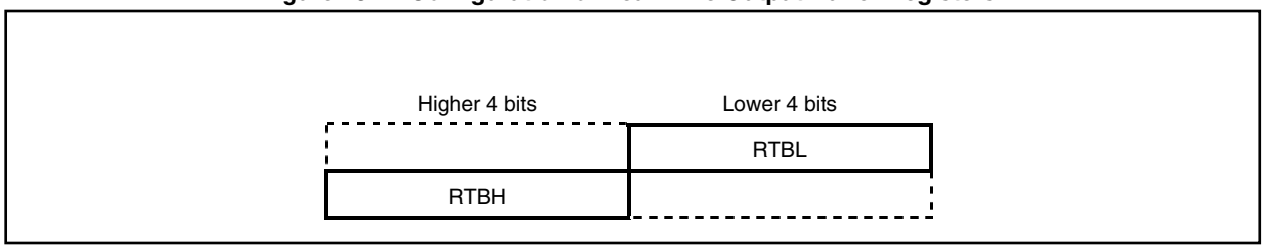


Table 13-2. Operation When Real-Time Output Buffer Registers Are Manipulated

Operation Mode	Register to Be Manipulated	Read[Note 1]		Write[Note 2]	
		Higher 4 bits	Lower 4 bits	Higher 4 bits	Lower 4 bits
4 bits × 2 channels	RTBL	RTBH	RTBL	Invalid	RTBL
	RTBH	RTBH	RTBL	RTBH	Invalid
8 bits × 1 channel	RTBL	RTBH	RTBL	RTBH	RTBL
	RTBH	RTBH	RTBL	RTBH	RTBL

Notes 1. Only the bits set in the real-time output port mode (RTPM) can be read. If a bit set in the port mode is read, 0 is read.

2. Set output data to RTBL and RTBH after setting the real-time output port until the real-time output trigger is generated.

★ (2) Output latch

This is the output latch to which the value set by the real-time output buffer register (RTBL, RTBH) is automatically transferred when the real-time output trigger occurs. Output latches cannot be accessed.

A port specified as a real-time output port cannot set data to the port output latch. To set the initial values of the real-time output port, set data to the port output latch in the port mode and then set to the real-time output port mode (refer to **13.5 Usage**).

13.4 Control Registers

RTO is controlled by using the following two registers.

- Real-time output port mode register (RTPM)
- Real-time output port control register (RTPC)

(1) Real-time output port mode register (RTPM)

This register selects real-time output port mode or port mode in 1-bit units.

RTPM is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RTPM to 00H.

After reset: 00H R/W Address: FFFFF3A4H

	7	6	5	4	3	2	1	0
RTPM	RTPM7	RTPM6	RTPM5	RTPM4	RTPM3	RTPM2	RTPM1	RTPM0

RTPMn	Selection of Real-Time Output Port (n = 0 to 7)
0	Port mode
1	Real-time output port mode

Cautions 1. Set a port pin to the output mode when it is used as a real-time output port pin.

2. Data cannot be set to the output latch for a port pin set as a real-time output port pin. To set an initial value, therefore, set the data to the output latch before setting the port pin to the real-time output port mode (refer to 13.5 Usage).

(2) Real-time output port control register (RTPC)

This register sets the operation mode and output trigger of the real-time output port.

The relationship between the operation mode and output trigger of the real-time output port is as shown in Table 13-3.

RTPC is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets RTPC to 00H.

After reset: 00H R/W Address: FFFFF3A6H

	7	6	5	4	3	2	1	0
RTPC	RTPOE	RTPEG	BYTE	EXTR	0	0	0	0

RTPOE	Control of Operation of Real-Time Output Port
0	Operation disabled[Note]
1	Operation enabled

RTPEG	Valid Edge of RTPTRG Signal
0	Falling edge
1	Rising edge

BYTE	Operation Mode of Real-Time Output Port
0	4 bits × 2 channels
1	8 bits × 1 channel

EXTR	Control of Real-Time Output by RTPTRG Signal
0	RTPTRG not used as real-time output trigger
1	RTPTRG used as real-time output trigger

Note RTP0 to RTP7 output 0 if the real-time output operation is disabled (RTPOE = 0).

Table 13-3. Operation Mode and Output Trigger of Real-Time Output Port

BYTE	EXTR	Operation Mode	RTBH → Port Output	RTBL → Port Output
0	0	4 bits × 2 channels	INTTM5	INTTM4
	1		INTTM4	RTPTRG
1	0	8 bits × 1 channel	INTTM4	
	1		RTPTRG	

★ 13.5 Usage

(1) Disable the real-time output operation.
Clear bit 7 (RTPOE) of the real-time output port control register (RTPC) to 0.

(2) Initialization
(i) Set the initial value to the output latch of port 10.
(ii) Set the PM10 register to output mode.
(iii) Specify the real-time output port mode or port mode in 1-bit units.
Set the real-time output port mode register (RTPM).
(iv) Selects a trigger and valid edge.
Set bits 4, 5, and 6 (EXTR, BYTE, and RTPEG) of RTPC.
(v) Set the same value as (i) to the real-time output buffer registers (RTBH and RTBL).

(3) Enable the real-time output operation.
Set RTPOE to 1.

(4) Set the output latch of port 10 to 0 are set the next output to RTBH and RTBL before the selected transfer trigger is generated.

(5) Set the next real-time output value to RTBH and RTBL by interrupt servicing corresponding to the selected trigger.

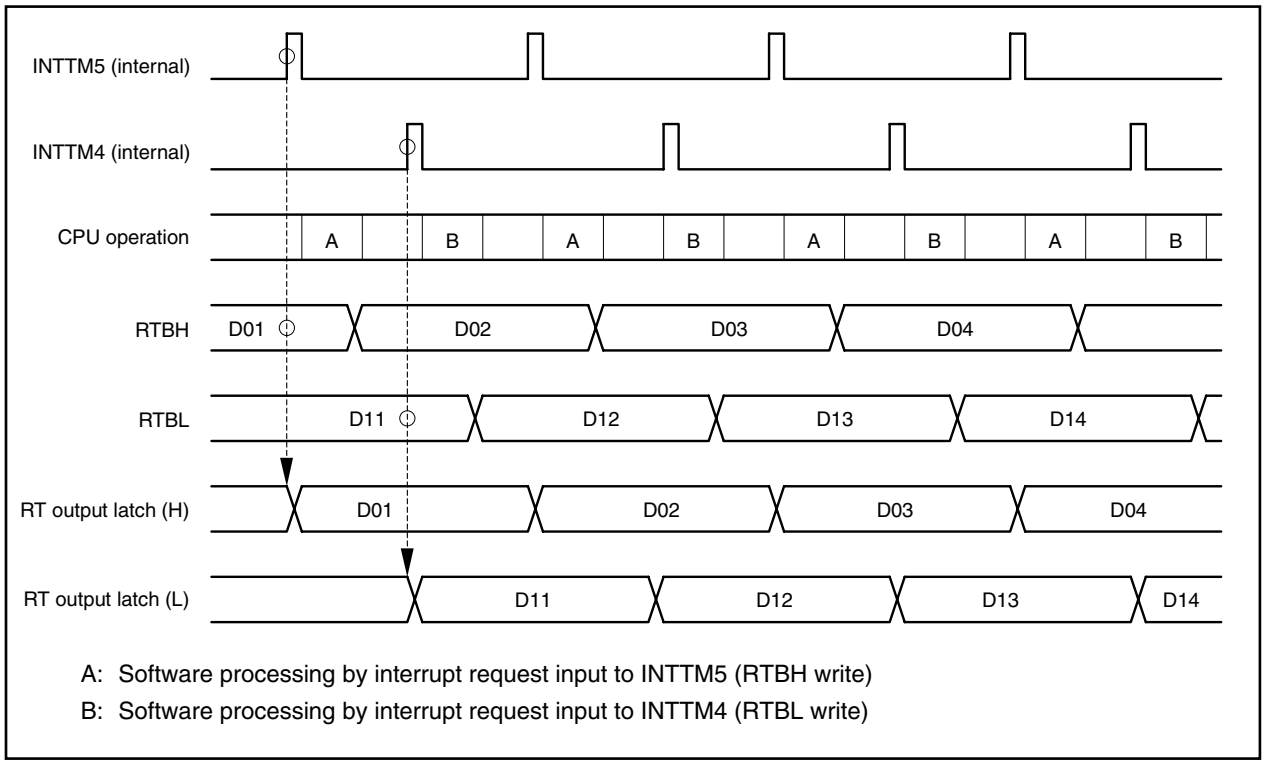
13.6 Operation

If the real-time output operation is enabled by setting bit 7 (RTPOE) of the real-time output port control register (RTPC) to 1, the data of the real-time output buffer registers (RTBH and RTBL) is transferred to the output latch in synchronization with the generation of the selected transfer trigger (set by EXTR and BYTE[Note]). Of the transferred data, only the data of the bits specified in the real-time output port mode by the real-time output port mode register (RTPM) is output from the bits of RTP0 to RTP7. The bits specified in the port mode by RTPM output 0.

If the real-time output operation is disabled by clearing RTPOE to 0, RTP0 to RTP7 output 0 regardless of the setting of RTPM.

Note EXTR: Bit 4 of real-time output port control register (RTPC)
BYTE: Bit 5 of real-time output port control register (RTPC)

Figure 13-3. Example of Operation Timing of RTO (When EXTR = 0, BYTE = 0)



A: Software processing by interrupt request input to INTTM5 (RTBH write)
B: Software processing by interrupt request input to INTTM4 (RTBL write)

13.7 Cautions

(1) Before performing initialization, disable the real-time output operation by clearing bit 7 (RTPOE) of the real-time output port control register (RTPC) to 0.

(2) Once the real-time output operation is disabled (RTPOE = 0), be sure to set the same initial value as the output latch to the real-time output buffer registers (RTBH and RTBL) before enabling the real-time output operation (RTPOE = 0 → 1).

★ (3) Operation cannot be guaranteed if a conflict between the following signals occurs. Use a software to avoid a conflict.

- Conflict between the switch operation from the real-time output port mode to the port mode (RTPOE = 0) and the valid edge of the selected real-time output trigger

- Conflict between the write operation to the real-time output buffer register (RTBL, RTBH) in the real-time output port mode and the valid edge of the selected real-time output trigger

CHAPTER 14 PORT FUNCTION

14.1 Port Configuration

The V850/SA1 includes 85 I/O port pins configuring ports 0 to 12 (13 ports are input only).

There are three power supplies for the I/O buffers; AV_{DD}, BV_{DD}, and V_{DD}, which are described below.

★ **Table 14-1. Pin I/O Buffer Power Supplies**

Power Supply	Corresponding Pins	Usable Voltage Range
AV_{DD}	Port 7, port 8	2.7 V ≤ AV_{DD} ≤ 3.6 V
BV_{DD}	Port 4, port 5, port 6, port 9, port 12 CLKOUT	2.7 V ≤ BV_{DD} ≤ 3.6 V
V_{DD}	Port 0, port 1, port 2, port 3, port 10, port 11, $\overline{\text{RESET}}$	2.7 V ≤ V_{DD} ≤ 3.6 V

14.2 Port Pin Function

14.2.1 Port 0

Port 0 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. A pull-up resistor can be connected in 1-bit units (software pull-up function).

When P00 to P04 are used as the NMI and INTP0 to INTP3 pins, noise is eliminated from these pins by an analog noise eliminator.

When P05 to P07 are used as the INTP4/ADTRG, INTP5/RTPTRG, and INTP6 pins, noise is eliminated from these pins by a digital noise eliminator.

After reset: 00H R/W Address: FFFFF000H

	7	6	5	4	3	2	1	0
P0	P07	P06	P05	P04	P03	P02	P01	P00

P0n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Output 0
1	Output 1

Remark In input mode: When port 0 (P0) is read, the pin levels at that time are read. Writing to P0 writes the values to that register. This does not affect the input pins.

In output mode: When port 0 (P0) is read, the P0 values are read. Writing to P0 writes the values to that register, and those values are immediately output.

Port 0 includes the following alternate functions.

Table 14-2. Alternate Functions of Port 0

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 0	P00	NMI	I/O	Yes	Analog noise elimination
	P01	INTP0			
	P02	INTP1			
	P03	INTP2			
	P04	INTP3			
	P05	INTP4/ADTRG			Digital noise elimination
	P06	INTP5/RTPTRG			
	P07	INTP6			

Note Software pull-up function

(1) Function of P0 pins

Port 0 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 0 mode register (PM0).

In output mode, the values set to each bit are output to port 0 (P0). When using this port in output mode, either the valid edge of each interrupt request should be made invalid or each interrupt request should be masked (except for NMI requests).

When using this port in input mode, the pin statuses can be read by reading P0. Also, the P0 register (output latch) values can be read by reading P0 while in output mode.

The valid edge of NMI and INTP0 to INTP6 are specified via rising edge specification register 0 (EGP0) and falling edge specification register 0 (EGN0).

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 0 (PU0).

When a reset is input, the settings are initialized to input mode. Also, the valid edge of each interrupt request becomes invalid (NMI and INTP0 to INTP6 do not function immediately after reset).

(2) Noise elimination

(a) Elimination of noise from NMI and INTP0 to INTP3 pins

An on-chip noise eliminator uses analog delay to eliminate noise. Consequently, if a signal having a constant level is input for longer than a specified time to these pins, it is detected as a valid edge. Such edge detection occurs after the specified amount of time.

(b) Elimination of noise from INTP4 to INTP6, ADTRG, and RTPTRG pins

A digital noise eliminator is provided on chip.

This circuit uses digital sampling. A pin's input level is detected using a sampling clock (f_{XX}), and noise elimination is performed if the same level is not detected three times consecutively.

Cautions 1. If the input pulse width is 2 to 3 clocks, whether it will be detected as a valid edge or eliminated as noise is undefined.

2. To ensure correct detection of pulses as valid edges, constant-level input is required for 3 clocks or more.

3. If noise is occurring in synchronization with the sampling clock, noise cannot be eliminated. In such cases, attach a filter to the input pins to eliminate the noise.

4. Noise elimination is not performed when these pins are used as an normal input port pins.

(3) Control registers

(a) Port 0 mode register (PM0)

PM0 can be read/written in 1-bit or 8-bit units.

After reset: FFH R/W Address: FFFFF020H

	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

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

PU0 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF080H

	7	6	5	4	3	2	1	0
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00

PU0n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(c) Rising edge specification register 0 (EGP0)

EGP0 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF0C0H

	7	6	5	4	3	2	1	0
EGP0	EGP07	EGP06	EGP05	EGP04	EGP03	EGP02	EGP01	EGP00

EGP0n	Control of Rising Edge Detection (n = 0 to 7)
0	Interrupt request signal did not occur at rising edge
1	Interrupt request signal occurred at rising edge

Remark n = 0: Control of NMI pin
n = 1 to 7: Control of INTP0 to INTP6 pins

(d) Falling edge specification register 0 (EGN0)

EGN0 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF0C2H

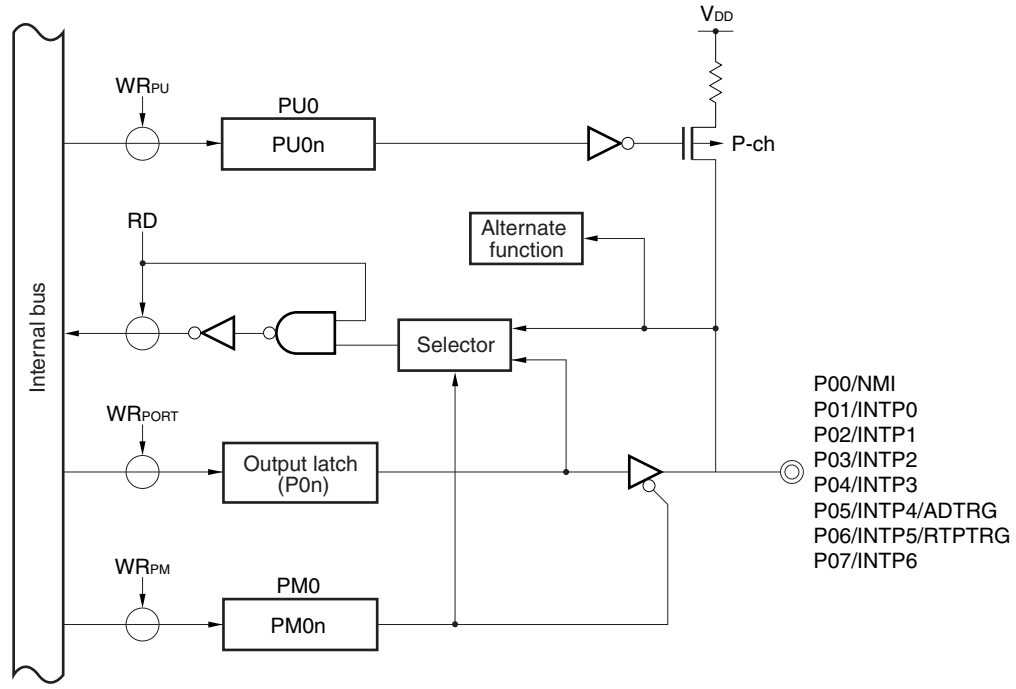
	7	6	5	4	3	2	1	0
EGN0	EGN07	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00

EGN0n	Control of Falling Edge Detection (n = 0 to 7)
0	Interrupt request signal did not occur at falling edge
1	Interrupt request signal occurred at falling edge

Remark n = 0: Control of NMI pin
n = 1 to 7: Control of INTP0 to INTP6 pins

(4) Block diagram (port 0)

Figure 14-1. Block Diagram of P00 to P07



Remarks 1. PU0: Pull-up resistor option register 0
PM0: Port 0 mode register
RD: Read signal of port 0
WR: Write signal of port 0

2. n = 0 to 7

14.2.2 Port 1

Port 1 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units. A pull-up resistor can be connected in 1-bit units (software pull-up function).

Bits 0, 1, 2, 4, and 5 are selectable as normal outputs or N-ch open-drain outputs.

After reset: 00H R/W Address: FFFFF002H

	7	6	5	4	3	2	1	0
P1	0	0	P15	P14	P13	P12	P11	P10

P1n	Control of Output Data (in Output Mode) (n = 0 to 5)
0	Output 0
1	Output 1

Remark In input mode: When port 1 (P1) is read, the pin levels at that time are read. Writing to P1 writes the values to that register. This does not affect the input pins.

In output mode: When port 1 (P1) is read, the P1 values are read. Writing to P1 writes the values to that register, and those values are immediately output.

Port 1 includes the following alternate functions.

Table 14-3. Alternate Functions of Port 1

Pin Name		Alternate Function	I/O	PULL[Note 1]	Remark
Port 1	P10	SI0/SDA[Note 2]	I/O	Yes	Selectable as N-ch open-drain output
	P11	SO0			
	P12	$\overline{\text{SCK0}}$/SCL0[Note 2]			
	P13	SI1/RXD0			–
	P14	SO1/TXD0			Selectable as N-ch open-drain output
	P15	$\overline{\text{SCK1}}$/ASCK0			

Notes 1. Software pull-up function

★ **2.** μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, 70F3017AY only.

(1) Function of P1 pins

Port 1 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 1 mode register (PM1).

In output mode, the values set to each bit are output to port 1 (P1). The port 1 function register (PF1) can be used to specify whether P10 to P12, P14, and P15 are normal outputs or N-ch open-drain outputs.

When using this port in input mode, the pin statuses can be read by reading P1. Also, the P1 (output latch) values can be read by reading P1 while in output mode.

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 1 (PU1).

Clear P1 and PM1 to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 1 mode register (PM1)

PM1 can be read/written in 1-bit or 8-bit units.

After reset: 3FH R/W Address: FFFFF022H

	7	6	5	4	3	2	1	0
PM1	0	0	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Control of I/O Mode (n = 0 to 5)
0	Output mode
1	Input mode

(b) Pull-up resistor option register 1 (PU1)

PU1 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF082H

	7	6	5	4	3	2	1	0
PU1	0	0	PU15	PU14	PU13	PU12	PU11	PU10

PU1n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 5)
0	Do not connect
1	Connect

(c) Port 1 function register (PF1)

PF1 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF0A2H

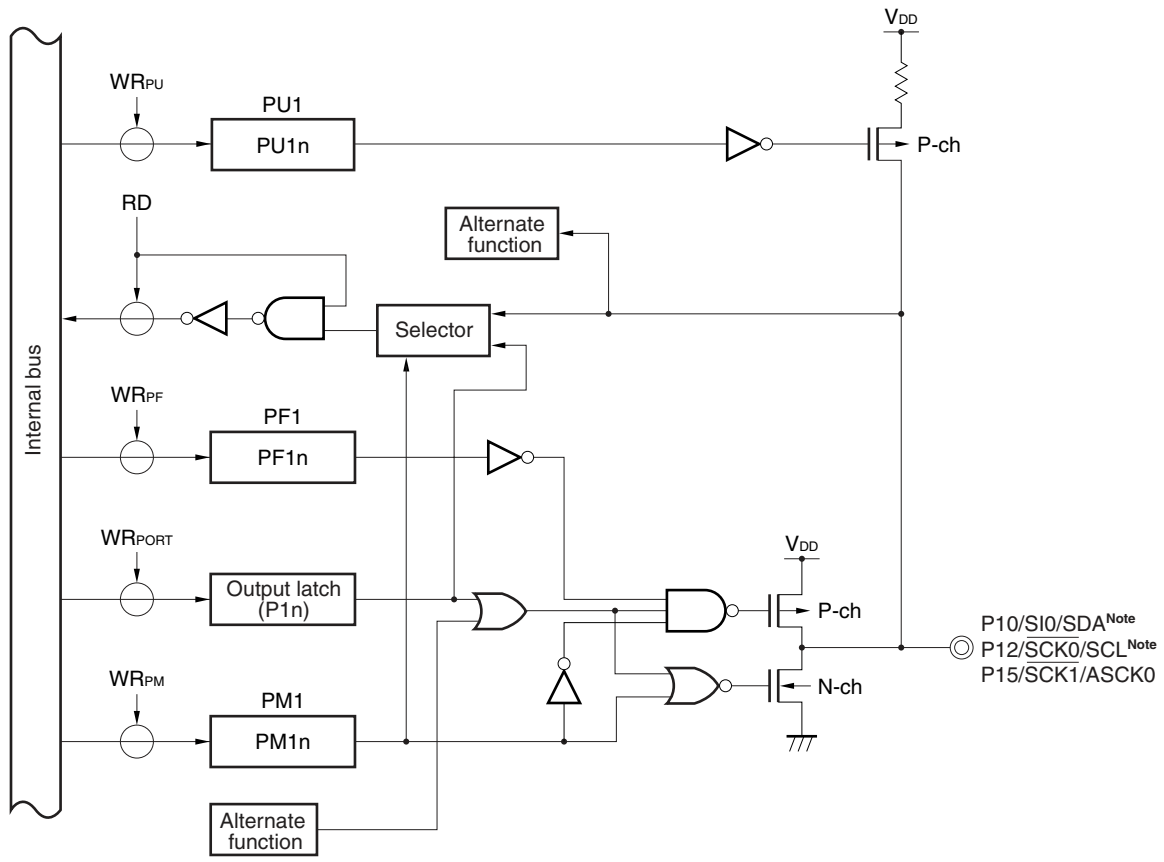
	7	6	5	4	3	2	1	0
PF1	0	0	PF15	PF14	0[Note]	PF12	PF11	PF10

PF1n	Control of Normal Output/N-ch Open-Drain Output (n = 0 to 2, 4, 5)
0	Normal output
1	N-ch open-drain output

Note Bit 3 is fixed to normal output.

(3) Block diagrams (port 1)

Figure 14-2. Block Diagram of P10, P12, and P15

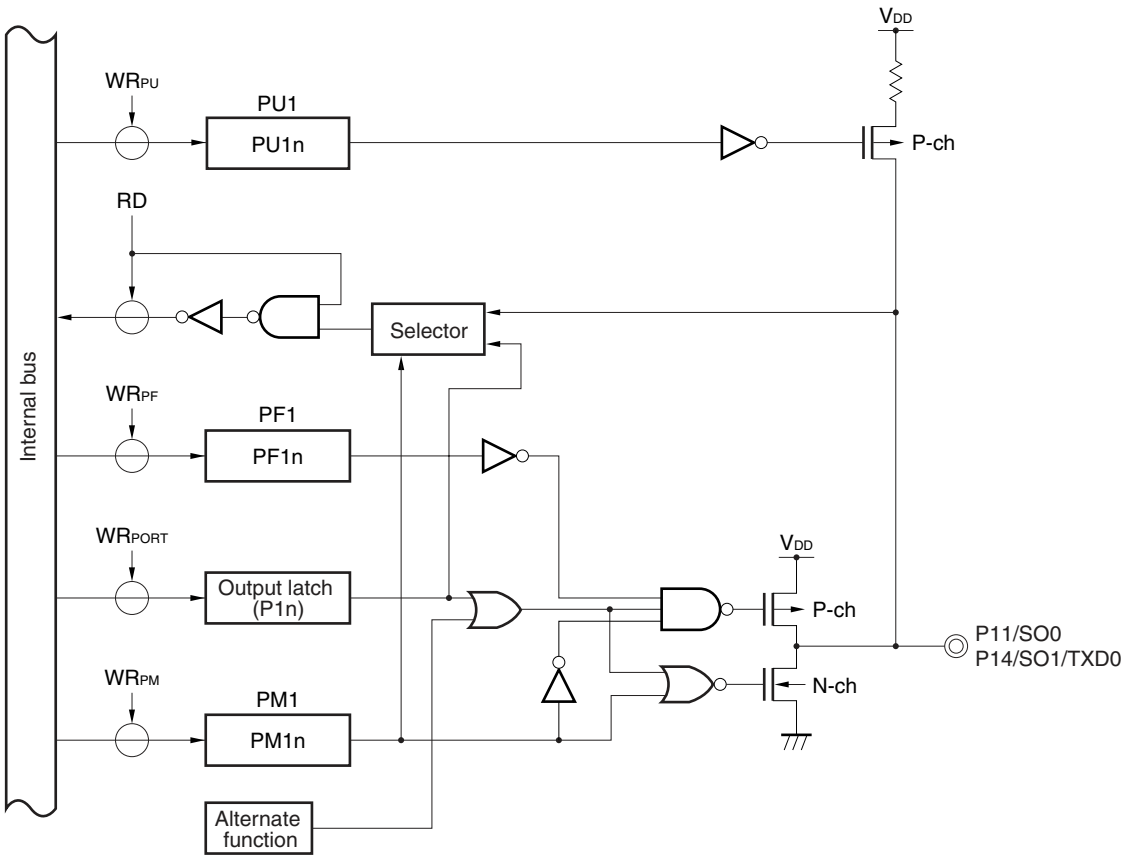


★ **Note** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY

Remarks **1.** PU1: Pull-up resistor option register 1
PF1: Port 1 function register
PM1: Port 1 mode register
RD: Read signal of port 1
WR: Write signal of port 1

2. n = 0, 2, 5

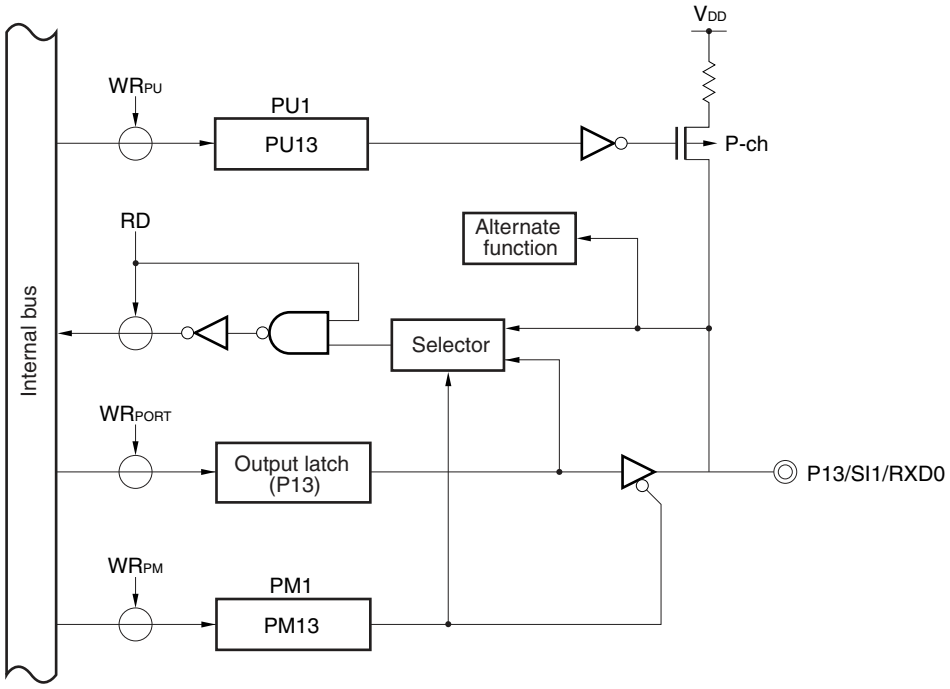
Figure 14-3. Block Diagram of P11 and P14



Remarks **1.** PU1: Pull-up resistor option register 1
PF1: Port 1 function register
PM1: Port 1 mode register
RD: Read signal of port 1
WR: Write signal of port 1

2. n = 1, 4

Figure 14-4. Block Diagram of P13



Remark PU1: Pull-up resistor option register 1
PM1: Port 1 mode register
RD: Read signal of port 1
WR: Write signal of port 1

14.2.3 Port 2

Port 2 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. A pull-up resistor can be connected in 1-bit units (software pull-up function).

P21 and P22 are selectable as normal outputs or N-ch open-drain outputs.

When P26 and P27 are used as the TI2/TI3 pins, noise is eliminated from these pins by a digital noise eliminator.

After reset: 00H R/W Address: FFFFF004H

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23	P22	P21	P20

P2n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Output 0
1	Output 1

Remark In input mode: When port 2 (P2) is read, the pin levels at that time are read. Writing to P2 writes the values to that register. This does not affect the input pins.

In output mode: When port 2 (P2) is read, the P2 values are read. Writing to P2 writes the values to that register, and those values are immediately output.

Port 2 includes the following alternate functions.

Table 14-4. Alternate Functions of Port 2

Pin Name		Alternate Function	I/O	PULL Note	Remark
Port 2	P20	SI2	I/O	Yes	–
	P21	SO2			Selectable as N-ch open-drain output
	P22	$\overline{\text{SCK2}}$			
	P23	RXD1			–
	P24	TXD1			
	P25	ASCK1			
	P26	TI2/TO2			Digital noise elimination
	P27	TI3/TO3			

Note Software pull-up function

(1) Function of P2 pins

Port 2 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 2 mode register (PM2).

In output mode, the values set to each bit are output to port 2 (P2). The port 2 function register (PF2) can be used to specify whether P21 and P22 are normal outputs or N-ch open-drain outputs.

When using this port in input mode, the pin statuses can be read by reading P2. Also, the P2 (output latch) values can be read by reading P2 while in output mode.

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 2 (PU2).

When using the alternate function TI2 and TI3 pins, noise elimination is provided by a digital noise eliminator (same as digital noise eliminator for port 0).

Clear P2 and PM2 to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 2 mode register (PM2)

PM2 can be read/written in 1-bit or 8-bit units.

After reset: FFH R/W Address: FFFFF024H

	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

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

PU2 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF084H

	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

PU2n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(c) Port 2 function register (PF2)

PF2 can be read/written in 1-bit or 8-bit units.

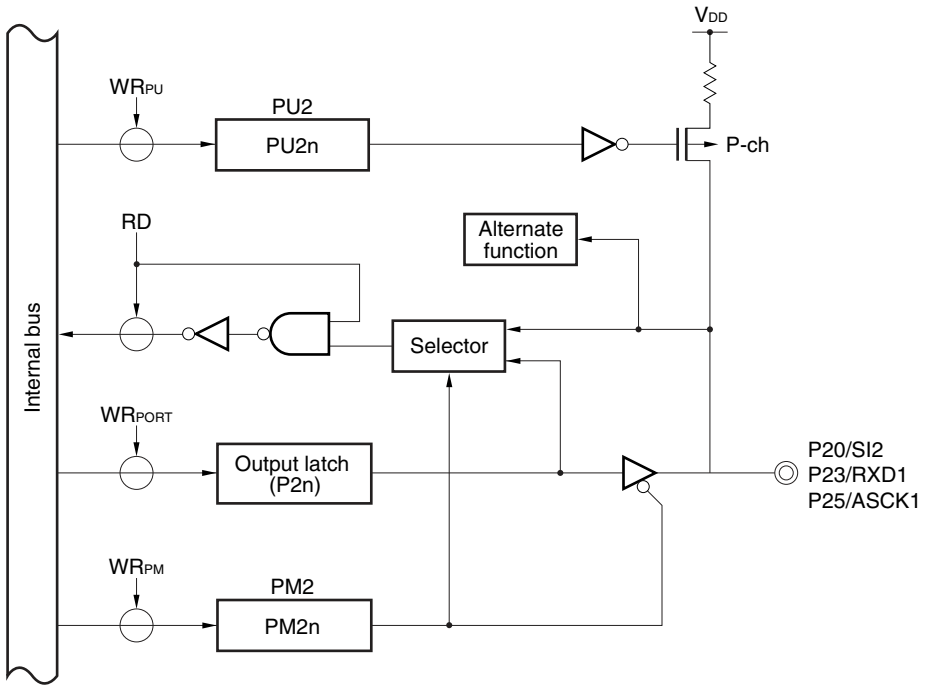
After reset: 00H R/W Address: FFFFF0A4H

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

PF2n	Control of Normal Output/N-ch Open-Drain Output (n = 1, 2)
0	Normal output
1	N-ch open-drain output

(3) Block diagrams (port 2)

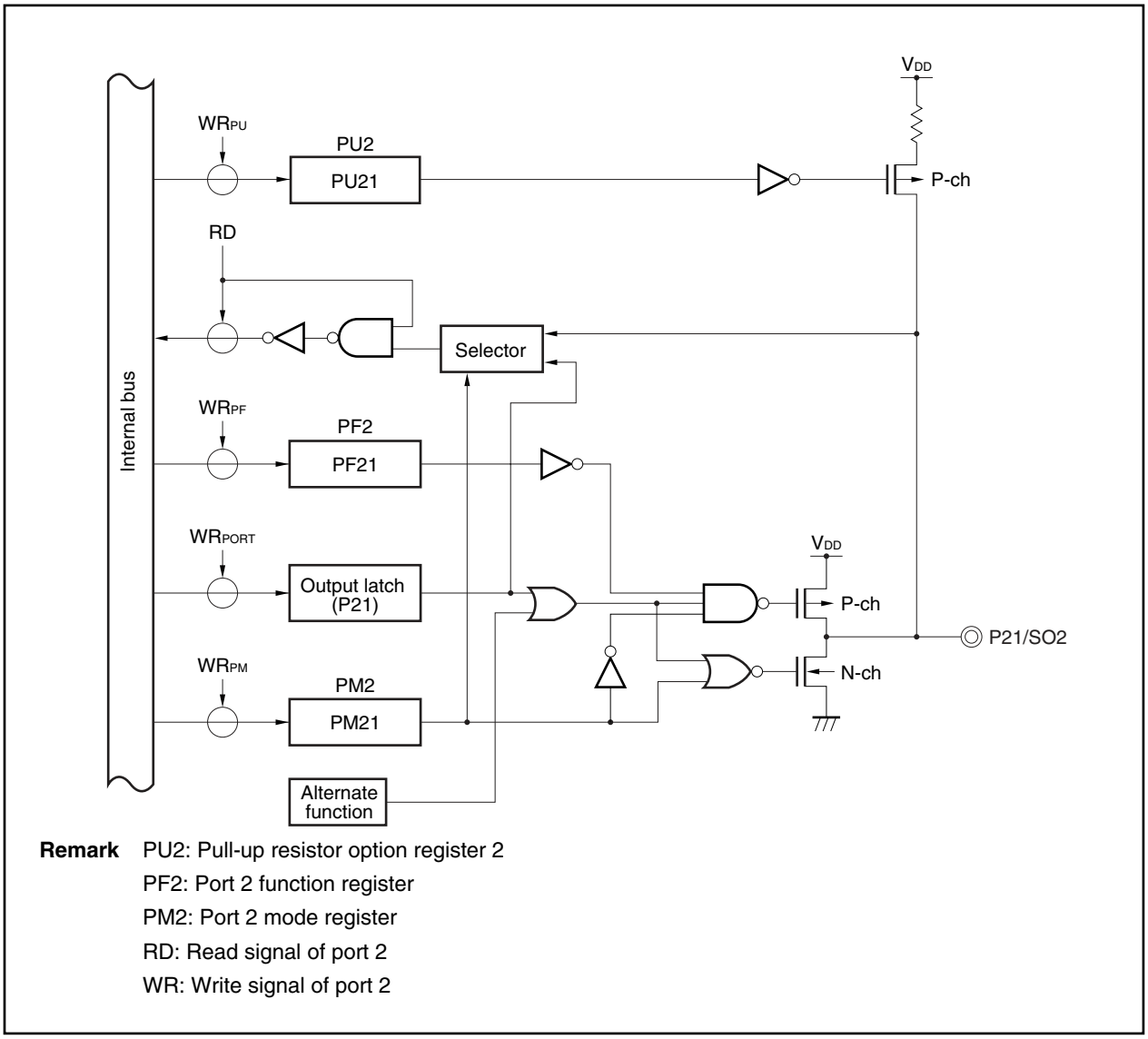
Figure 14-5. Block Diagram of P20, P23, and P25



Remarks 1. PU2: Pull-up resistor option register 2
PM2: Port 2 mode register
RD: Read signal of port 2
WR: Write signal of port 2

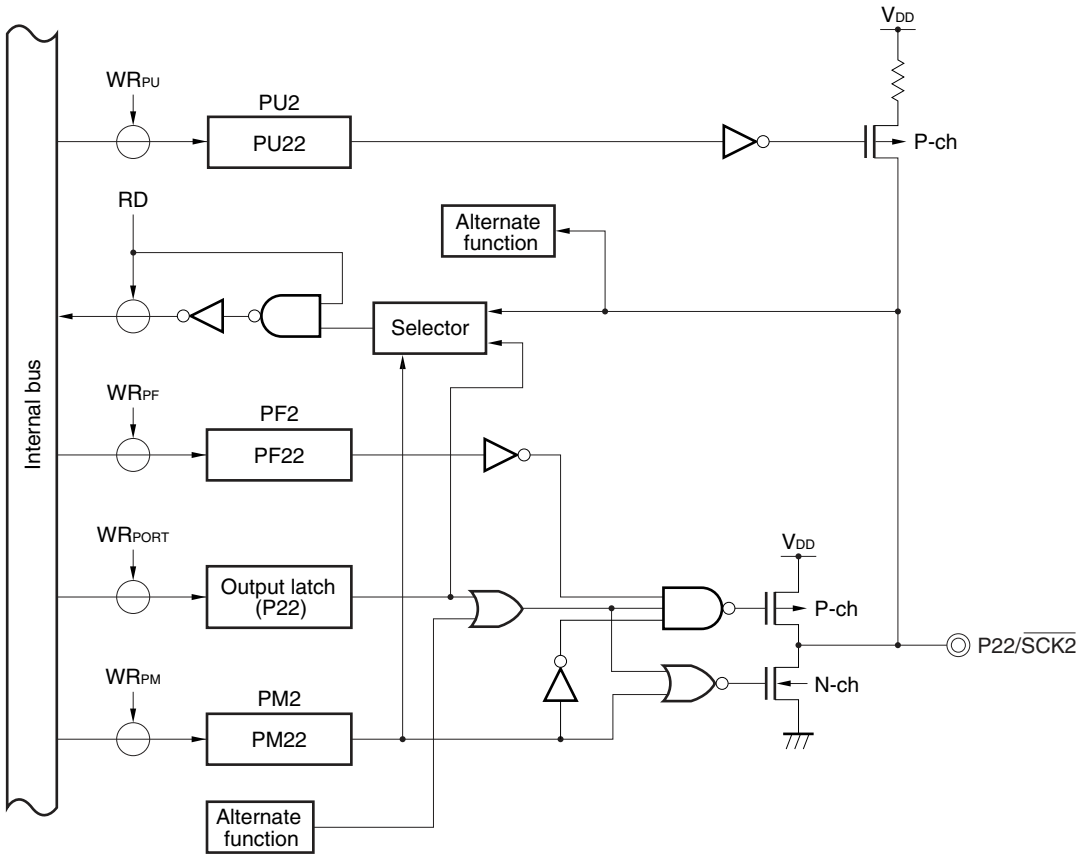
2. n = 0, 3, 5

Figure 14-6. Block Diagram of P21



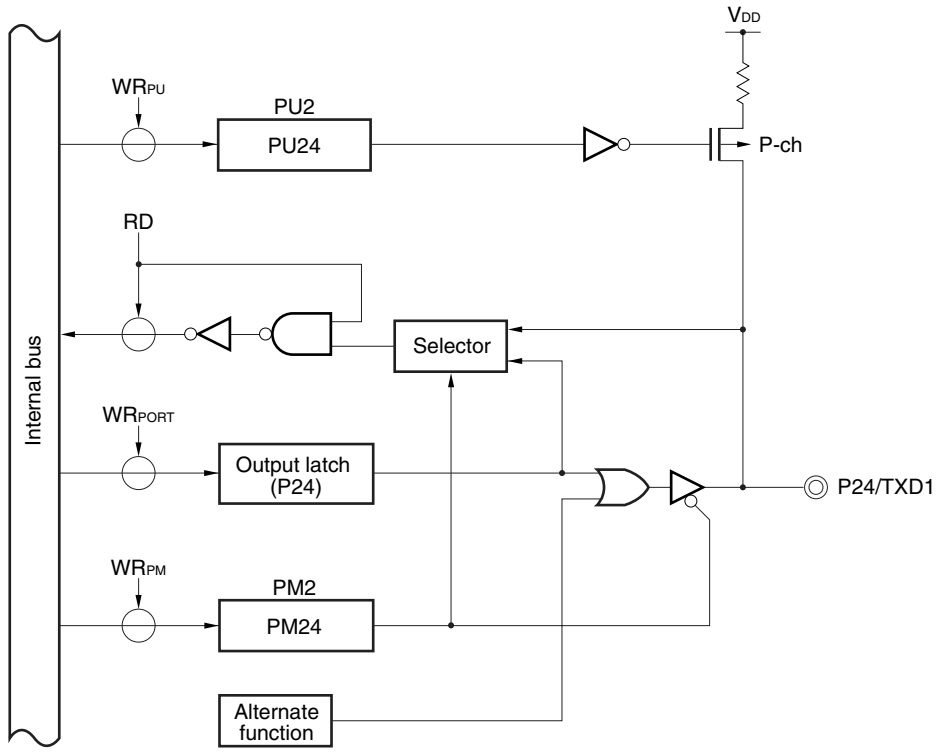
Remark PU2: Pull-up resistor option register 2
PF2: Port 2 function register
PM2: Port 2 mode register
RD: Read signal of port 2
WR: Write signal of port 2

Figure 14-7. Block Diagram of P22



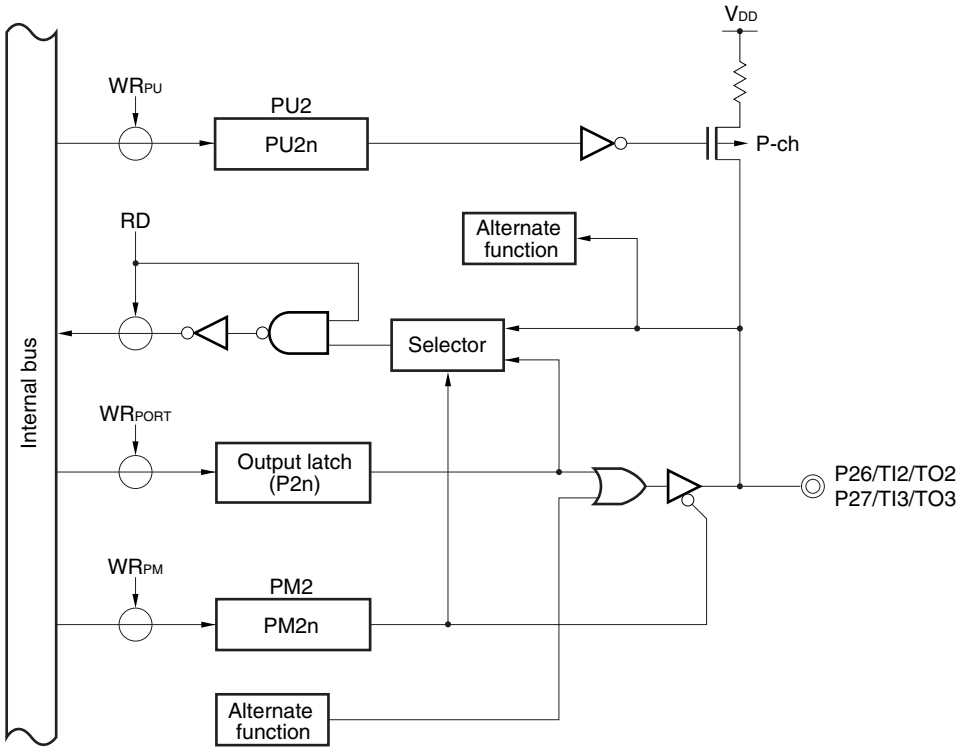
Remark PU2: Pull-up resistor option register 2
PF2: Port 2 function register
PM2: Port 2 mode register
RD: Read signal of port 2
WR: Write signal of port 2

Figure 14-8. Block Diagram of P24



Remark PU2: Pull-up resistor option register 2
PM2: Port 2 mode register
RD: Read signal of port 2
WR: Write signal of port 2

Figure 14-9. Block Diagram of P26 and P27



Remarks 1. PU2: Pull-up resistor option register 2
PM2: Port 2 mode register
RD: Read signal of port 2
WR: Write signal of port 2

2. n = 6, 7

14.2.4 Port 3

Port 3 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. A pull-up resistor can be connected in 1-bit units (software pull-up function).

When P36 and P37 are used as the TI4 and TI5 pins, noise is eliminated from these pins by a digital noise eliminator.

After reset: 00H R/W Address: FFFFF006H

	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Output 0
1	Output 1

Remark In input mode: When port 3 (P3) is read, the pin levels at that time are read. Writing to P3 writes the values to that register. This does not affect the input pins.

In output mode: When port 3 (P3) is read, the P3 values are read. Writing to P3 writes the values to that register, and those values are immediately output.

Port 3 includes the following alternate functions.

Table 14-5. Alternate Functions of Port 3

Pin Name		Alternate Function	I/O	PULL [Note]	Remark
Port 3	P30	TI00	I/O	Yes	–
	P31	TI01			
	P32	TI10			
	P33	TI11			
	P34	TO0/A13			
	P35	TO1/A14			
	P36	TI4/TO4/A15			Digital noise elimination
	P37	TI5/TO5			

Note Software pull-up function

(1) Function of P3 pins

Port 3 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 3 mode register (PM3).

In output mode, the values set to each bit are output to port 3 (P3).

When using this port in input mode, the pin statuses can be read by reading P3. Also, the P3 (output latch) values can be read by reading P3 while in output mode.

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 3 (PU3).

When using the alternate-function TI4 and TI5 pins, noise elimination is provided by a digital noise eliminator (same as digital noise eliminator for port 0).

When using the alternate-function A13 to A15 pins, set the pin functions via the memory address output mode register (MAM). At this time, be sure to set PM3 (PM34 to PM36) to 0.

Clear P3 and PM3 to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 3 mode register (PM3)

PM3 can be read/written in 1-bit or 8-bit units.

After reset: FFH R/W Address: FFFFF026H

	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

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

PU3 can be read/written in 1-bit or 8-bit units.

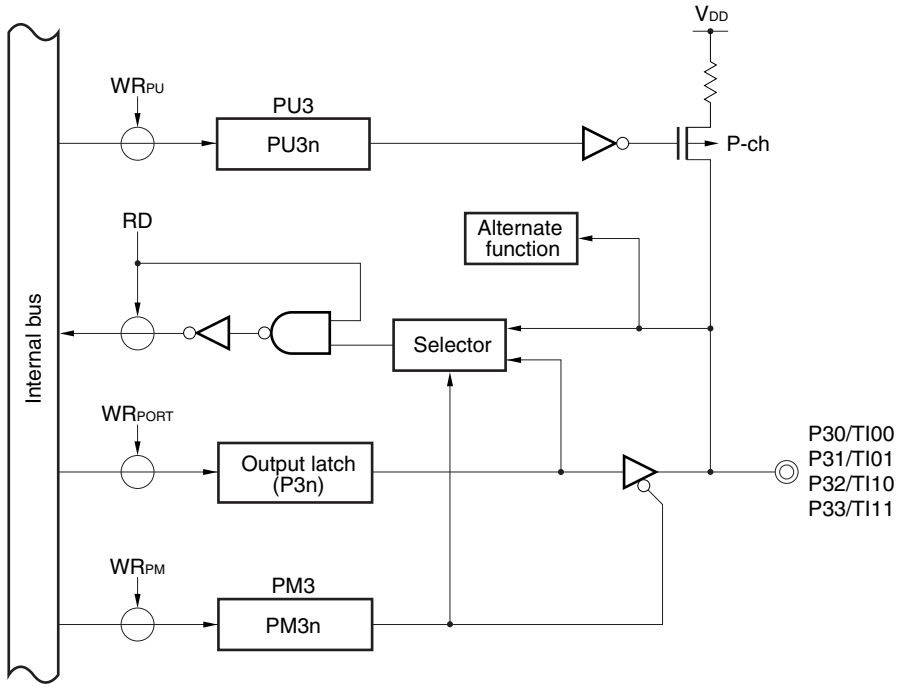
After reset: 00H R/W Address: FFFFF086H

	7	6	5	4	3	2	1	0
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(3) Block diagrams (port 3)

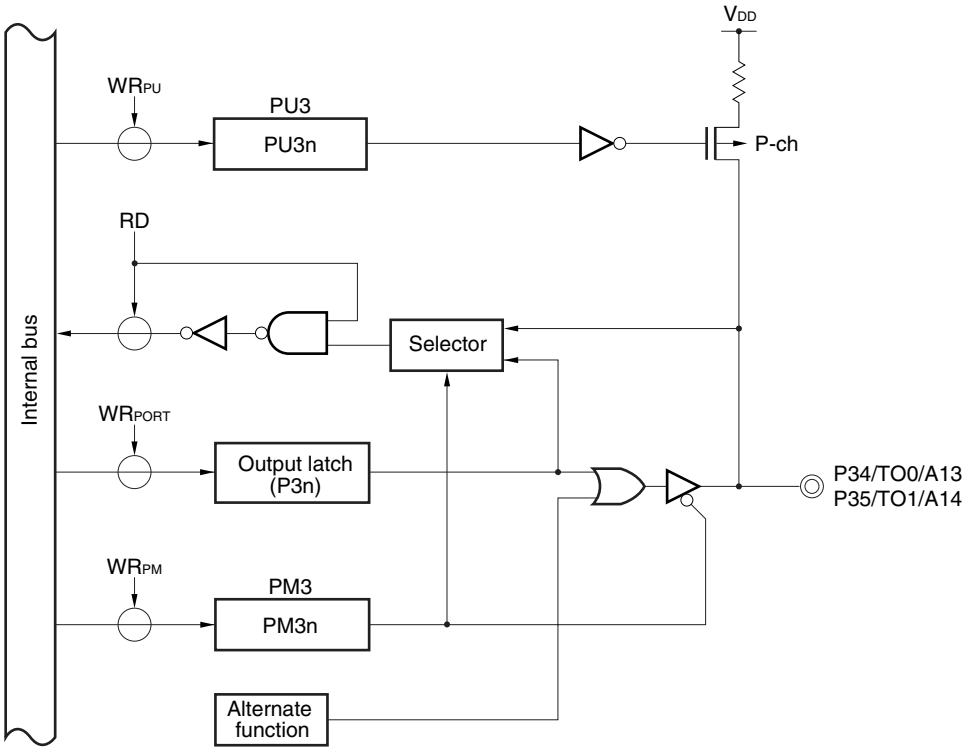
Figure 14-10. Block Diagram of P30 to P33



Remarks 1. PU3: Pull-up resistor option register 3
PM3: Port 3 mode register
RD: Read signal of port 3
WR: Write signal of port 3

2. n = 0 to 3

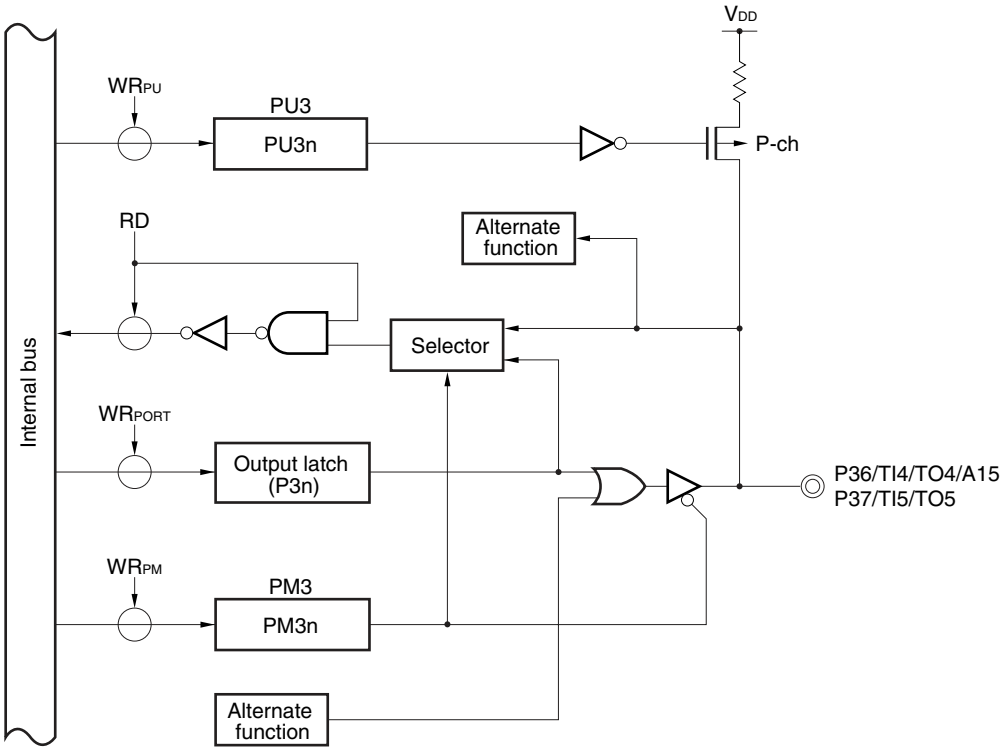
Figure 14-11. Block Diagram of P34 and P35



Remarks 1. PU3: Pull-up resistor option register 3
PM3: Port 3 mode register
RD: Read signal of port 3
WR: Write signal of port 3

2. n = 4, 5

Figure 14-12. Block Diagram of P36 and P37



Remarks 1. PU3: Pull-up resistor option register 3
PM3: Port 3 mode register
RD: Read signal of port 3
WR: Write signal of port 3

2. n = 6, 7

14.2.5 Ports 4 and 5

Ports 4 and 5 are 8-bit I/O ports for which I/O settings can be controlled in 1-bit units.

After reset: 00H R/W Address: FFFFF008H, FFFFF00AH

	7	6	5	4	3	2	1	0
Pn (n = 4, 5)	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnx	Control of Output Data (in Output Mode) (n = 4, 5, x = 0 to 7)
0	Output 0
1	Output 1

Remark In input mode: When port 4 (P4) and port 5 (P5) are read, the pin levels at that time are read. Writing to P4 and P5 writes the values to those registers. This does not affect the input pins.

In output mode: When port 4 (P4) and port 5 (P5) are read, their values are read. Writing to P4 and P5 writes the values to those registers, and those values are immediately output.

Ports 4 and 5 include the following alternate functions.

Table 14-6. Alternate Functions of Ports 4 and 5

Pin Name		Alternate Function	I/O	PULL [Note]	Remark
Port 4	P40	AD0	I/O	No	–
	P41	AD1			
	P42	AD2			
	P43	AD3			
	P44	AD4			
	P45	AD5			
	P46	AD6			
	P47	AD7			
Port 5	P50	AD8	I/O	No	–
	P51	AD9			
	P52	AD10			
	P53	AD11			
	P54	AD12			
	P55	AD13			
	P56	AD14			
	P57	AD15			

Note Software pull-up function

(1) Functions of P4 and P5 pins

Ports 4 and 5 are 8-bit I/O ports for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 4 mode register (PM4) and port 5 mode register (PM5).

In output mode, the values set to each bit are output to ports 4 and 5 (P4 and P5).

When using these ports in input mode, the pin statuses can be read by reading P4 and P5. Also, the P4 and P5 (output latch) values can be read by reading P4 and P5 while in output mode.

A software pull-up function is not implemented.

When using the alternate-function AD0 to AD15 pins, set the pin functions via the memory expansion mode register (MM). This does not affect PM4 and PM5.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 4 mode register and port 5 mode register (PM4 and PM5)

PM4 and PM5 can be read/written in 1-bit or 8-bit units.

After reset: FFH R/W Address: FFFFF028H, FFFFF02AH

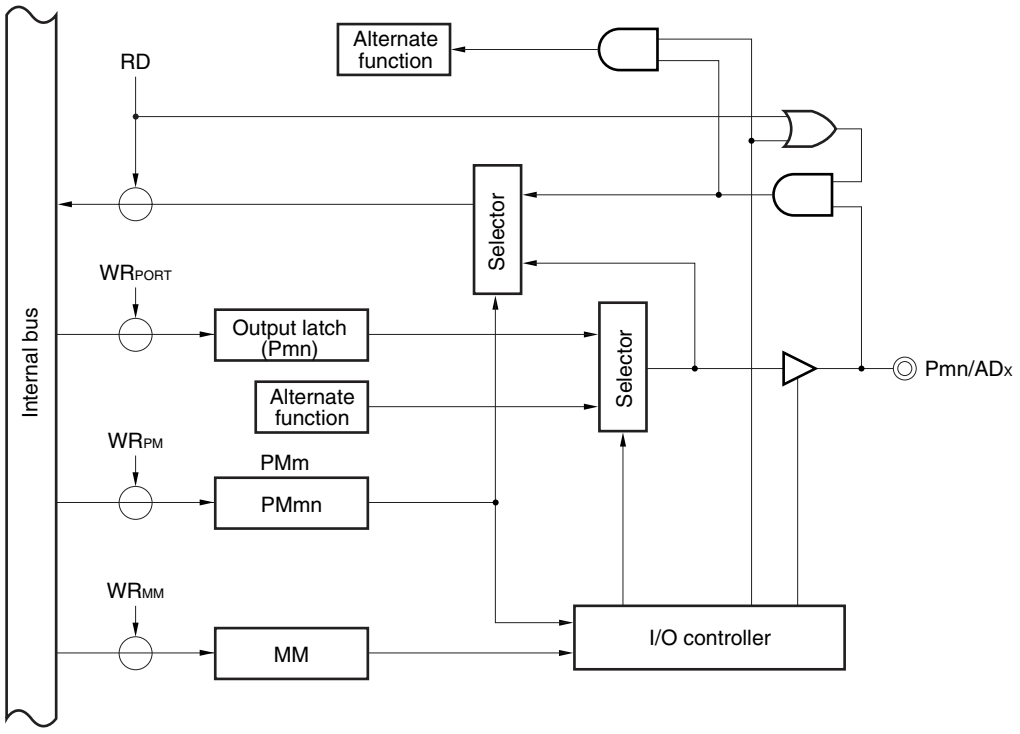
	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

(n = 4, 5)

PMnx	Control of I/O Mode (n = 4, 5, x = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagram (port 4, port 5)

Figure 14-13. Block Diagram of P40 to P47 and P50 to P57



Remarks 1. PM4: Port 4 mode register
PM5: Port 5 mode register
MM: Memory expansion mode register
RD: Read signals of ports 4 and 5
WR: Write signals of ports 4 and 5

2. m = 4, 5
n = 0 to 7
x = 0 to 15

14.2.6 Port 6

Port 6 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units.

After reset: 00H R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
P6	0	0	P65	P64	P63	P62	P61	P60

P6n	Control of Output Data (in Output Mode) (n = 0 to 5)
0	Output 0
1	Output 1

Remark In input mode: When port 6 (P6) is read, the pin levels at that time are read. Writing to P6 writes the values to that register. This does not affect the input pins.

In output mode: When port 6 (P6) is read, the P6 values are read. Writing to P6 writes the values to that register, and those values are immediately output.

Port 6 includes the following alternate functions.

Table 14-7. Alternate Functions of Port 6

Pin Name		Alternate Function	I/O	PULL Note	Remark
Port 6	P60	A16	I/O	No	–
	P61	A17			
	P62	A18			
	P63	A19			
	P64	A20			
	P65	A21			

Note Software pull-up function

(1) Function of P6 pins

Port 6 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 6 mode register (PM6).

In output mode, the values set to each bit are output to port 6 (P6).

When using this port in input mode, the pin statuses can be read by reading P6. Also, the P6 (output latch) values can be read by reading P6 while in output mode.

A software pull-up function is not implemented.

When using the alternate-function A16 to A21 pins, set the pin functions via the memory expansion mode register (MM). This does not affect PM6.

When a reset is input, the settings are initialized to input mode.

(2) Control register

(a) Port 6 mode register (PM6)

PM6 can be read/written in 1-bit or 8-bit units.

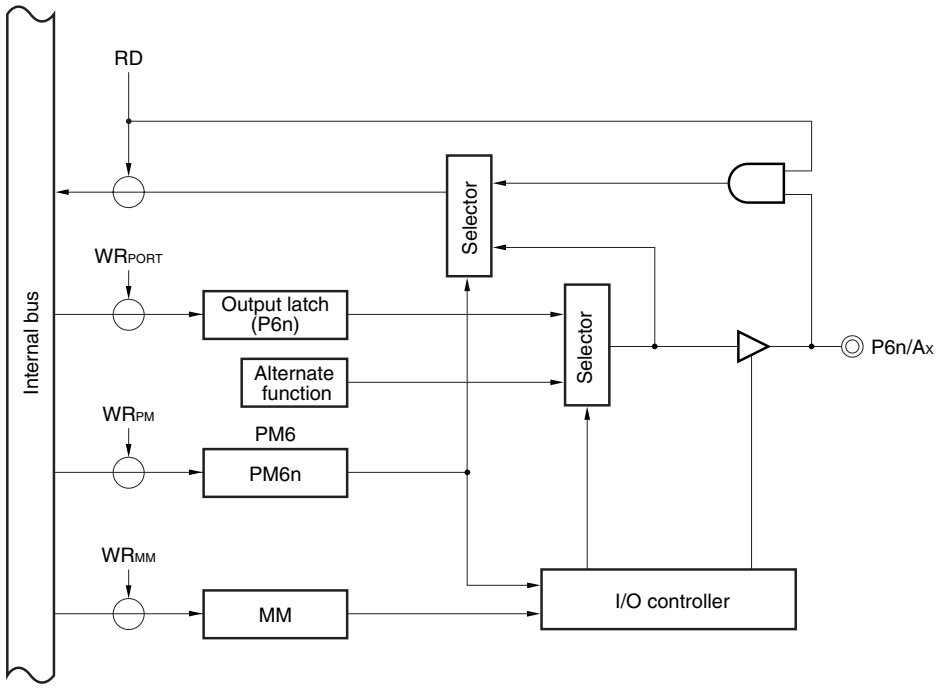
After reset: 3FH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PM6	0	0	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	Control of I/O Mode (n = 0 to 5)
0	Output mode
1	Input mode

(3) Block diagram (port 6)

Figure 14-14. Block Diagram of P60 to P65



Remarks **1.** PM6: Port 6 mode register
MM: Memory expansion mode register
RD: Read signal of port 6
WR: Write signal of port 6

2. n = 0 to 5
x = 16 to 21

14.2.7 Ports 7 and 8

Port 7 is an 8-bit input-only port and port 8 is a 4-bit input-only port. Both ports are read-only and are accessible in 8-bit or 1-bit units.

After reset: Undefined R Address: FFFFF00EH

	7	6	5	4	3	2	1	0
P7	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Pin Level (n = 0 to 7)
0/1	Read pin level of bit n

After reset: Undefined R Address: FFFFF010H

	7	6	5	4	3	2	1	0
P8	0	0	0	0	P83	P82	P81	P80

P8n	Pin Level (n = 0 to 3)
0/1	Read pin level of bit n

Ports 7 and 8 include the following alternate functions.

Table 14-8. Alternate Functions of Ports 7 and 8

Pin Name		Alternate Function	I/O	PULL [Note]	Remark
Port 7	P70	ANI0	Input	No	–
	P71	ANI1			
	P72	ANI2			
	P73	ANI3			
	P74	ANI4			
	P75	ANI5			
	P76	ANI6			
	P77	ANI7			
Port 8	P80	ANI8	Input	No	–
	P81	ANI9			
	P82	ANI10			
	P83	ANI11			

Note Software pull-up function

(1) Functions of P7 and P8 pins

Port 7 is an 8-bit input-only port and port 8 is a 4-bit input-only port.

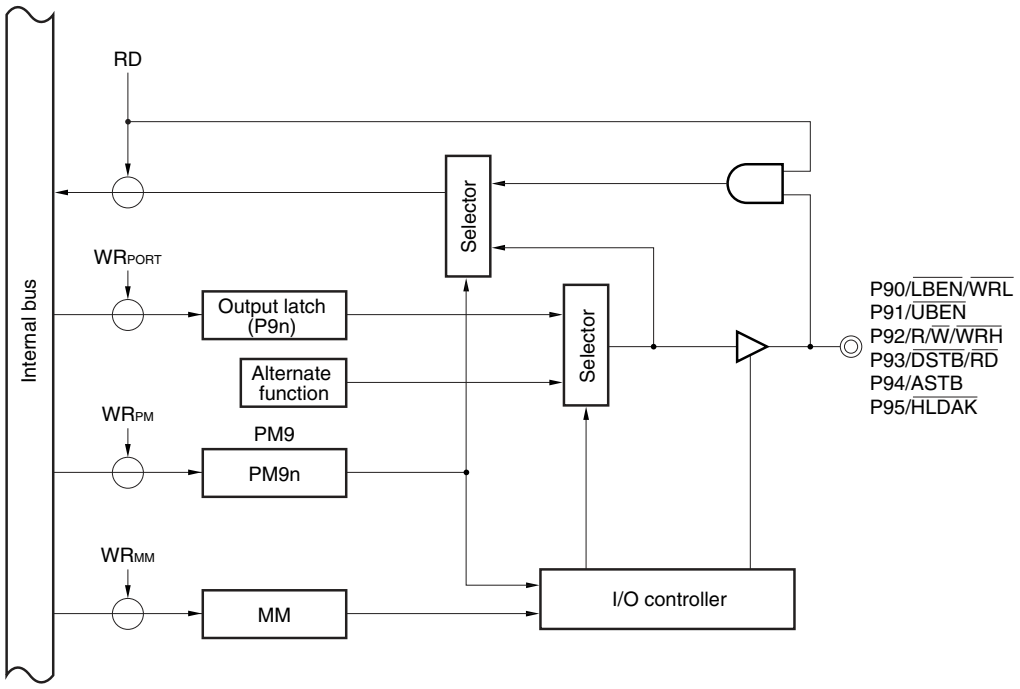
The pin statuses can be read by reading ports 7 and 8 (P7 and P8). Data cannot be written to P7 or P8.

A software pull-up function is not implemented.

Values read from pins specified as analog inputs are undefined values. Do not read values from P7 or P8 during A/D conversion.

(2) Block diagram (port 7, port 8)

Figure 14-15. Block Diagram of P70 to P77 and P80 to P83



Remarks 1. RD: Read signals of ports 7 and 8

2. m = 7, 8

n = 0 to 7 (when m = 7), 0 to 3 (when m = 8)

x = 0 to 7 (when m = 7), 8 to 11 (when m = 8)

14.2.8 Port 9

Port 9 is a 7-bit I/O port for which I/O settings can be controlled in 1-bit units.

After reset: 00H R/W Address: FFFFF012H

	7	6	5	4	3	2	1	0
P9	0	P96	P95	P94	P93	P92	P91	P90

P9n	Control of Output Data (in Output Mode) (n = 0 to 6)
0	Output 0
1	Output 1

Remark In input mode: When port 9 (P9) is read, the pin levels at that time are read. Writing to P9 writes the values to that register. This does not affect the input pins.

In output mode: When port 9 (P9) is read, the P9 values are read. Writing to P9 writes the values to that register, and those values are immediately output.

Port 9 includes the following alternate functions.

Table 14-9. Alternate Functions of Port 9

Pin Name		Alternate Function	I/O	PULL Note	Remark
Port 9	P90	$\overline{LBEN}/\overline{WRL}$	I/O	No	–
	P91	$\overline{UBEN}$			
	P92	R/$\overline{W}$/$\overline{WRH}$			
	P93	$\overline{DSTB}$/$\overline{RD}$			
	P94	ASTB			
	P95	$\overline{HLDAK}$			
	P96	$\overline{HLDRQ}$			

Note Software pull-up function

(1) Function of P9 pins

Port 9 is a 7-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 9 mode register (PM9).

In output mode, the values set to each bit are output to port 9 (P9).

When using this port in input mode, the pin statuses can be read by reading P9. Also, the P9 (output latch) values can be read by reading P9 while in output mode.

A software pull-up function is not implemented.

When using the alternate-function external expansion function pins, set the pin functions via the memory expansion mode register (MM).

When a reset is input, the settings are initialized to input mode.

★ **Caution When using port 9 as an I/O port, set the BIC bit of the system control register (SYC) to 0. After the system is reset, the BIC bit is 0.**

(2) Control register

(a) Port 9 mode register (PM9)

PM9 can be read/written in 1-bit or 8-bit units.

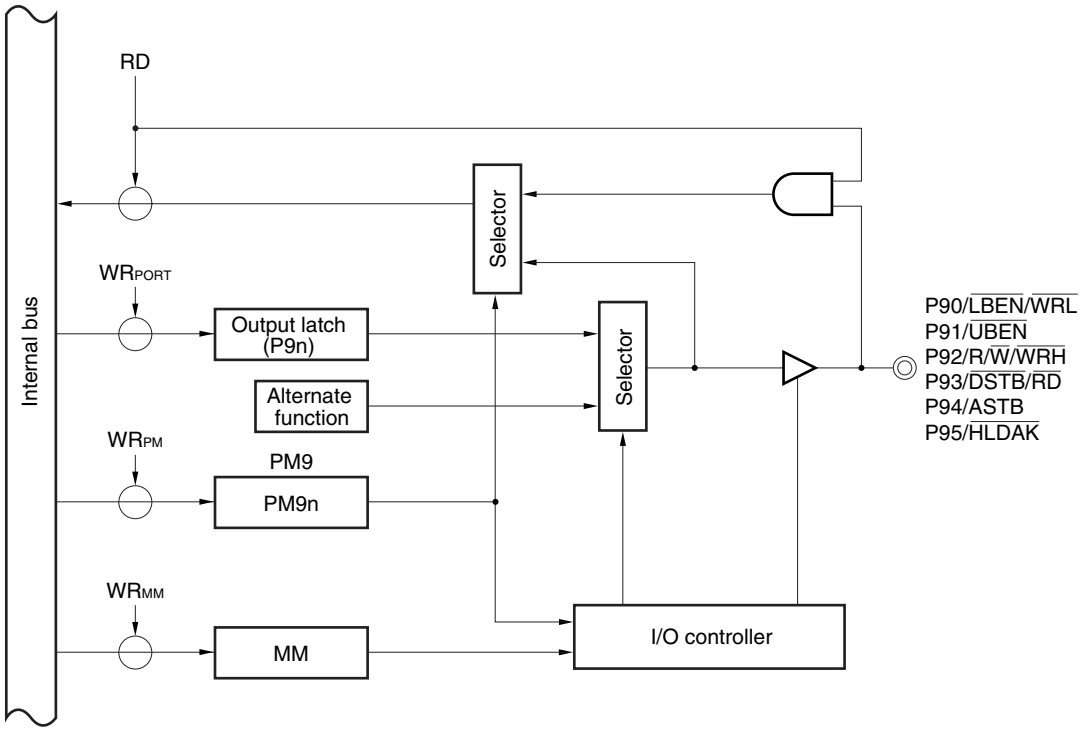
After reset: 7FH R/W Address: FFFFF032H

	7	6	5	4	3	2	1	0
PM9	0	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	Control of I/O Mode (n = 0 to 6)
0	Output mode
1	Input mode

(3) Block diagrams (port 9)

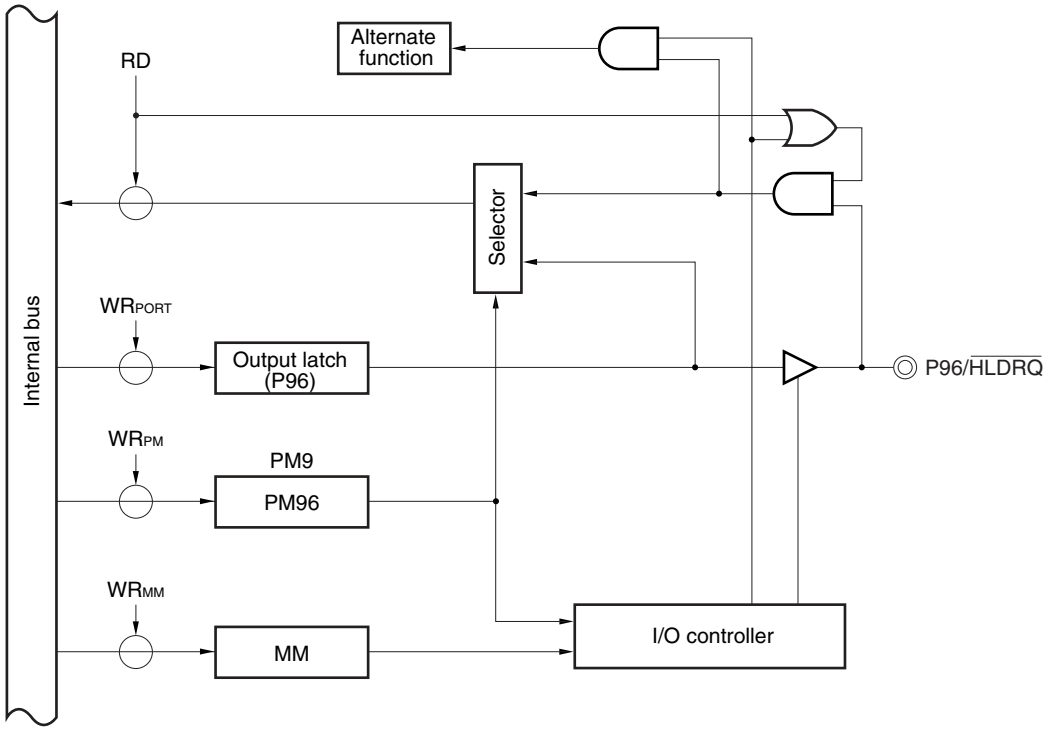
Figure 14-16. Block Diagram of P90 to P95



Remarks 1. PM9: Port 9 mode register
MM: Memory expansion mode register
RD: Read signal of port 9
WR: Write signal of port 9

2. n = 0 to 5

Figure 14-17. Block Diagram of P96



Remark PM9: Port 9 mode register
MM: Memory expansion mode register
RD: Read signal of port 9
WR: Write signal of port 9

14.2.9 Port 10

Port 10 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. A pull-up resistor can be connected in 1-bit units (software pull-up function).

The pins in this port are selectable as normal outputs or N-ch open-drain outputs.

After reset: 00H R/W Address: FFFFF014H

	7	6	5	4	3	2	1	0
P10	P107	P106	P105	P104	P103	P102	P101	P100

P10n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Output 0
1	Output 1

Remark In input mode: When port 10 (P10) is read, the pin levels at that time are read. Writing to P10 writes the values to that register. This does not affect the input pins.

In output mode: When port 10 (P10) is read, the P10 values are read. Writing to P10 writes the values to that register, and those values are immediately output.

Port 10 includes the following alternate functions.

Table 14-10. Alternate Functions of Port 10

Pin Name		Alternate Function	I/O	PULL [Note]	Remark
Port 10	P100	RTP0/A5	I/O	Yes	Selectable as N-ch open-drain outputs
	P101	RTP1/A6			
	P102	RTP2/A7			
	P103	RTP3/A8			
	P104	RTP4/A9			
	P105	RTP5/A10			
	P106	RTP6/A11			
	P107	RTP7/A12			

Note Software pull-up function

(1) Function of P10 pins

Port 10 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 10 mode register (PM10).

In output mode, the values set to each bit are output to port 10 (P10). The port 10 function register (PF10) can be used to specify whether outputs are normal outputs or N-ch open-drain outputs.

When using this port in input mode, the pin statuses can be read by reading P10. Also, the P10 (output latch) values can be read by reading P10 while in output mode.

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 10 (PU10).

When using the alternate-function A5 to A12 pins, set the pin functions via the memory address output mode register (MAM). At this time, be sure to set PM10 to 0.

When using alternate-function pins as outputs, the ORed result of the port output and the alternate-function pin is output from the pins.

When a reset is input, the settings are initialized to input mode.

Caution When using port 10 as a real-time output port, set in accordance with 13. 5 Usage.

(2) Control registers

(a) Port 10 mode register (PM10)

PM10 can be read/written in 1-bit or 8-bit units.

After reset: FFH R/W Address: FFFFF034H

	7	6	5	4	3	2	1	0
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100

PM10n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

(b) Pull-up resistor option register 10 (PU10)

PU10 can be read/written in 1-bit or 8-bit units.

After reset: 00H R/W Address: FFFFF094H

	7	6	5	4	3	2	1	0
PU10	PU107	PU106	PU105	PU104	PU103	PU102	PU101	PU100

PU10n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(c) Port 10 function register (PF10)

PF10 can be read/written in 1-bit or 8-bit units.

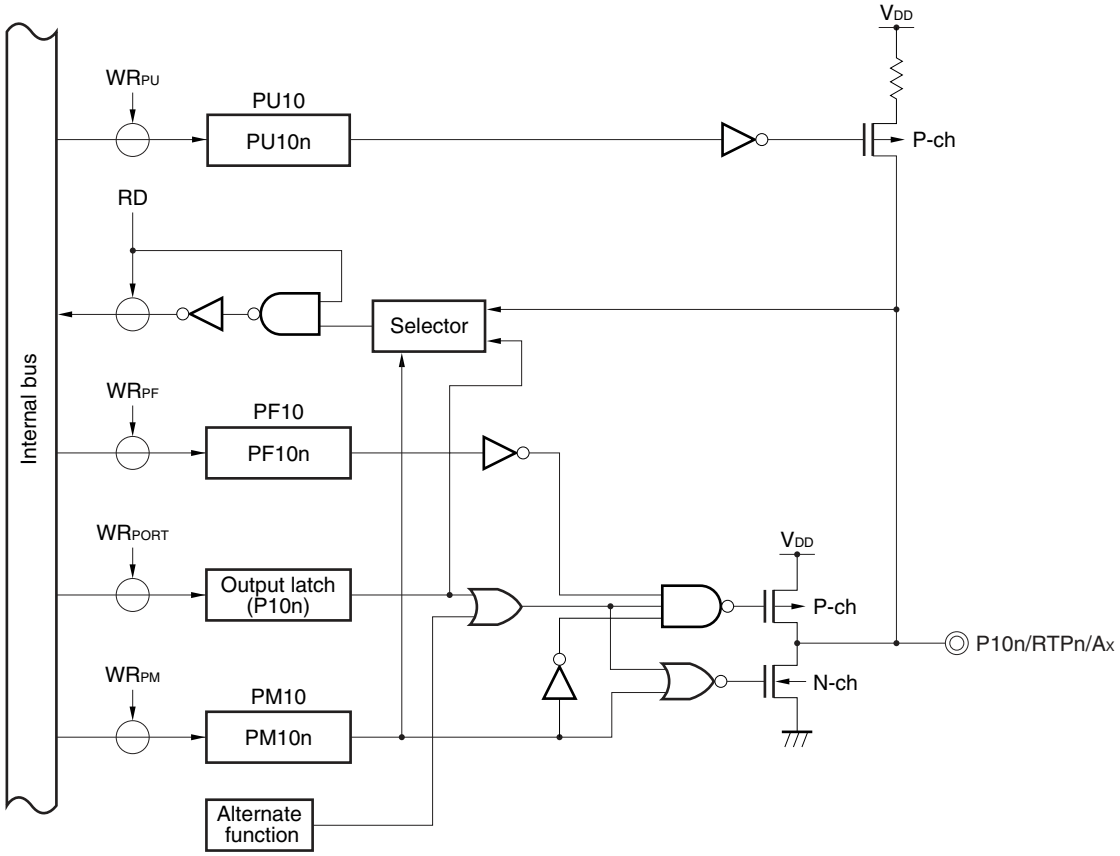
After reset: 00H R/W Address: FFFFF0B4H

	7	6	5	4	3	2	1	0
PF10	PF107	PF106	PF105	PF104	PF103	PF102	PF101	PF100

PF10n	Control of Normal Output/N-ch Open-Drain Output (n = 0 to 7)
0	Normal output
1	N-ch open-drain output

(3) Block diagram (port 10)

Figure 14-18. Block Diagram of P100 to P107



Remarks 1. PU10: Pull-up resistor option register 10
PF10: Port 10 function register
PM10: Port 10 mode register
RD: Read signal of port 10
WR: Write signal of port 10

2. n = 0 to 7
x = 5 to 12

14.2.10 Port 11

Port 11 includes P114, which is an input-only port, and P110 to P113, which comprise an I/O port for which I/O settings can be controlled in 1-bit units. A pull-up resistor can be connected to bits 0 to 3 in 1-bit units (software pull-up function).

P11 can be read/written in 1-bit or 8-bit units. However, bit 4 can only be read.

After reset: 00H R/W Address: FFFFF016H

	7	6	5	4	3	2	1	0
P11	0	0	0	P114	P113	P112	P111	P110

P11n	Control of Output Data (in Output Mode) (n = 0 to 3)
0	Output 0
1	Output 1

P114	Read Pin Level
0/1	Read P114 pin level

Remark In input mode: When port 11 (P11) is read, the pin levels at that time are read. Writing to P11 writes the values to that register. This does not affect the input pins.

In output mode: When port 11 (P11) is read, the P11 values are read. Writing to P11 writes the values to that register, and those values are immediately output (except P114).

Port 11 includes the following alternate functions.

Table 14-11. Alternate Functions of Port 11

Pin Name		Alternate Function	I/O	PULL [Note]	Remark
Port 11	P110	A1	I/O	Yes	–
	P111	A2			
	P112	A3			
	P113	A4			
	P114	XT1	Input	No	Also used as subclock (XT1)

Note Software pull-up function

(1) Function of P11 pins

Port 11 is a 5-bit (total) port that includes P114, which is an input-only port, and P110 to P113, which comprise an I/O port for which I/O settings can be controlled in 1-bit units.

In output mode, the values set to each bit (bit 0 to bit 3) are output to port 11 (P11).

When using this port in input mode, the pin statuses can be read by reading P11. Also, the P11 (output latch) values can be read by reading P11 while in output mode (bit 0 to bit 3 only).

A pull-up resistor can be connected in 1-bit units for P110 to P113 when specified via pull-up resistor option register 11 (PU11).

When using the alternate-function A1 to A4 pins, set the pin functions via the memory address output mode register (MAM). At this time, be sure to set PM11 (PM110 to PM113) to 0.

When a reset is input, the settings are initialized to input mode.

Caution Because the P114/XT1 pin is internally connected to the XT2 pin via a circuit, the P114/XT1 and XT2 pins will interfere with each other, even when the subclock is not being used. Therefore, leave the XT2 pin open when not using the subclock.

(2) Control registers

(a) Port 11 mode register (PM11)

PM11 can be read/written in 1-bit or 8-bit units.

After reset: 1FH R/W Address: FFFFF036H

	7	6	5	4	3	2	1	0
PM11	0	0	0	1	PM113	PM112	PM111	PM110

PM11n	Control of I/O Mode (n = 0 to 3)
0	Output mode
1	Input mode

(b) Pull-up resistor option register 11 (PU11)

PU11 can be read/written in 1-bit or 8-bit units.

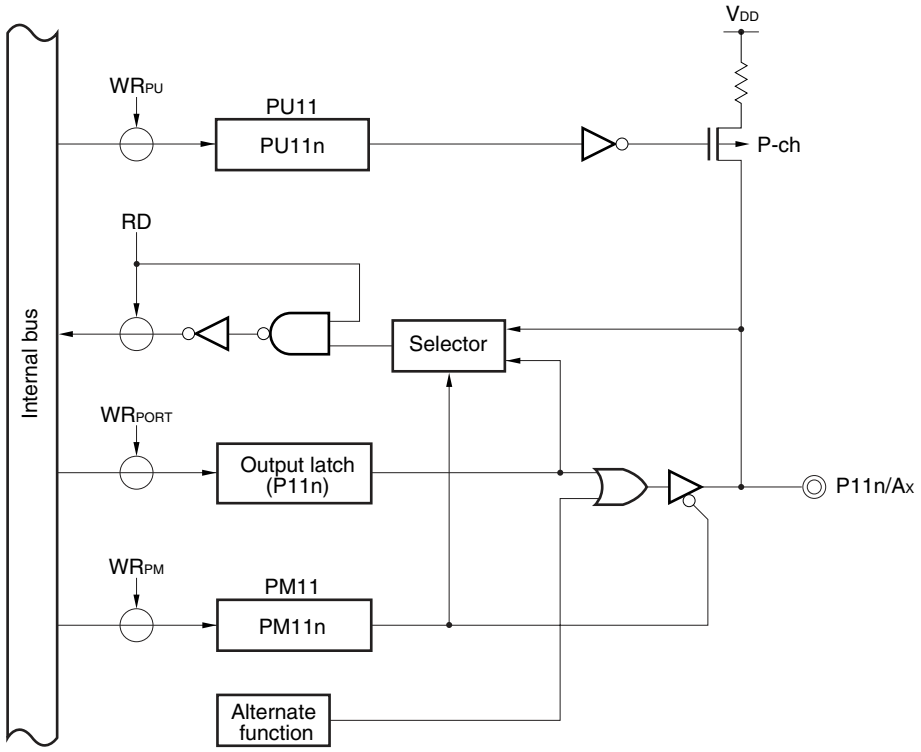
After reset: 00H R/W Address: FFFFF096H

	7	6	5	4	3	2	1	0
PU11	0	0	0	0	PU113	PU112	PU111	PU110

PU11n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 3)
0	Do not connect
1	Connect

(3) Block diagrams (port 11)

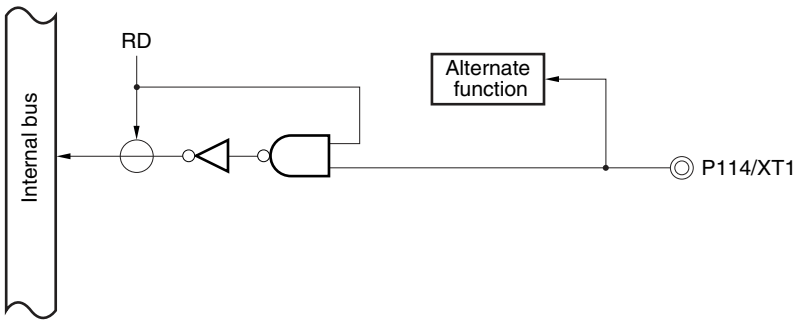
Figure 14-19. Block Diagram of P110 to P113



Remarks 1. PU11: Pull-up resistor option register 11
PM11: Port 11 mode register
RD: Read signal of port 11
WR: Write signal of port 11

2. n = 0 to 3
x = 1 to 4

Figure 14-20. Block Diagram of P114



Remark RD: Read signal of port 11

14.2.11 Port 12

Port 12 is a 1-bit I/O port.

After reset: 00H R/W Address: FFFFF018H

	7	6	5	4	3	2	1	0
PU12	0	0	0	0	0	0	0	P120

P120	Control of Output Data (in Output Mode)
0	Output 0
1	Output 1

Remark In input mode: When port 12 (P12) is read, the pin levels at that time are read. Writing to P12 writes the values to that register. This does not affect the input pins.
In output mode: When port 12 (P12) is read, the P12 values are read. Writing to P12 writes the values to that register, and those values are immediately output.

Port 12 includes the following alternate function.

Table 14-12. Alternate Function of Port 12

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 12	P120	$\overline{\text{WAIT}}$	I/O	No	–

Note Software pull-up function

(1) Function of P12 pin

Port 12 is a 1-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 12 mode register (PM12).

In output mode, the set value is output to port 12 (P12).

When using this port in input mode, the pin status can be read by reading P12. Also, the P12 (output latch) value can be read by reading P12 while in output mode.

When using the alternate-function $\overline{\text{WAIT}}$ pin, set the pin function via the port 12 mode control register (PMC12).

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 12 mode register (PM12)

PM12 can be read/written in 1-bit or 8-bit units.

After reset: 01H R/W Address: FFFFF038H

	7	6	5	4	3	2	1	0
PM12	0	0	0	0	0	0	0	PM120

P120	Control of Input Mode
0	Output mode
1	Input mode

(b) Port 12 mode control register (PMC12)

PMC12 can be read/written in 1-bit or 8-bit units.

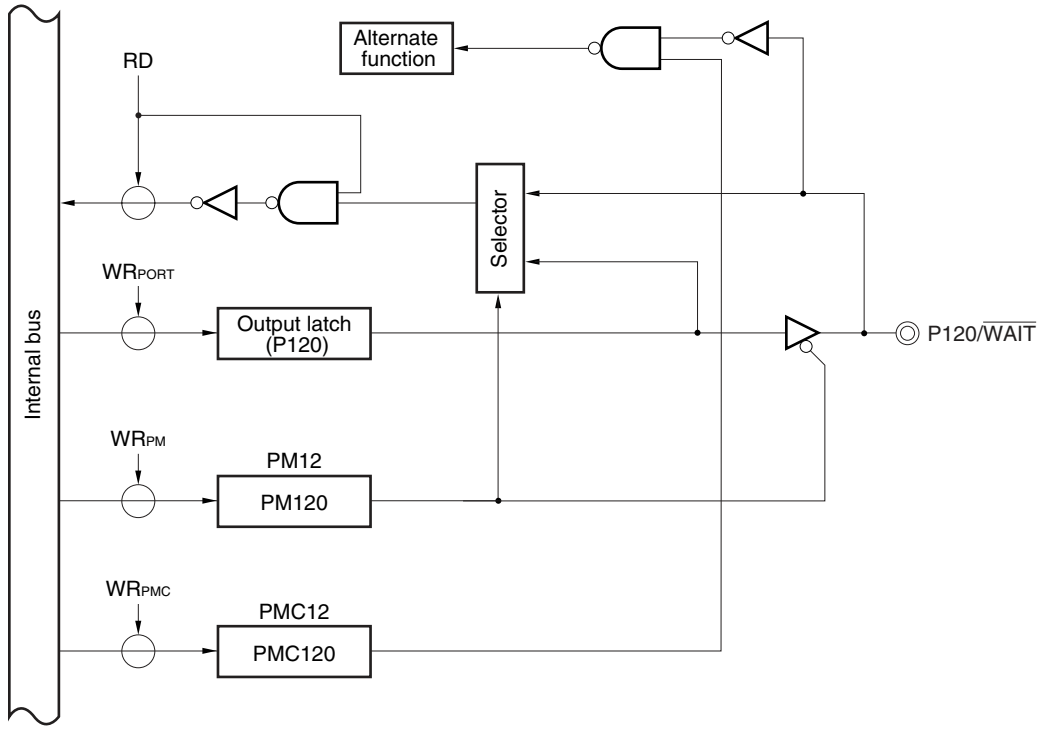
After reset: 00H R/W Address: FFFFF058H

	7	6	5	4	3	2	1	0
PMC12	0	0	0	0	0	0	0	PMC120

PMC120	Switching of Alternate Function
0	Use as port mode
1	$\overline{\text{Use as WAIT pin}}$

(3) Block diagram (port 12)

Figure 14-21. Block Diagram of P120



Remark PM12: Port 12 mode register
PMC12: Port 12 mode control register
RD: Read signal of port 12
WR: Write signal of port 12

14.3 Setting When Port Pin Is Used as Alternate Function

When a port pin is used for an alternate function, set the port n mode register (PM0 to PM6 and PM9 to PM12) and output latch as shown below.

Table 14-13. Setting When Port Pin Is Used for Alternate Function (1/3)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P00	NMI	Input	PM00 = 1	Setting not needed for P00	—
P01	INTP0	Input	PM01 = 1	Setting not needed for P01	—
P02	INTP1	Input	PM02 = 1	Setting not needed for P02	—
P03	INTP2	Input	PM03 = 1	Setting not needed for P03	—
P04	INTP3	Input	PM04 = 1	Setting not needed for P04	—
P05	INTP4	Input	PM05 = 1	Setting not needed for P05	—
	ADTRG	Input			
P06	INTP5	Input	PM06 = 1	Setting not needed for P06	—
	RTPTRG	Input			
P07	INTP6	Input	PM07 = 1	Setting not needed for P07	—
P10	SI0	Input	PM10 = 1	Setting not needed for P10	—
	SDA[Note]	I/O	PM10 = 0	P10 = 0	PF10 = 1
P11	SO0	Output	PM11 = 0	P11 = 0	—
P12	$\overline{\text{SCK0}}$	Input	PM12 = 1	Setting not needed for P12	—
		Output	PM12 = 0	P12 = 0	
	SCL[Note]	I/O			PF12 = 1
P13	SI1	Input	PM13 = 1	Setting not needed for P13	—
	RXD0	Input			
P14	SO1	Output	PM14 = 0	P14 = 0	—
	TXD0	Output			
P15	$\overline{\text{SCK1}}$	Input	PM15 = 1	Setting not needed for P15	—
		Output	PM15 = 0	P15 = 0	
	ASCK0	Input	PM15 = 1	Setting not needed for P15	

★ **Note** μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY only

Table 14-13. Setting When Port Pin Is Used as Alternate Function (2/3)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P20	SI2	Input	PM20 = 1	Setting not needed for P20	—
P21	SO2	Output	PM21 = 0	P21 = 0	—
P22	$\overline{SCK2}$	Input	PM22 = 1	Setting not needed for P22	—
		Output	PM22 = 0	P22 = 0	
P23	RXD1	Input	PM23 = 1	Setting not needed for P23	—
P24	TXD1	Output	PM24 = 0	P24 = 0	—
P25	ASCK1	Input	PM25 = 1	Setting not needed for P25	—
P26	TI2	Input	PM26 = 1	Setting not needed for P26	—
	TO2	Output	PM26 = 0	P26 = 0	
P27	TI3	Input	PM27 = 1	Setting not needed for P27	—
	TO3	Output	PM27 = 0	P27 = 0	
P30	TI00	Input	PM30 = 1	Setting not needed for P30	—
P31	TI01	Input	PM31 = 1	Setting not needed for P31	—
P32	TI10	Input	PM32 = 1	Setting not needed for P32	—
P33	TI11	Input	PM33 = 1	Setting not needed for P33	—
P34	TO0	Output	PM34 = 0	P34 = 0	—
	A13	Output			Refer to **3.4.6 (2)** (MAM)
P35	TO1	Output	PM35 = 0	P35 = 0	—
	A14	Output			Refer to **3.4.6 (2)** (MAM)
P36	TI4	Input	PM36 = 1	Setting not needed for P36	—
	TO4	Output	PM36 = 0	P36 = 0	
	A15	Output			Refer to **3.4.6 (2)** (MAM)
P37	TI5	Input	PM37 = 1	Setting not needed for P37	—
	TO5	Output	PM37 = 0	P37 = 0	
P40 to P47	AD0 to AD7	I/O	Setting not needed for PM40 to PM47	Setting not needed for P40 to P47	Refer to **3.4.6 (1)** (MM)

Note μPD703035Y and 70F3035Y only

Table 14-13. Setting When Port Pin Is Used as Alternate Function (3/3)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P50 to P57	AD8 to AD15	I/O	Setting not needed for PM50 to PM57	Setting not needed for P50 to P57	Refer to **3.4.6 (1)** (MM)
P60 to P65	A16 to A21	Output	Setting not needed for PM60 to PM65	Setting not needed for P60 to P65	Refer to **3.4.6(1)** (MM)
P70 to P77	ANI0 to ANI7	Input	None	Setting not needed for P70 to P77	—
P80 to P83	ANI8 to ANI11	Input	None	Setting not needed for P80 to P83	—
P90	$\overline{\text{LBEN}}$	Output	Setting not needed for PM90	Setting not needed for P90	Refer to **3.4.6 (1)** (MM)
	$\overline{\text{WRL}}$	Output			
P91	$\overline{\text{UBEN}}$	Output	Setting not needed for PM91	Setting not needed for P91	Refer to **3.4.6 (1)** (MM)
P92	R/$\overline{\text{W}}$	Output	Setting not needed for PM92	Setting not needed for P92	Refer to **3.4.6 (1)** (MM)
	$\overline{\text{WRH}}$	Output			
P93	$\overline{\text{DSTB}}$	Output	Setting not needed for PM93	P93 = 1	Refer to **3.4.6 (1)** (MM)
	$\overline{\text{RD}}$	Output			
P94	ASTB	Output	Setting not needed for PM94	P94 = 1	Refer to **3.4.6 (1)** (MM)
P95	$\overline{\text{HLDAK}}$	Output	Setting not needed for PM95	Setting not needed for P95	Refer to **3.4.6 (1)** (MM)
P96	$\overline{\text{HLDRQ}}$	Input	Setting not needed for PM96	Setting not needed for P96	Refer to **3.4.6 (1)** (MM)
P100 to P107	RTP0 to RTP7	Output	PM100 to PM107 = 0	P100 to P107 = 0	–
	A5 to A12	Output			Refer to **3.4.6 (2)** (MAM)
P110 to P113	A1 to A4	Output	PM110 to PM113 = 0	P110 to P113 = 0	Refer to **3.4.6 (2)** (MAM)
P114	XT1	Input	None	Setting not needed for P114	–
P120	$\overline{\text{WAIT}}$	Input	PM120 = 1	Setting not needed for P120	PMC120 = 1 (PMC12)

Caution When changing the output level of port 0 by setting the of port function output mode of port 0, the interrupt request flag will be set because port 0 also has an alternate function as an external interrupt request input. Therefore, be sure to set the corresponding interrupt mask flag to 1 before using the output mode.

mark PMnx bit of PMn register and Pnx bit of Pn

n: 0 (x = 0 to 7) n: 1 (x = 0 to 5) n: 2 (x = 0 to 7) n: 3 (x = 0 to 7) n: 4 (x = 0 to 7)
n: 5 (x = 0 to 7) n: 6 (x = 0 to 5) n: 7 (x = 0 to 7) n: 8 (x = 0 to 3) n: 9 (x = 0 to 6)
n: 10 (x = 0 to 7) n: 11 (x = 0 to 4) n: 12 (x = 0)

⋆ 14.4 Operation of Port Function

The operation of a port differs depending on whether the port is in the input or output mode, as described below.

14.4.1 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 bit manipulation instruction (CLR1, SET1, NOT1) 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, are overwritten to the current input pin status and become undefined.

14.4.2 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 15 RESET FUNCTION

15.1 General

When a low level is input to the $\overline{\text{RESET}}$ pin, a system reset is performed and the various on-chip hardware devices are reset to their initial settings. In addition, oscillation of the main clock is stopped during the reset period, although oscillation of the subclock continues.

When the input at the $\overline{\text{RESET}}$ pin changes from low level to high level, the reset status is released and the CPU resumes program execution. The contents of the various registers should be initialized within the program as necessary.

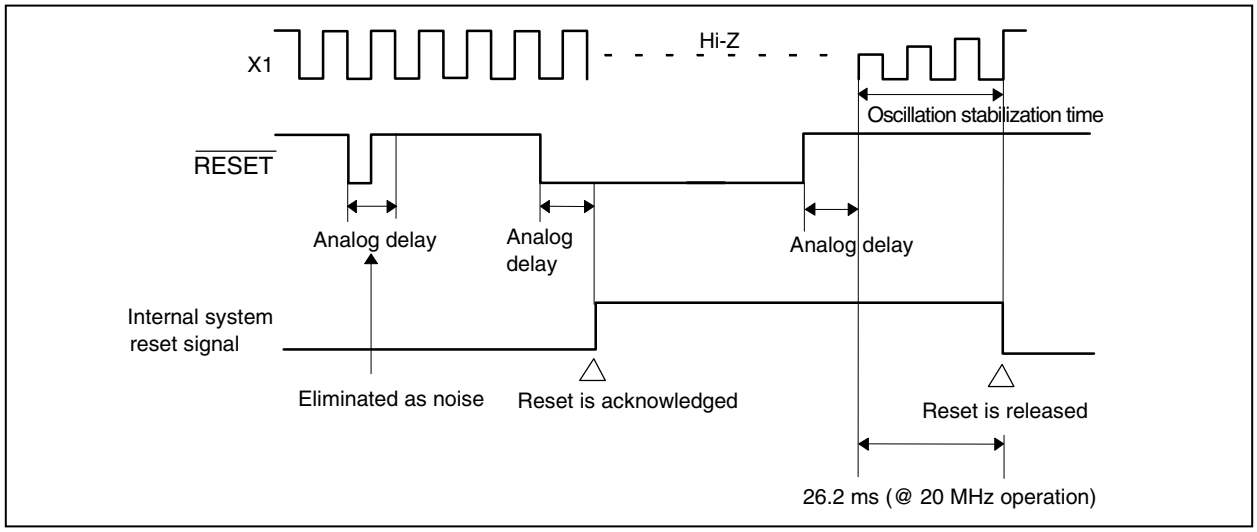
An on-chip noise eliminator uses analog delay to prevent noise-related malfunction at the $\overline{\text{RESET}}$ pin.

15.2 Pin Operations

During the system reset period, almost all pins are set to high impedance (all pins except for $\overline{\text{RESET}}$, X2, XT2, AV_{REF}, V_{DD}, V_{SS}, AV_{DD}, AV_{SS}, BV_{DD}, BV_{SS}, and IC/V_{PP}).

Accordingly, if connected to an external memory device, be sure to attach a pull-up (or pull-down) resistor to each pin of ports 3 to 6 and 9 to 11. If such a resistor is not attached, these pins will be set to high impedance, which could damage the data in memory devices. Likewise, make sure the pins are handled so as to prevent a similar effect at the signal outputs of on-chip peripheral I/O functions and output ports.

Figure 15-1. System Reset Timing



CHAPTER 16 FLASH MEMORY

The following products are the flash memory versions of the V850/SA1.

★ **Caution The flash memory version and mask ROM version differ in noise immunity and noise radiation. If replacing a flash memory version with a mask ROM version when changing from trial production to mass production, make a thorough evaluation by using the CS model (not ES model) of the mask ROM version.**

μPD70F3015B, 70F3015BY: 128 KB flash memory versions
μPD70F3017A, 70F3017AY: 256 KB flash memory versions

In the instruction fetch to this flash memory, 4 bytes can be accessed by a single clock in the same way as 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 is connected to the target system to perform writing.

The following can be considered as the development environment and applications using flash memory.

- Software can be altered after the V850/SA1 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.

16.1 Features

- 4-byte/1-clock access (in instruction fetch access)
- All area one-shot erase/area-unit erase (μPD70F3017A, 70F3017AY only)
- Communication via serial interface with the dedicated flash programmer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming
- Flash memory programming by self-programming is possible in area units (128 KB) (all areas in the μPD70F3015B, 70F3015BY)

★ 16.1.1 Erasing unit

The erasing unit differs depending on the product.

(1) μPD70F3015B, 70F3015BY

The erasing units for 128 KB flash memory versions are shown below.

(a) All area one-shot erase

The area of xx000000H to xx01FFFFH can be erased in one shot.

(2) μPD70F3017A, 70F3017AY

The erasing units for 256 KB flash memory versions are shown below.

(a) All area one-shot erase

The area of xx000000H to xx03FFFFH can be erased in one shot.

(b) Area erase

Erasure can be performed in area units (there are two 128 KB unit areas).

Area 0: The area of xx000000H to xx01FFFFH (128 KB) is erased
Area 1: The area of xx020000H to xx03FFFFH (128 KB) is erased

16.2 Writing by Flash Programmer

Writing can be performed either on-board or off-board by the dedicated flash programmer.

(1) On-board programming

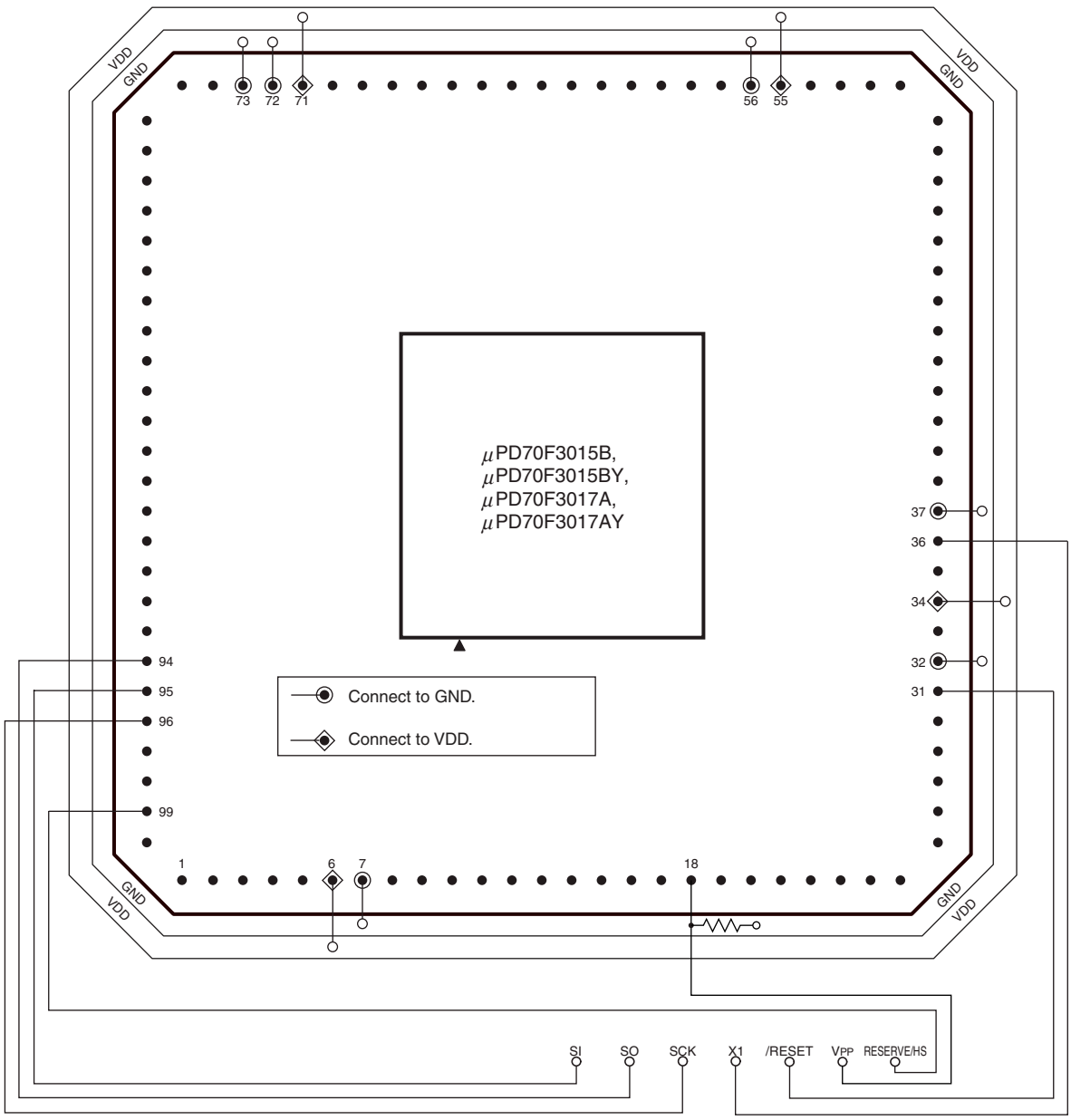
The contents of the flash memory are rewritten after the V850/SA1 is mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

(2) Off-board programming

Writing to the flash memory is performed by the dedicated program adapter (FA Series), etc., before mounting the V850/SA1 on the target system.

Remark The FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

★ **Figure 16-1. Wiring Example of V850/SA1 Flash Writing Adapter (FA100GC-8EU)**



Remarks 1. Pins not described above should be handled according to the recommended connection of unused pins (refer to **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**). When connecting to V_{DD} via a resistor, connecting a resistor of 1 kΩ to 10 kΩ is recommended.

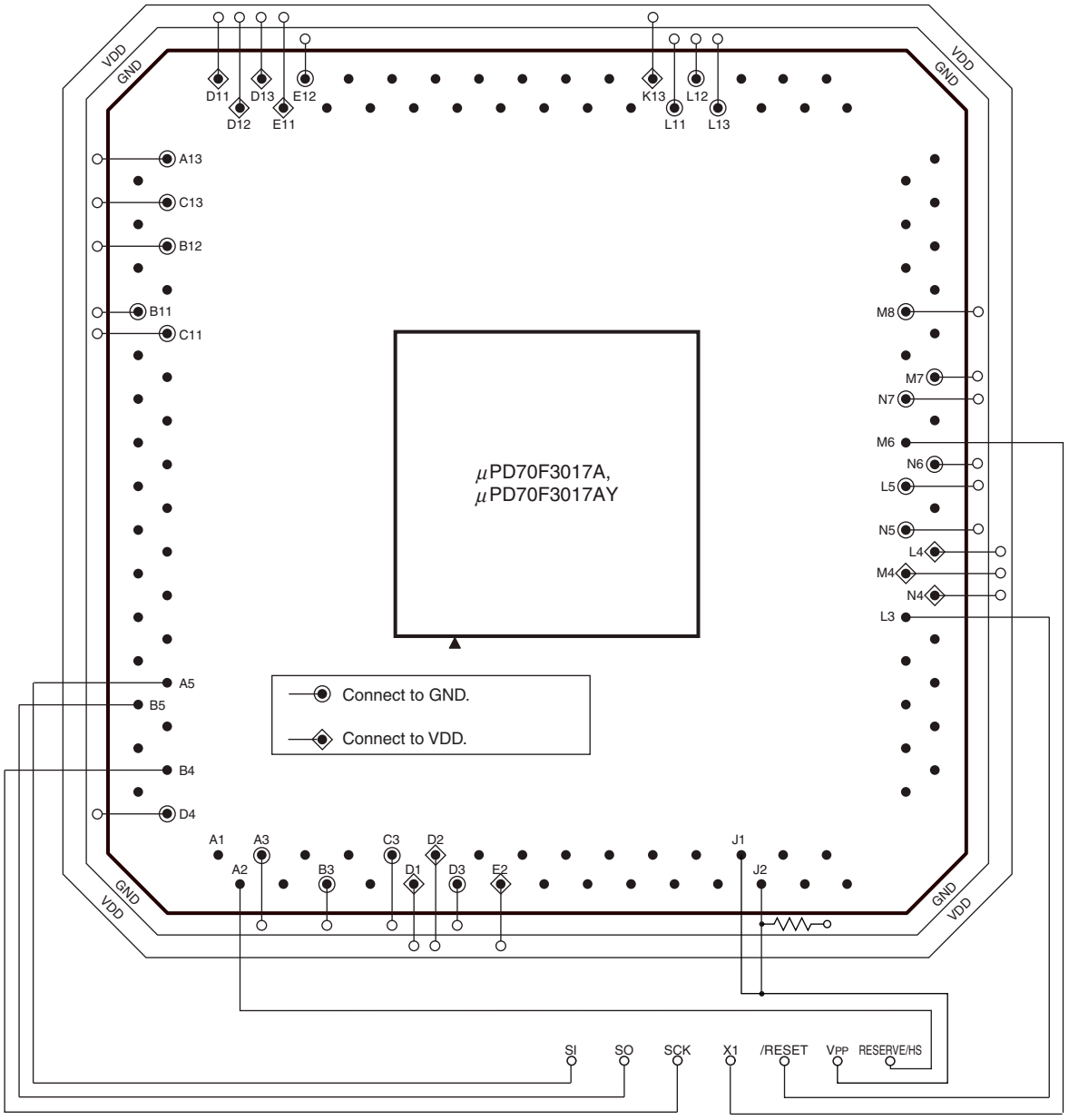
2. This adapter is for a 100-pin plastic LQFP package.

3. This figure indicates the connection when CSI supporting handshake is used.

★ **Table 16-1. Wiring Table of V850/SA1 Flash Writing Adapter (FA-100GC-8EU)**

Flash Programmer (PG-FP3/PG-FP4) Connection Pin			When Using CSI0 + HS		When Using CSI0		When Using UART0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	P11/SO0	95	P11/SO0	95	P14/TXD0	98
SO/TxD	Output	Transmit signal	P10/SI0	94	P10/SI0	94	P13/RXD0	97
SCK	Output	Transfer clock	P12/$\overline{\text{SCK0}}$	96	P12/$\overline{\text{SCK0}}$	68	Unnecessary	Unnecessary
CLK	Output	Clock to V850/SA1	X1	36	X1	36	X1	36
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	31	$\overline{\text{RESET}}$	31	$\overline{\text{RESET}}$	31
VPP	Output	Writing voltage	V_{PP}	18	V_{PP}	18	V_{PP}	18
HS	Input	Handshake signal of CSI0 + HS communication	P15	99	Unnecessary	Unnecessary	Unnecessary	Unnecessary
VDD	–	VDD voltage generation/voltage monitoring	V_{DD}	6, 34	V_{DD}	6, 34	V_{DD}	6, 34
			AV_{DD}	71	AV_{DD}	71	AV_{DD}	71
			BV_{DD}	55	BV_{DD}	55	BV_{DD}	55
GND	–	Ground	V_{SS}	7, 37	V_{SS}	7, 37	V_{SS}	7, 37
			AV_{SS}	72	AV_{SS}	72	AV_{SS}	72
			AV_{REF}	73	AV_{REF}	73	AV_{REF}	73
			BV_{SS}	56	BV_{SS}	56	BV_{SS}	56

★ **Figure 16-2. Wiring Example of V850/SA1 Flash Writing Adapter (FA-121F1-EA6)**



Remarks 1. Pins not described above should be handled according to the recommended connection of unused pins (refer to **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**). When connecting to V_{DD} via a resistor, connecting a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is for a 121-pin plastic FBGA package.

3. This figure indicates the connection when CSI supporting handshake is used.

★ **Table 16-2. Wiring Table of V850/SA1 Flash Writing Adapter (FA-121F1-EA6)**

Flash Programmer (PG-FP3/PG-FP4) Connection Pin			When Using CSI0 + HS		When Using CSI0		When Using UART0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	P11/SO0	A5	P11/SO0	A5	P14/TXD0	B2
SO/TxD	Output	Transmit signal	P10/SI0	B5	P10/SI0	B5	P13/RXD0	A4
SCK	Output	Transfer clock	P12/$\overline{SCK0}$	B4	P12/$\overline{SCK0}$	B4	Unnecessary	Unnecessary
CLK	Output	Clock to V850/SA1	X1	M6	X1	M6	X1	M6
/RESET	Output	Reset signal	$\overline{RESET}$	L3	$\overline{RESET}$	L3	$\overline{RESET}$	L3
VPP	Output	Writing voltage	V_{PP}	J1, J2	V_{PP}	J1, J2	V_{PP}	J1, J2
HS	Input	Handshake signal of CSI0 + HS communication	P15	A2	Unnecessary	Unnecessary	Unnecessary	Unnecessary
VDD	–	VDD voltage generation/voltage monitoring	V_{DD}	**Note 1**	V_{DD}	**Note 1**	V_{DD}	**Note 1**
			AV_{DD}	**Note 2**	AV_{DD}	**Note 2**	AV_{DD}	**Note 2**
			BV_{DD}	K13	BV_{DD}	K13	BV_{DD}	K13
GND	–	Ground	V_{SS}	**Note 3**	V_{SS}	**Note 3**	V_{SS}	**Note 3**
			AV_{SS}	**Note 4**	AV_{SS}	**Note 4**	AV_{SS}	**Note 4**
			AV_{REF}	C13	AV_{REF}	C13	AV_{REF}	C13
			BV_{SS}	L11 to L13	BV_{SS}	L11 to L13	BV_{SS}	L11 to L13

Notes 1. D1, D2, E2, L4, M4, N4

2. D11 to D13, E11

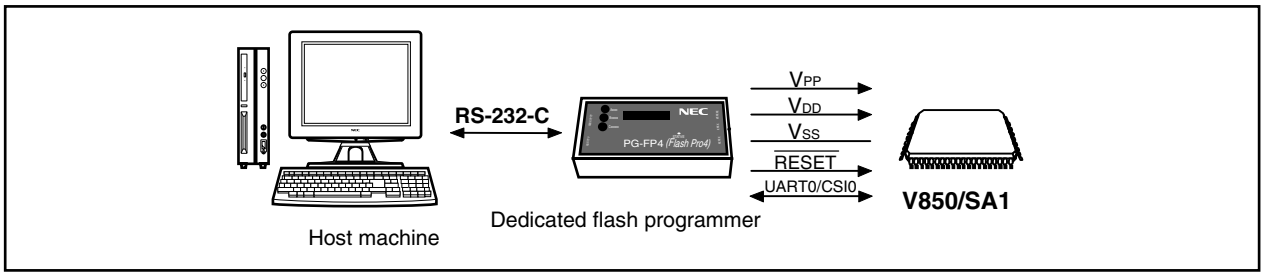
3. A3, B3, C3, D3, L5, M7, M8, N6, N7

4. A13, B11, B12, C11

16.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850/SA1.

Figure 16-3. Environment 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 V850/SA1 to perform writing, erasing, etc. A dedicated program adapter (FA Series) is required for off-board writing.

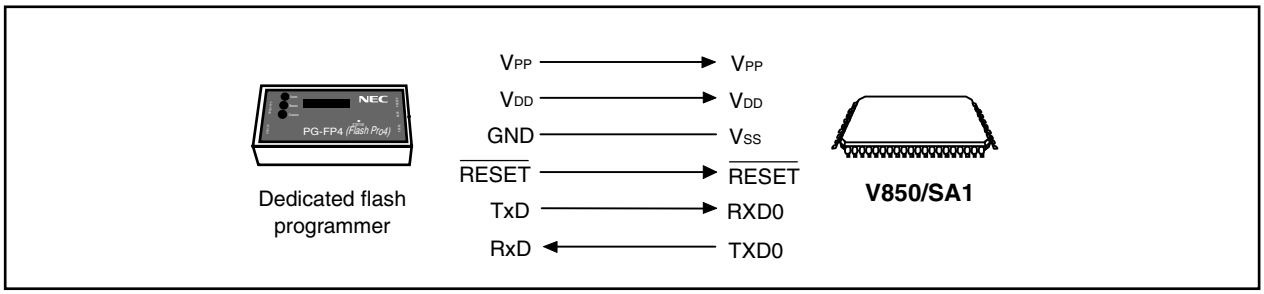
16.4 Communication System

The communication between the dedicated flash programmer and the V850/SA1 is performed by serial communication using UART0 or CSI0.

(1) UART0

Transfer rate: 9600 to 76800 bps

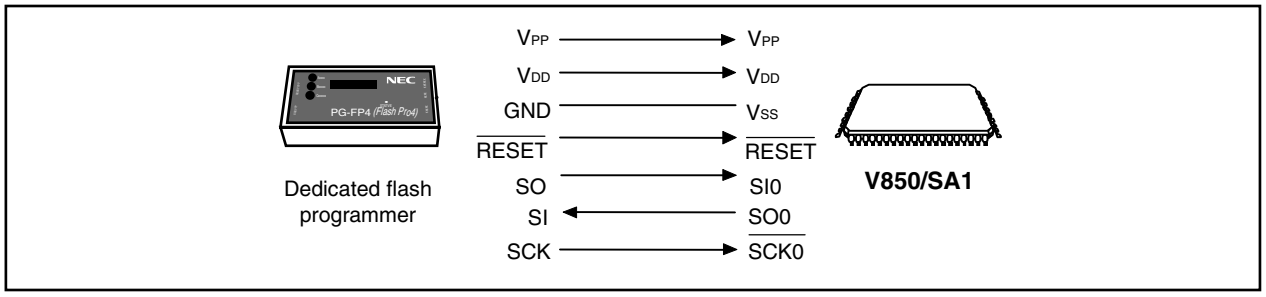
Figure 16-4. Communication with Dedicated Flash Programmer (UART0)



(2) CSI0

Serial clock: Up to 1 MHz (MSB first)

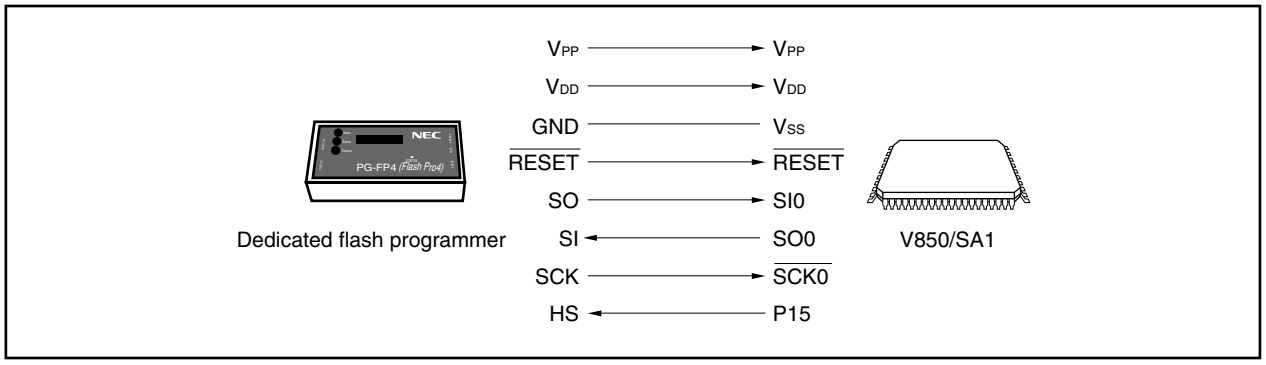
Figure 16-5. Communication with Dedicated Flash Programmer (CSI0)



(3) CSI0 + HS

Serial clock: Up to 1 MHz (MSB first)

Figure 16-6. Communication with Dedicated Flash Programmer (CSI0 + HS)



The dedicated flash programmer outputs the transfer clock, and the V850/SA1 operates as a slave.

When the PG-FP3 or PG-FP4 is used as the dedicated flash programmer, it generates the signals shown in Table 16-3 to the V850/SA1. For the details, refer to **PG-FP3 User's Manual (U13502E)**, or **PG-FP4 User's Manual (U15260E)**.

Table 16-3. Signal Generation of Dedicated Flash Programmer (PG-FP3 or PG-FP4)

PG-FP3 or PG-FP4			V850/SA1	Measures When Connected		
Signal Name	I/O	Pin Function	Pin Name	CSI0	UART0	CSI0 + HS
V_{PP}	Output	Writing voltage	V_{PP}	◎	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD}	◎	◎	◎
GND	–	Ground	V_{SS}	◎	◎	◎
CLK[Note]	Output	Clock output to V850/SA1	X1	○	○	○
$\overline{\text{RESET}}$	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 communication	P15	×	×	◎

Note Supply clocks on the target board.

Remark ◎: Always connected
○: If this signal is generated on the target board, it does not need to be connected.
×: Does not need to be connected

16.5 Pin Connection

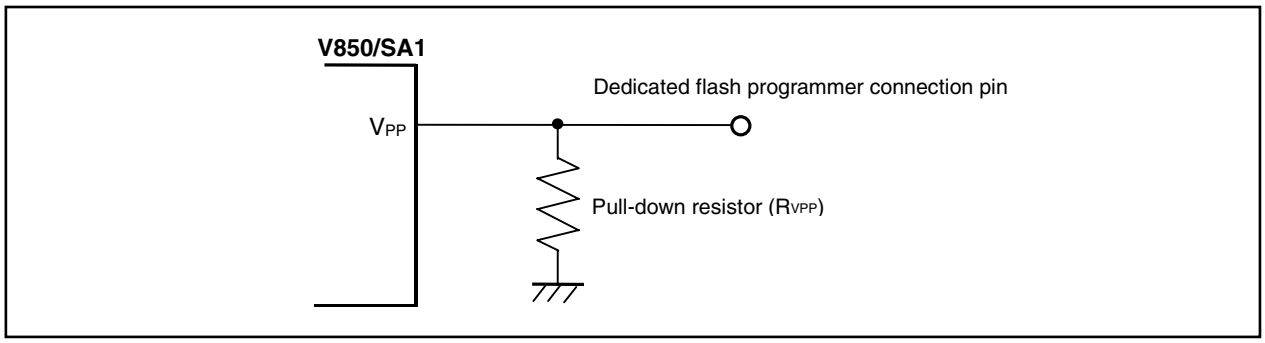
When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

When switched to the flash memory programming mode, all the pins not used for the flash memory programming become the same status as that immediately after reset. Therefore, all the ports become output high-impedance status, so that pin handling is required when the external device does not acknowledge the output high-impedance status.

16.5.1 V_{PP} pin

In the normal operation mode, 0 V is input to V_{PP} pin. In the flash memory programming mode, 7.8 V writing voltage is supplied to V_{PP} pin. The following shows an example of the connection of V_{PP} pin.

Figure 16-7. Connection Example of V_{PP} Pin



16.5.2 Serial interface pin

The following shows the pins used by each serial interface.

Table 16-4. Pins Used by Each Serial Interface

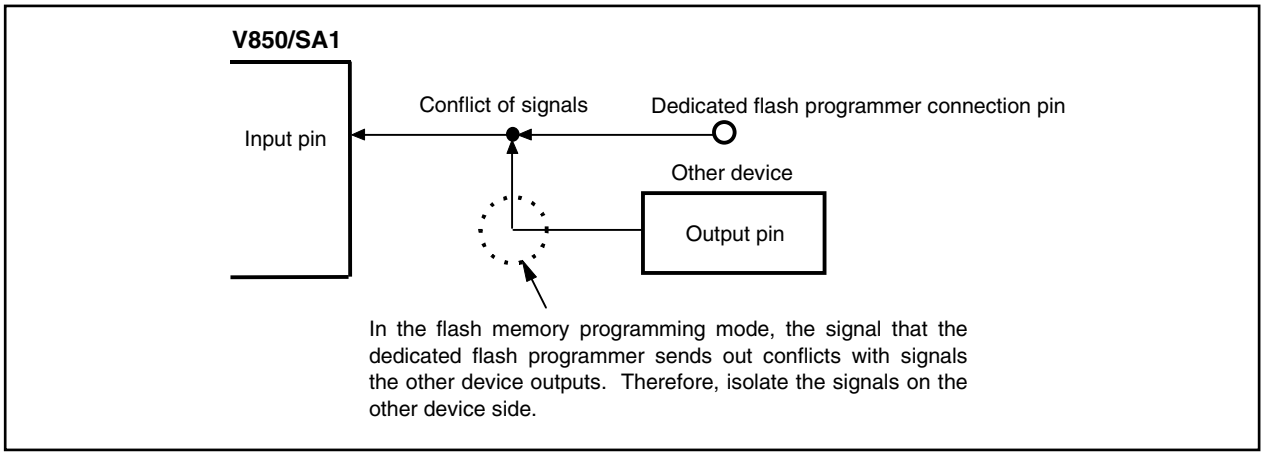
Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$
CSI0 + HS	SO0, SI0, $\overline{SCK0}$, P15
UART0	TXD0, RXD0

When connecting a dedicated flash programmer to a serial interface pin that is connected to other devices on-board, care should be taken to avoid conflict of signals and malfunction of the other devices, etc.

(1) Conflict of signals

When connecting a dedicated flash programmer (output) to a serial interface pin (input) that is connected to another device (output), 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.

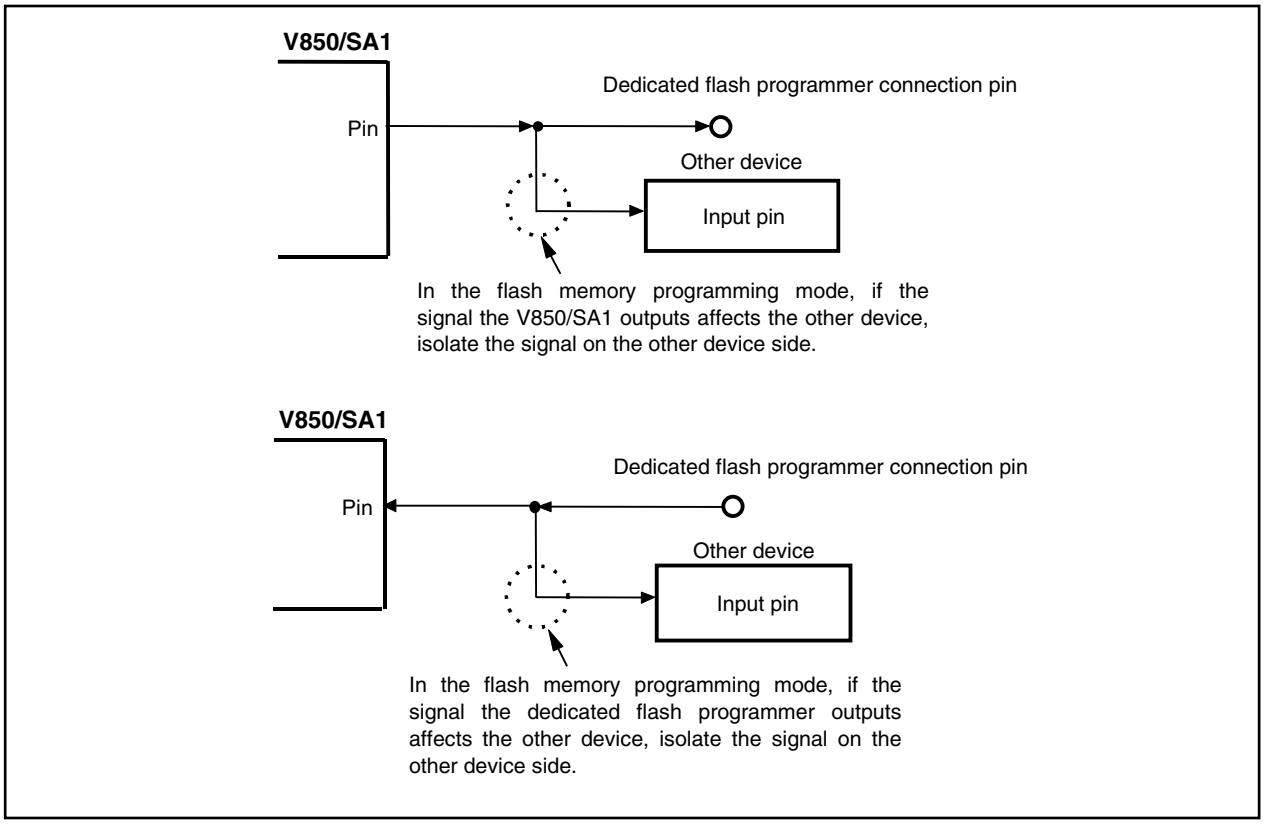
Figure 16-8. Conflict of Signals (Serial Interface Input Pin)



(2) Malfunction of the other device

When connecting a dedicated flash programmer (output or input) to a serial interface pin (input or output) that is connected to another device (input), the signal output to the other device may cause 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 16-9. Malfunction of Other Device

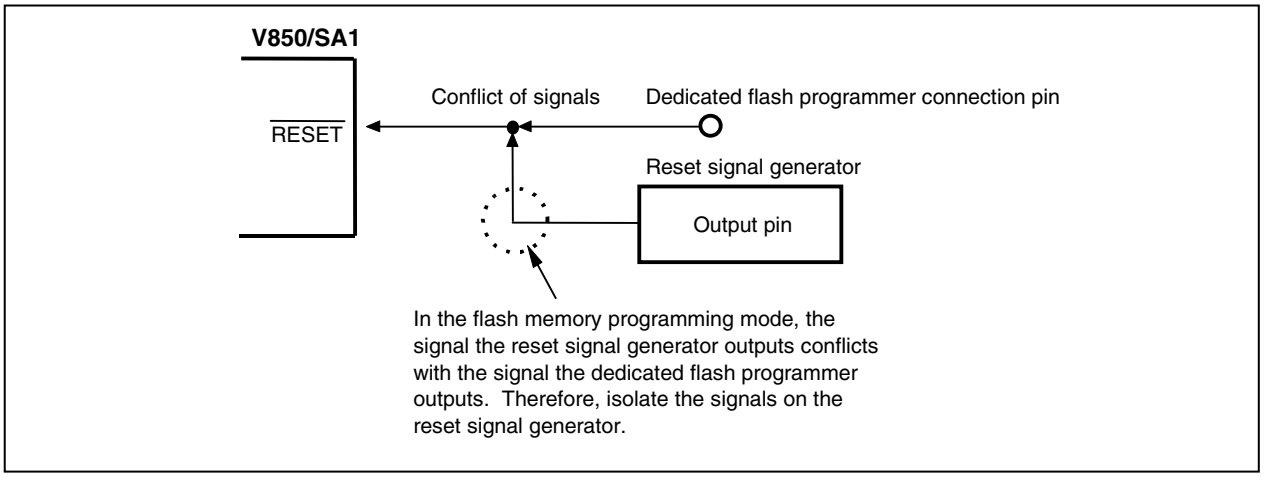


16.5.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, 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 16-10. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



16.5.4 Port pin (including NMI)

When the flash memory programming mode is set, all the port pins except the pins that communicate with the dedicated flash programmer become output high-impedance status. If problems such as disabling output high-impedance status should occur to the external devices connected to the port, connect them to V_{DD} or V_{SS} via resistors.

16.5.5 Other signal pins

Connect X1, X2, XT2, and AV_{REF} to the same status as that in the normal operation mode.

16.5.6 Power supply

Supply the same power supply (V_{DD}, V_{SS}, AV_{DD}, AV_{SS}, BV_{DD}, BV_{SS}) as when in normal operation mode.

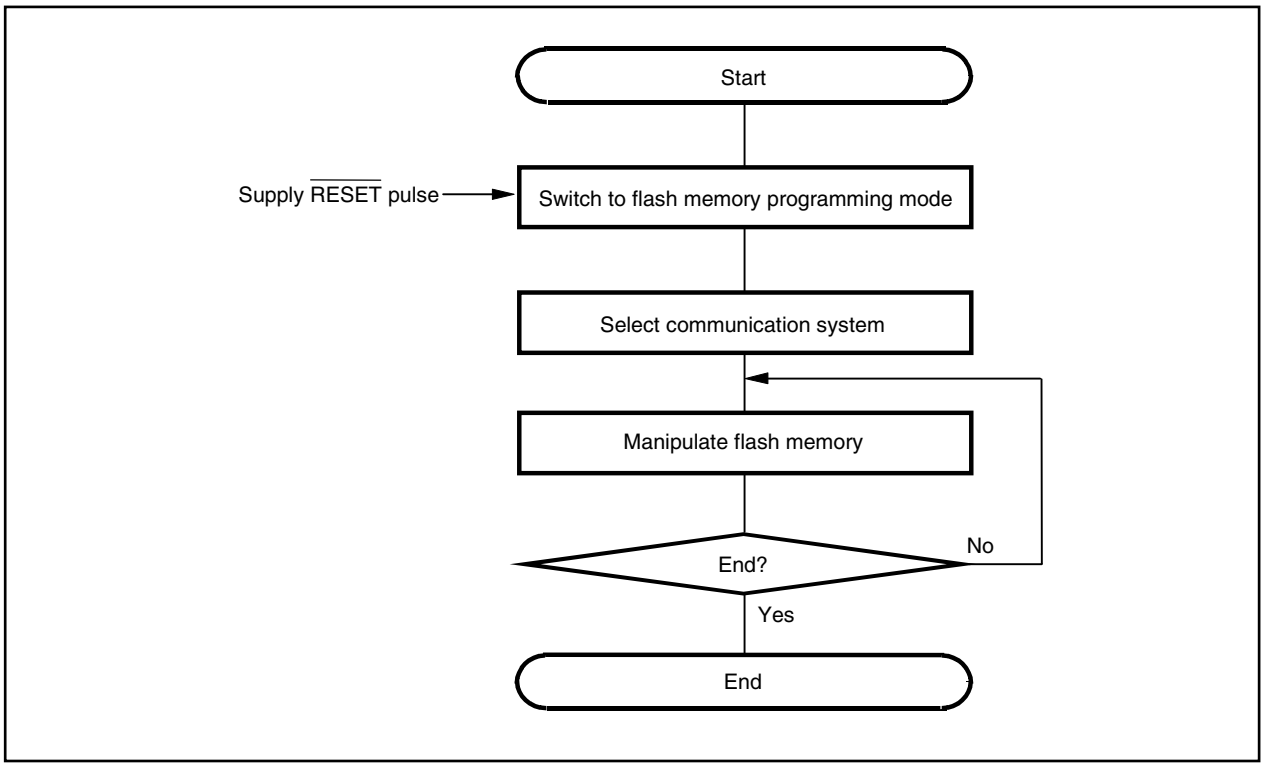
In addition, connect V_{DD} and V_{SS} to V_{DD} and GND of the dedicated flash programmer (V_{DD} of the dedicated flash programmer has a power supply monitoring function).

16.6 Programming Method

16.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 16-11. Manipulation Procedure of Flash Memory

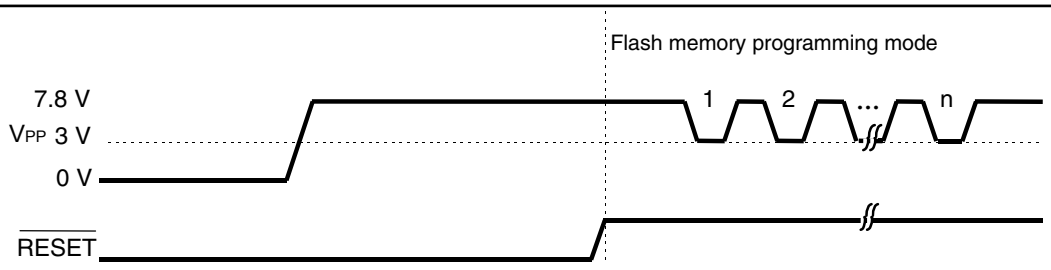


16.6.2 Flash memory programming mode

When rewriting the contents of the flash memory using the dedicated flash programmer, set the V850/SA1 in the flash memory programming mode. When switching modes, set V_{PP} pin before releasing reset.

When performing on-board writing, change modes using a jumper, etc.

Figure 16-12. Flash Memory Programming Mode



VPP	Operation Mode
0 V	Normal operation mode
7.8 V	Flash memory programming mode

16.6.3 Selection of communication mode

In the V850/SA1, the communication mode is selected by inputting a pulse (16 pulses max.) to VPP pin after switching to the flash memory programming mode. The VPP pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

Table 16-5. List of Communication Modes

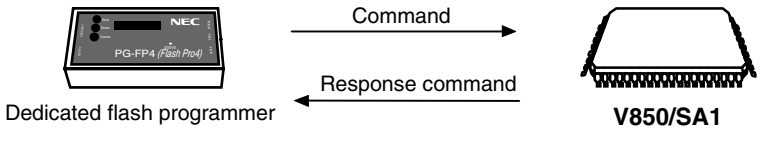
VPP Pulse	Communication Mode	Remarks
0	CSI0	V850/SA1 performs slave operation, MSB first
3	CSI0 + HS	V850/SA1 performs slave operation, MSB first
8	UART0	Communication rate: 9600 bps (at reset), LSB first
Other	RFU	Setting prohibited

Caution When UART0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the VPP pulse.

16.6.4 Communication command

The V850/SA1 communicates with the dedicated flash programmer by means of commands. The command sent from the dedicated flash programmer to the V850/SA1 is called a “command”. The response signal sent from the V850/SA1 to the dedicated flash programmer is called a “response command”.

Figure 16-13. Communication Command



The following shows the commands for flash memory control of the V850/SA1. All of these commands are issued from the dedicated flash programmer, and the V850/SA1 performs the various processing corresponding to the commands.

Table 16-6. Commands for Flash Memory Control

	Category	Command Name	Function
	Verify	One-shot verify command	Compares the contents of the entire memory and the input data.
★		Area verify command	Compares the contents of the specified area and the input data.
	Erase	One-shot erase command	Erases the contents of the entire memory.
★		Area erase command	Erases the contents of the specified area.
		Write back command	Writes back the contents which is overerased.
	Blank check	One-shot blank check command	Checks the erase state of the entire memory.
★		Area blank check command	Checks the erase state of the specified area.
	Data write	High-speed write command	Writes data by the specification of the write address and the number of bytes to be written, and executes a verify check.
		Continuous write command	Writes data from the address following the high-speed write command executed immediately before, and executes a verify check.
	System setting and control	Status read out command	Acquires the status of operations.
		Oscillating frequency setting command	Sets the oscillation frequency.
		Erasing time setting command	Sets the erasing time of one-shot erase.
		Writing time setting command	Sets the writing time of data write.
		Write back time setting command	Sets the write back time.
		Baud rate setting command	Sets the baud rate when using UART.
		Silicon signature command	Reads outs the silicon signature information.
		Reset command	Escapes from each state.

The V850/SA1 sends back response commands to the commands issued from the dedicated flash programmer. The following shows the response commands the V850/SA1 sends out.

Table 16-7. Response Commands

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

16.6.5 Resources used

The resources used in the flash memory programming mode are all the FFE000H to FFE7FFH area of the internal RAM and all the registers. The FFE800H to FFEFFFH area of the internal RAM retains data as long as the power is on. The registers that are initialized by reset are changed to the default values.

★ 16.7 Flash Memory Programming by Self-Programming

The V850/SA1 supports a self-programming function to rewrite the flash memory using a user program. By using this function, the flash memory can be rewritten by a user application. This self-programming function can also be used to upgrade the program in the field.

16.7.1 Outline of self-programming

Self-programming implements erasure and writing of the flash memory by calling the self-programming function (device's internal processing) on the program placed in other than the internal ROM area (000000H to 0FFFFFH). To place the program in the block 0 space and internal ROM area, copy the program to areas other than 000000H to 0FFFFFH (e.g. internal RAM area) and execute the program to call the self-programming function.

To call the self-programming function, change the operating mode from normal mode to self-programming mode using the flash programming mode control register (FLPMC).

Figure 16-14. Outline of Self-Programming (1/2)

(a) μPD70F3015B, 70F3015BY

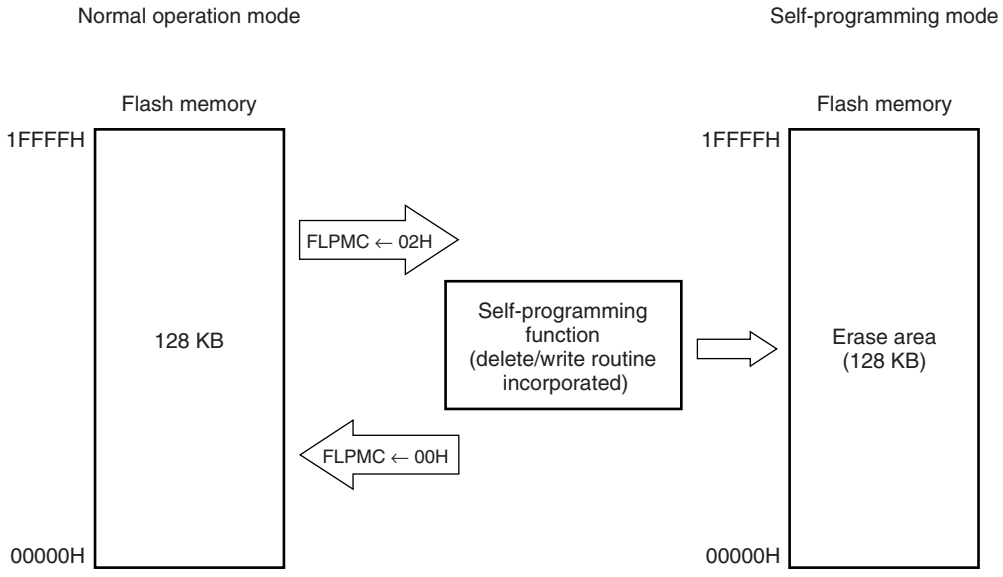
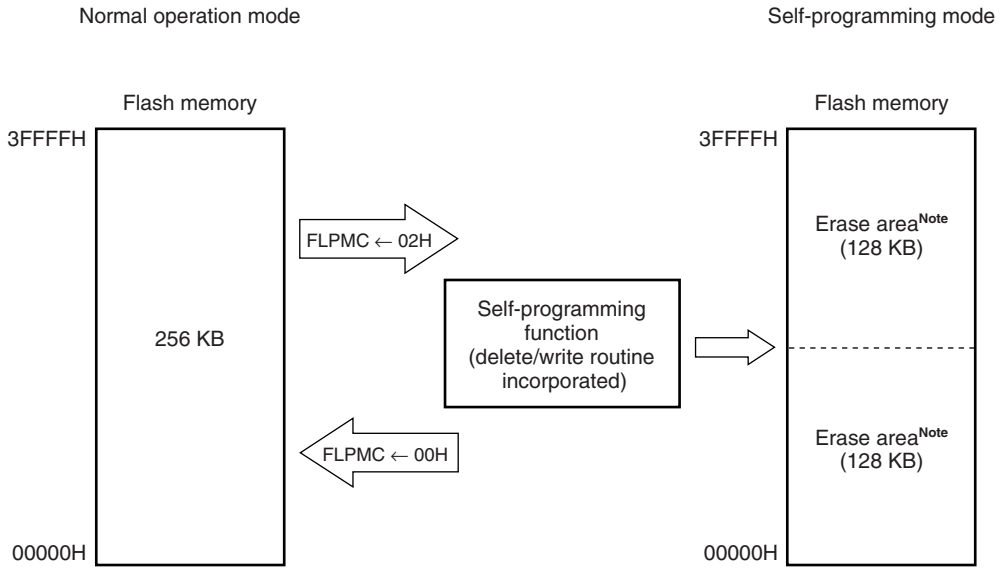


Figure 16-14. Outline of Self-Programming (2/2)

(b) μPD70F3017A, 70F3017AY



Note Data is deleted in area units (128 KB).

16.7.2 Self-programming function

The V850/SA1 provides self-programming functions, as shown below. By combining these functions, erasing/writing flash memory becomes possible.

Table 16-8. Function List

Type	Function Name	Function
Erase	Area erase	Erases the specified area.
Write	Continuous write in word units	Continuously writes the specified memory contents from the specified flash memory address, for the number of words specified in 4-byte units.
	Prewrite	Writes 0 to flash memory before erasure.
Check	Erase verify	Checks whether an overerase occurred after erasure.
	Erase byte verify	Checks whether erasure is complete.
	Internal verify	Checks whether the signal level of the post-write data in flash memory is appropriate.
Write back	Area write back	Writes back the flash memory area in which an overerase occurred.
Acquire information	Flash memory information read	Reads out information about flash memory.

16.7.3 Outline of self-programming interface

To execute self-programming using the self-programming interface, the environmental conditions of the hardware and software for manipulating the flash memory must be satisfied.

It is assumed that the self-programming interface is used in an assembly language.

(1) Entry program

This program is used to call the internal processing of the device.

It is a part of the application program, and must be executed in memory other than the internal ROM area (flash memory).

(2) Device internal processing

This is manipulation of the flash memory executed inside the device.

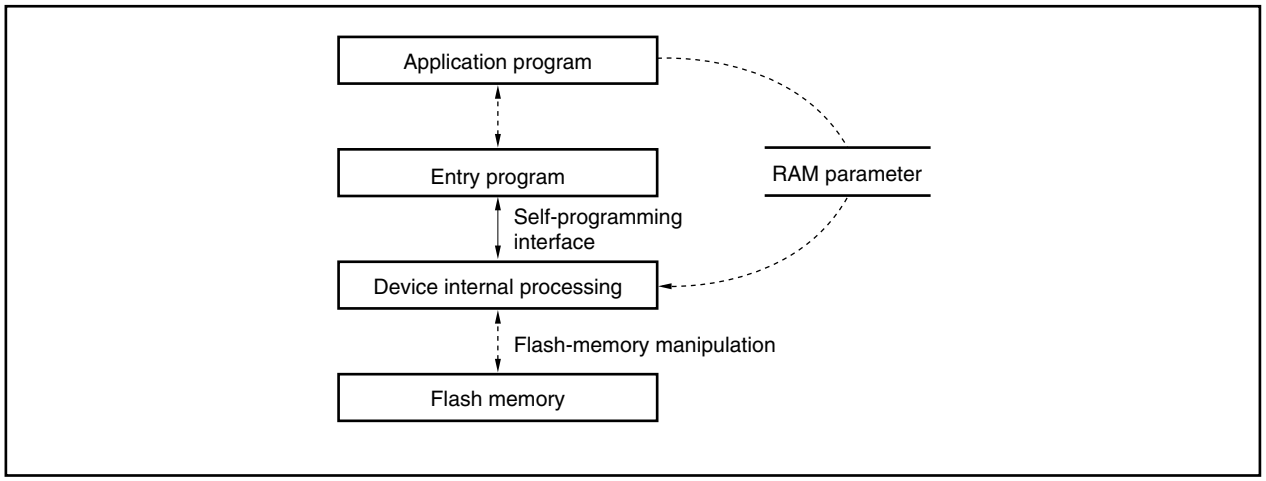
This processing manipulates the flash memory after it has been called by the entry program.

(3) RAM parameter

This is a RAM area to which the parameters necessary for self-programming, such as write time and erase time, are written. It is set by the application program and referenced by the device internal processing.

The self-programming interface is outlined below.

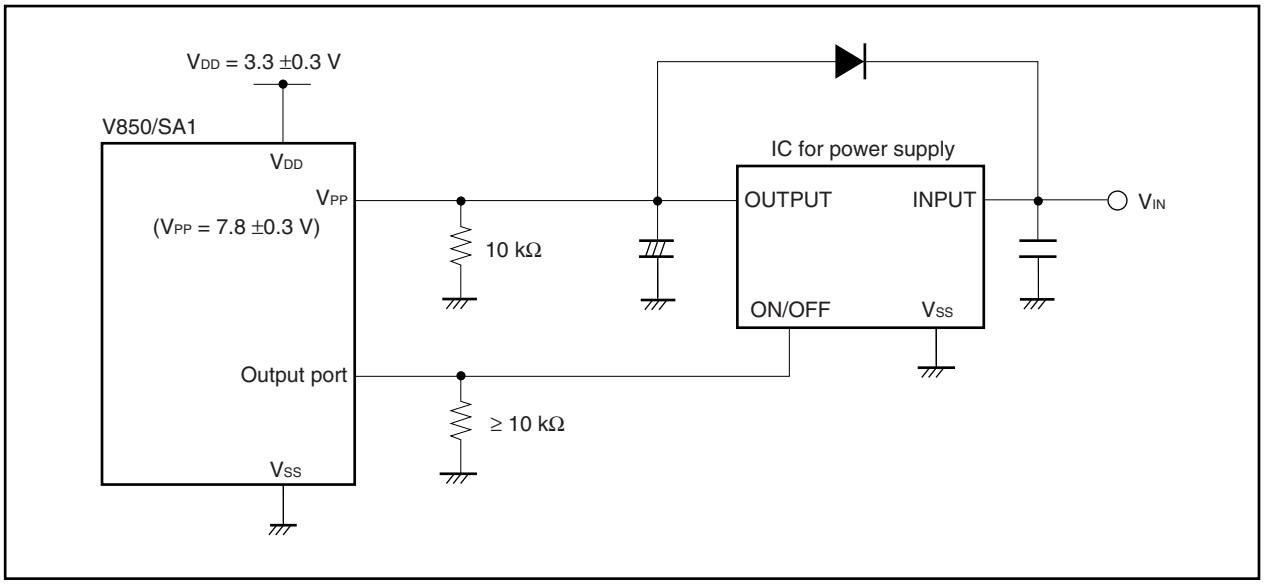
Figure 16-15. Outline of Self-Programming Interface



16.7.4 Hardware environment

To write or erase the flash memory, a high voltage must be applied to the V_{PP} pin. To execute self-programming, a circuit that can generate a write voltage (V_{PP}) and that can be controlled by software is necessary on the application system. An example of a circuit that can select a voltage to be applied to the V_{PP} pin by manipulating a port is shown below.

Figure 16-16. Example of Self-Programming Circuit Configuration



The voltage applied to the V_{PP} pin must satisfy the following conditions.

- Hold the voltage applied to the V_{PP} pin at 0 V in the normal operation mode and hold the V_{PP} voltage only while the flash memory is being manipulated.
- The V_{PP} voltage must be stable from before manipulation of the flash memory starts until manipulation is complete.

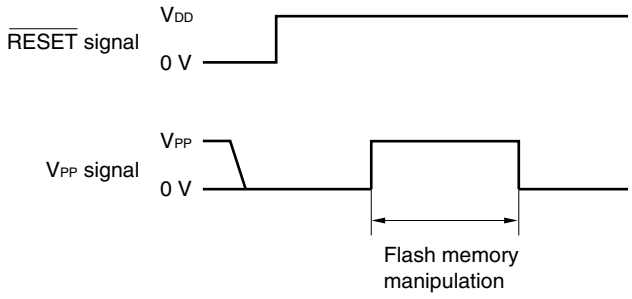
Cautions 1. Apply 0 V to the V_{PP} pin when reset is released.

2. Implement self-programming in single-chip mode.

3. Apply the voltage to the V_{PP} pin in the entry program.

4. If both writing and erasing are executed by using the self-programming function and flash memory programmer on the target board, be sure to communicate with the programmer using CSI0 (do not use the handshake-supporting CSI and UART0).

Figure 16-17. Timing to Apply Voltage to V_{PP} Pin



16.7.5 Software environment

The following conditions must be satisfied before using the entry program to call the device internal processing.

Table 16-9. Software Environmental Conditions

Item	Description
Location of entry program	Execute the entry program in memory other than the flash memory area. The device internal processing cannot be directly called by the program that is executed on the flash memory.
Execution status of program	The device internal processing cannot be called while an interrupt is being serviced (NP bit of PSW = 0, ID bit of PSW = 1).
Masking interrupts	Mask all the maskable interrupts used. Mask each interrupt by using the corresponding interrupt control register. Mask the maskable interrupts even when the ID bit of the PSW = 1 (interrupts are disabled).
Manipulation of V_{PP} voltage	Stabilize the voltage applied to the V_{PP} pin (V_{PP} voltage) before starting manipulation of the flash memory. After completion of manipulation, return the voltage of the V_{PP} pin to 0 V.
Initialization of internal timer	Do not use 16-bit timer 0 while the flash memory is being manipulated. Because 16-bit timer 0 is initialized after the flash memory has been used, initialize the timer with the application program to use the timer again.
Stopping reset signal input	Do not input the reset signal while the flash memory is being manipulated. If the reset signal is input while the flash memory is being manipulated, the contents of the flash memory under manipulation become undefined.
Stopping NMI signal input	Do not input the NMI signal while the flash memory is being manipulated. If the NMI signal is input while the flash memory is being manipulated, the flash memory may not be correctly manipulated by the device internal processing. If an NMI occurs while the device internal processing is in progress, the occurrence of the NMI is reflected in the NMI flag of the RAM parameter. If manipulation of the flash memory is affected by the occurrence of the NMI, the function of each self-programming function is reflected in the return value.
Reserving stack area	The device internal processing takes over the stack used by the user program. It is necessary that an area of 300 bytes be reserved for the stack size of the user program when the device internal processing is called. r3 is used as the stack pointer.
Saving general-purpose registers	The device internal processing rewrites the contents of r6 to r14, r20, and r31 (lp). Save and restore these register contents as necessary.

16.7.6 Self-programming function number

To identify a self-programming function, the following numbers are assigned to the respective functions. These function numbers are used as parameters when the device internal processing is called.

Table 16-10. Self-Programming Function Numbers

Function No.	Function Name
0 to 2	RFU
3	Erase verify
4	Erase byte verify
5	Flash information acquisition
6	RFU
7	Successive write in word units
8 to 10	RFU
11	Pre-write
12	Successive write in word units
13	Area write back
14	Area erase
Other	Prohibited

Remark RFU: Reserved for Future Use

16.7.7 Calling parameters

The arguments used to call the self-programming function are shown in the table below. In addition to these arguments, parameters such as the write time and erase time are set to the RAM parameters indicated by ep (r30).

Table 16-11. Calling Parameters

Function Name	First Argument (r6) Function No.	Second Argument (r7)	Third Argument (r8)	Fourth Argument (r9)	Return Value (r10)
Erase verify	3	None (acts on erase manipulation area immediately before)	–	–	0: Normal completion Other than 0: Error
Erase byte verify	4	Verify start address	Number of bytes to be verified	–	0: Normal completion Other than 0: Error
Acquiring flash information	5	Option number[Note 1]	–	–	**Note 1**
Successive write in word units[Note 2]	7	Write start address[Note 3]	Start address of write source data[Note 3]	Number of words to be written (word units)	0: Normal completion Other than 0: Error
Pre-write	11	Write start address	Number of bytes to be written	–	0: Normal completion Other than 0: Error
Internal verify	12	Verify start address	Number of bytes to be verified	–	0: Normal completion Other than 0: Error
Area write back	13	None (acts on erase manipulation area immediately before)	–	–	None
Erasing area	14	Area erase start address	–	–	0: Normal completion Other than 0: Error

Notes 1. See **16.7.10 Flash information** for details.

2. Prepare write source data in memory other than the flash memory when data is written successively in word units.

3. This address must be at a 4-byte boundary.

Caution For all the functions, ep (r30) must indicate the first address of the RAM parameter.

16.7.8 Contents of RAM parameters

Reserve the following 48-byte area in the internal RAM or external RAM for the RAM parameters, and set the parameters to be input. Set the base addresses of these parameters to ep (r30).

Table 16-12. Description of RAM Parameter

Address	Size	I/O	Description
ep+0	4 bytes	−	For internal operations
ep+4:Bit 0[Note 1]	1 bit	Input	Internal flag 0: Always set to 0 1: Setting prohibited
ep+4:Bit 5[Note 2]	1 bit	Input	Operation flag (Be sure to set this flag to 1 before calling the device internal processing.) 0: Normal operation in progress 1: Self-programming in progress
ep+4:Bit 7[Notes 3, 4]	1 bit	Output	NMI flag 0: NMI not detected 1: NMI detected
ep+8	4 bytes	Input	Step erase time (unsigned 4 bytes) Expressed as 1 count value in units of the internal operation unit time (100 μs). Set value = Erase time (μs)/internal operation unit time (μs) Example: If erase time is 0.2 s $\rightarrow 0.2 \times 1{,}000{,}000/100 = 2{,}000$ (integer operation)
ep+0xC	4 bytes	Input	Write back time (unsigned 4 bytes) Expressed as 1 count value in units of the internal operation unit time (100 μs). Set value = Write back time (μs)/internal operation unit time (μs) Example: If write back time is 1 ms $\rightarrow 1 \times 1{,}000/100 = 10$ (integer operation)
ep+0x10	2 bytes	Input	Timer set value for creating internal operation unit time (unsigned 2 bytes) Write a set value that makes the value of 16-bit timer 0 the internal operation unit time (100 μs). Set value = Operating frequency (Hz)/1,000,000 $\times$ Internal operation unit time (μs)/ Timer division ratio (2) + 1[Note 5] Example: If the operating frequency is 20 MHz $\rightarrow 20{,}000{,}000/1{,}000{,}000 \times 100/2 + 1 = 1{,}001$ (integer operation)
ep+0x12	2 bytes	Input	Timer set value for creating write time (unsigned 2 bytes) Write a set value that makes the value of 16-bit timer 0 the write time. Set value = Operating frequency (Hz)/Write time (μs)/Timer division ratio (2) + 1[Note 5] Example: If the operating frequency is 20 MHz and the write time is 20 μs $\rightarrow 20{,}000{,}000/1{,}000{,}000 \times 20/2 + 1 = 201$ (integer operation)
ep+0x14	12 bytes	−	For internal operations

Notes 1. Bit 0 of the address of ep+4 (least significant bit is bit 0.)

2. 5th bit of address of ep+4 (least significant bit is bit 0.)

3. 7th bit of address of ep+4 (least significant bit is bit 0.)

4. Clear the NMI flag by the user program because it is not cleared by the device internal processing.

5. The device internal processing sets this value minus 1 to the timer. Because the fraction is rounded up, add 1 as indicated by the expression of the set value.

Caution Be sure to reserve the RAM parameter area at a 4-byte boundary.

16.7.9 Errors during self-programming

The following errors related to manipulation of the flash memory may occur during self-programming. An error occurs if the return value (r10) of each function is not 0.

Table 16-13. Errors During Self-Programming

Error	Function	Description
Overerase error	Erase verify	Excessive erasure occurs.
Undererase error (blank check error)	Erase byte verify	Erasure is insufficient. Additional erase operation is needed.
Verify error	Successive writing in word units	The written data cannot be correctly read. Either an attempt has been made to write to flash memory that has not been erased, or writing is not sufficient.
Internal verify error	Internal verify	The written data is not at the correct signal level.

Caution The overerase error and undererase error may simultaneously occur in the entire flash memory.

16.7.10 Flash information

For the flash information acquisition function (function No. 0), the option number (r7) to be specified and the contents of the return value (r10) are as follows. To acquire all flash information, call the function as many times as required in accordance with the format shown below.

Table 16-14. Flash Information

Option No. (r7)	Return Value (r10)
0	Specification prohibited
1	Bit representation of return value (MSB: bit 31) FFFFFFFFFFFFFFFFAAAAAAAABBBBBBBB (LSB: bit 0) Bits 31 to 16: FFFFFFFFFFFFFFFF (reserved for future use) Mask bits 31 to 16 because they are not normally 0. Bits 15 to 8: AAAAAAAA (number of areas) (unsigned 8 bits) Bits 7 to 0: FFFFFFFF (reserved for future use) Mask bits 7 to 0 because they are not normally 0.
2	RFU
3	RFU
: :	: :
Offset number + 1	RFU
Offset number + 2	End address of area 0
Offset number + 3	End address of area 1

Cautions 1. The start address of area 0 is 0. The "end address + 1" of the preceding area is the start address of the next area.

2. The flash information acquisition function does not check values such as the maximum number of areas specified by the argument of an option. If an illegal value is specified, an undefined value is returned.

Remark RFU: Reserved for Future Use

16.7.11 Area number

The area numbers and memory map of the V850/SA1 are shown below.

Figure 16-18. Area Configuration

16.7.12 Flash programming mode control register (FLPMC)

The flash memory mode control register (FLPMC) is a register used to enable/disable writing to flash memory and to specify the self-programming mode.

This register can be read/written in 8-bit or 1-bit units (the VPP bit (bit 2) is read-only).

Cautions 1. Be sure to transfer control to the internal RAM or external memory beforehand to manipulate the FLSPM bit. However, in on-board programming mode set by the flash programmer, the specification of FLSPM bit is ignored.

2. Be sure to set bits 0 and 5 to 7 to 0 and bit 4 to 1.

	7	6	5	4	3	2	1	0	Address	After reset[Note]
FLPMC	0	0	0	1	VPPDIS	VPP	FLSPM	0	FFFFF8D4H	18H/38H/1CH/3CH

Note 18H/38H: When writing voltage is not applied to the V_{PP} pin
1CH/3CH: When writing voltage is applied to the V_{PP} pin

Bit Position	Bit Name	Function
3	VPPDIS	V_{PP} Disable Enables/disables writing/erasing on-chip flash memory. When this bit is 1, writing/erasing on-chip flash memory is disabled even if a high voltage is applied to the V_{PP} pin. 0: Enables writing/erasing flash memory 1: Disables writing/erasing flash memory
2	VPP	V_{PP} Indicates that the voltage applied to the V_{PP} pin has reached the writing-enabled level. This bit is used to check whether writing is possible or not in the self-programming mode. 0: Indicates high-voltage application is not detected (the voltage has not reached the writing voltage enable level) 1: Indicates high-voltage application is detected (the voltage has reached the writing voltage enable level)
1	FLSPM	Flash Self Programming Mode Controls switching between internal ROM and the self-programming interface. This bit can switch the mode between the normal mode set by the mode pin on the application system and the self-programming mode. The setting of this bit is valid only if the voltage applied to the V_{PP} pin reaches the writing voltage enable level. 0: Normal mode (for all addresses, instruction fetch is performed from on-chip flash memory) 1: Self-programming mode (device internal processing is started.)

Remark Because the mask ROM versions (μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, and 703017AY) do not have the FLPMC register, an undefined value is read if an attempt is made to read the FLPMC register.

The following sequence shows the data setting of the FLPMC register.

<1> Disable DMA operation.
<2> Set the PSW NP bit to 1 (interrupts disabled).
<3> Write any 8-bit data in the command register (PRCMD).
<4> Write the set data in the FLPMC register (using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<5> Return the PSW NP bit to 0 (interrupt disable canceled).
<6> Insert five NOP instructions (when manipulating the FLSPM bit).
<7> If necessary, enable DMA operation.

No special sequence is required when reading the FLPMC register.

Cautions 1. If an interrupt request or a DMA request is acknowledged between the time PRCMD is generated (<3>) and the FLPMC register write operation (<4>) that follows immediately after, the write operation to the FLPMC register is not performed and a protection error (PRERR bit of SYS register is 1) may occur. Therefore, set the NP bit of the PSW to 1 (<2>) to disable the acknowledgement of INT/NMI or to disable DMA transfer.
The above also applies when a bit manipulation instruction is used to set the FLPMC register.
A description example is given below.

[Description example]

```
LDSR rX.5             ;  NP bit = 1
ST.B  r0, PRCMD [r0]  ;  Write to PRCMD
ST.B  rD, FLPMC [r0]  ;  FLPMC register setting
LDSR rY, 5            ;  NP bit = 0
NOP                   ;  Dummy instruction (5 instructions, when
                         manipulating FLSPM bit)
NOP
NOP
NOP
NOP
(next instruction)    ;  Execution routine following cancellation of
      .                  IDLE/software STOP mode
      .
      .
```

rX: Value to be written to PSW
rY: Value to be written back to PSW
rD: Value to be set to FLPMC

When saving the value of the PSW, the value of the PSW prior to setting the NP bit must be transferred to the rY register.

Cautions 2. Always stop the DMA prior to accessing specific registers.

16.7.13 Calling device internal processing

This section explains the procedure to call the device internal processing from the entry program.

Before calling the device internal processing, make sure that all the conditions of the hardware and software environments are satisfied and that the necessary arguments and RAM parameters have been set. Call the device internal processing by setting the FLSPM bit of the flash programming mode control register (FLPMC) to 1 and then executing the trap 0x1f instruction. The processing is always called using the same procedure. It is assumed that the program of this interface is described in an assembly language.

<1> Set the FLPMC register as follows:
- VPPDIS bit = 0 (to enable writing/erasing flash memory)
- FLSPM bit = 1 (to select self-programming mode)

<2> Clear the NP bit of the PSW to 0 (to enable NMIs (only when NMIs are used on the application)).

<3> Execute trap 0x1f to transfer the control to the device's internal processing.

<4> Set the NP bit and ID bit of the PSW to 1 (to disable all interrupts).

<5> Set the value to the peripheral command register (PHCMD) that is to be set to the FLPMC register.

<6> Set the FLPMC register as follows:
- VPPDIS bit = 1 (to disable writing/erasing flash memory)
- FLSPM bit = 0 (to select normal operation mode)

<7> Wait for the internal manipulation setup time (see **16.7.13 (5) Internal manipulation setup parameter**).

(1) Parameter

r6: First argument (sets a self-programming function number)
r7: Second argument
r8: Third argument
r9: Fourth argument
ep: First address of RAM parameter

(2) Return value

r10: Return value (return value from device internal processing of 4 bytes)

ep+4:Bit 7: NMI flag (flag indicating whether an NMI occurred while the device internal processing was being executed)

0: NMI did not occur while device internal processing was being executed.
1: NMI occurred while device internal processing was being executed.

If an NMI occurs while control is being transferred to the device internal processing, the NMI request may never be reflected. Because the NMI flag is not internally reset, this bit must be cleared before calling the device internal processing. After the control returns from the device internal processing, NMI dummy processing can be executed by checking the status of this flag using software.

(3) Description

Transfer control to the device internal processing specified by a function number using the trap instruction. To do this, the hardware and software environmental conditions must be satisfied. Even if trap 0x1f is used in the user application program, trap 0x1f is treated as another operation after the FLPMC register has been set. Therefore, use of the trap instruction is not restricted on the application.

(4) Program example

An example of a program in which the entry program is executed as a subroutine is shown below. In this example, the return address is saved to the stack and then the device internal processing is called. This program must be located in memory other than the block 0 space and flash memory area.

```
ISETUP     52                              -- Internal manipulation setup parameter
EntryProgram:
   add         -4, sp                      -- Prepare
   st.w        lp, 0[sp]                   -- Save return address
   movea       lo(0x00a0), r0, r10         --
   ldsr        r10, 5                      -- PSW = NP, ID
   mov         lo(0x0002), r10             --
   st.b        r10, PRCMD[r0]              -- PRCMD = 2
   st.b        r10, FLPMC[r0]              -- VPPDIS = 0, FLSPM = 1
   nop
   nop
   nop
   nop
   nop
   movea       lo(0x0020), r0, r10         --
   ldsr        r10, 5                      -- PSW = ID
   trap        0x1f                        -- Device Internal Process
   movea       lo(0x00a0), r0, r6          --
   ldsr        r6, 5                       -- PSW = NP, ID
   mov         lo(0x08), r6
   st.b        r6, PRCMD[r0]               -- PRCMD = 8
   st.b        r6, FLPMC[r0]               -- VPPDIS = 1, FLSPM = 0
   nop
   nop
   nop
   nop
   nop
   mov         ISETUP, lp                  -- loop time = 52
loop:
   divh        r6, r6                      -- To kill time
   add         -1, lp                      -- Decrement counter
   jne         loop                        --
   ld.w        0[sp], lp                   -- Reload lp
   add         4, sp                       -- Dispose
   jmp         [lp]                        -- Return to caller
```

(5) Internal manipulation setup parameter

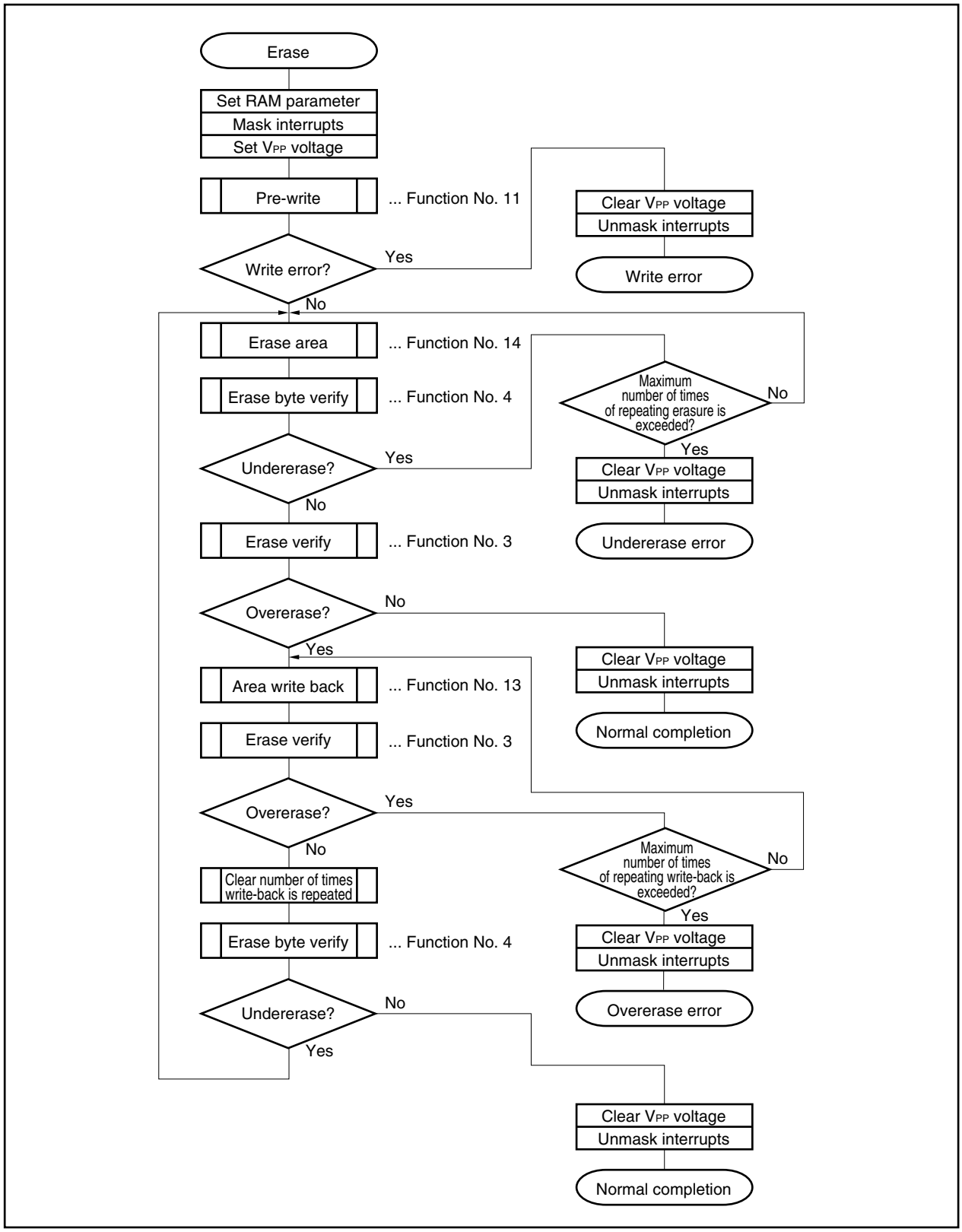
If the self-programming mode is switched to the normal operation mode, the V850/SA1 must wait for 100 μs before it accesses the flash memory. In the program example in (4) above, the elapse of this wait time is ensured by setting ISETUP to “52” (@ 20 MHz operation). The total number of execution clocks in this example is 39 clocks (divh instruction (35 clocks) + add instruction (1 clock) + jne instruction (3 clocks)). Ensure that a wait time of 100 μs elapses by using the following expression.

39 clocks (total number of execution clocks) $\times$ 50 ns (@ 20 MHz operation) $\times$ 52 (ISETUP) = 101.4 μs (wait time)

16.7.14 Flow of erasing flash memory

The procedure to erase the flash memory is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

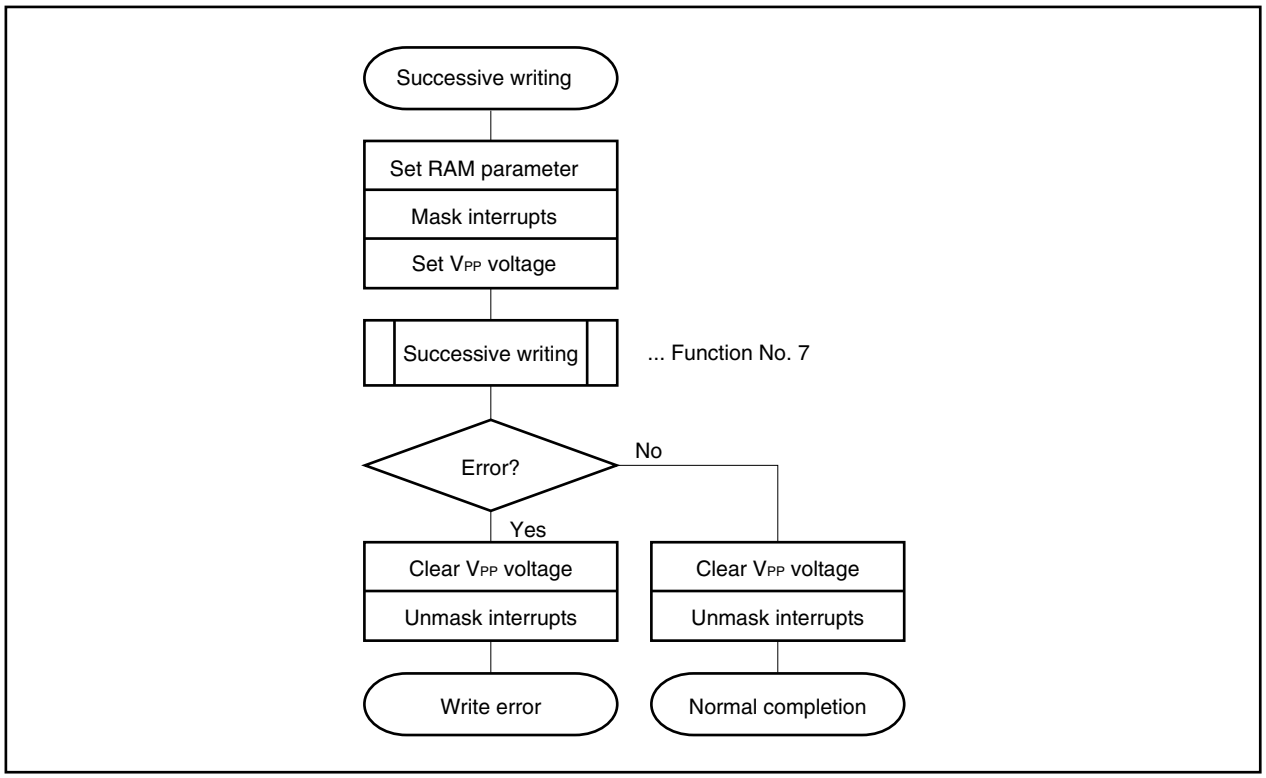
Figure 16-19. Flow of Erasing Flash Memory



16.7.15 Successive writing flow

The procedure to write data all at once to the flash memory by using the function to successively write data in word units is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

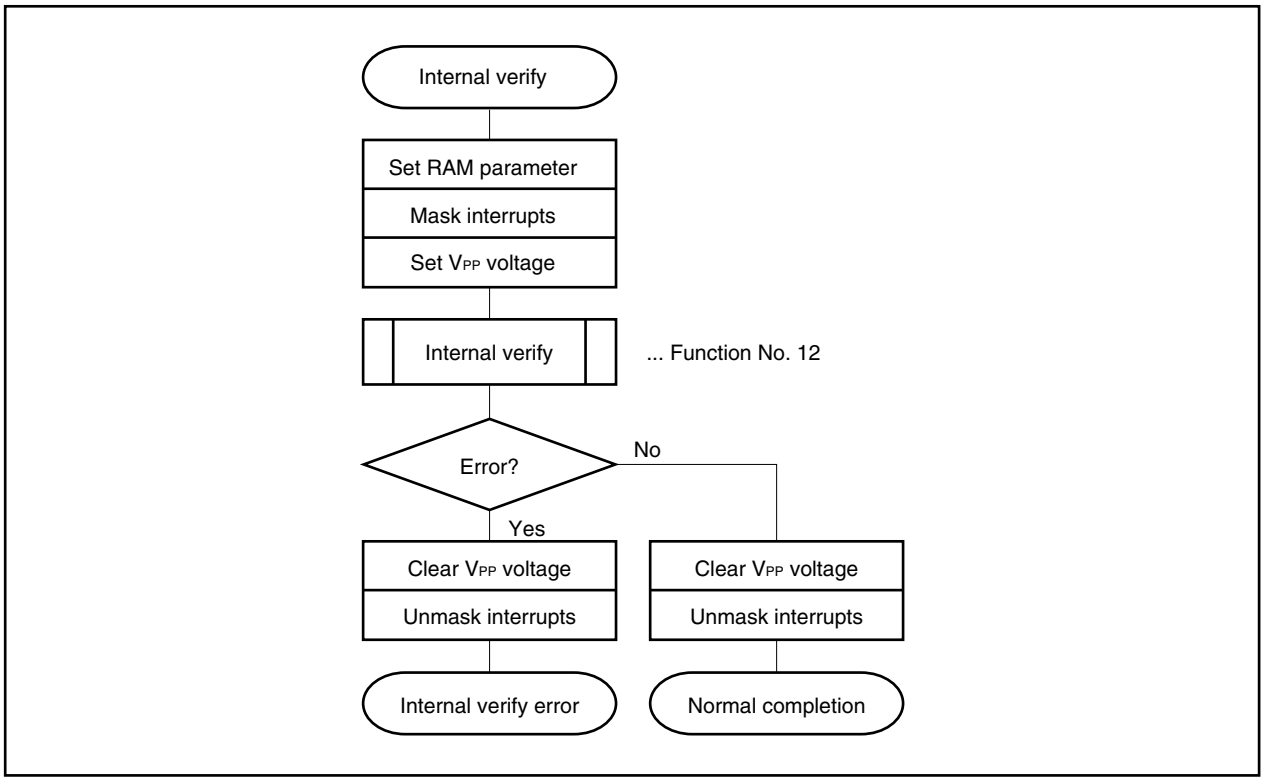
Figure 16-20. Successive Writing Flow



16.7.16 Internal verify flow

The procedure of internal verification is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

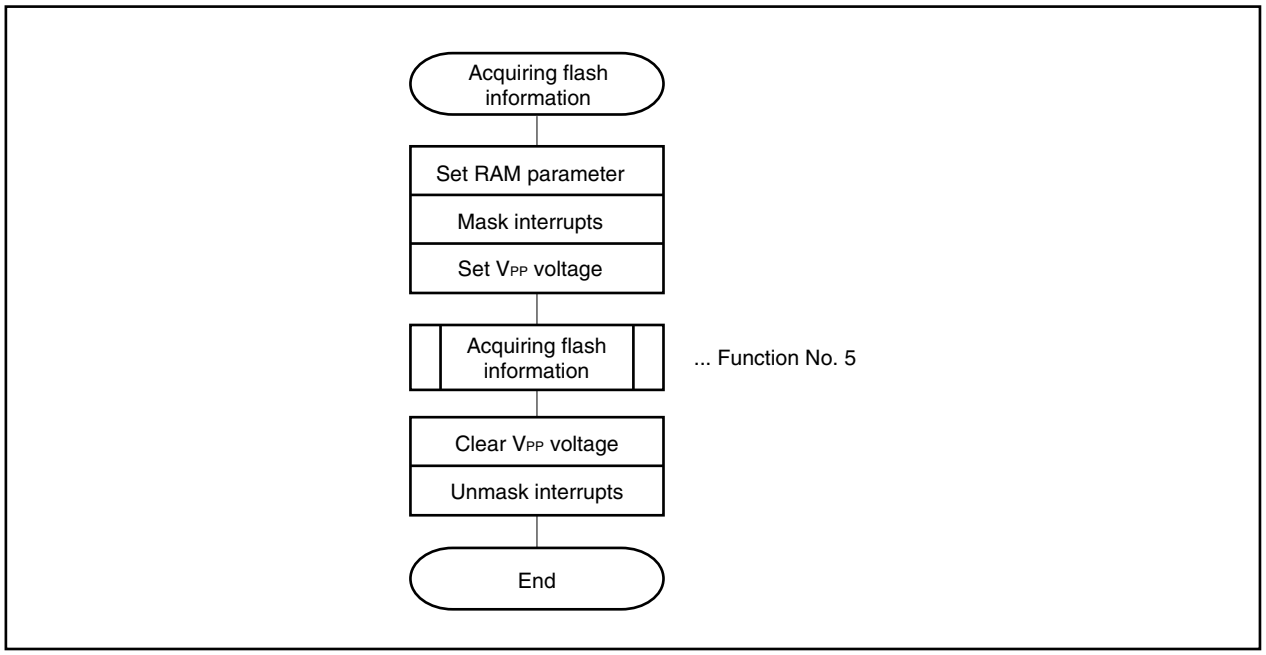
Figure 16-21. Internal Verify Flow



16.7.17 Flow of acquiring flash information

The procedure to acquire the flash information is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

Figure 16-22. Flow of Acquiring Flash Information



16.7.18 Self-programming library

The **V850 Series Flash Memory Self Programming Library User's Manual** is available for reference when executing self-programming.

In this manual, the library uses the self-programming interface of the V850 Series and can be used in C as a utility and as part of the application program. When using the library, thoroughly evaluate it on the application system.

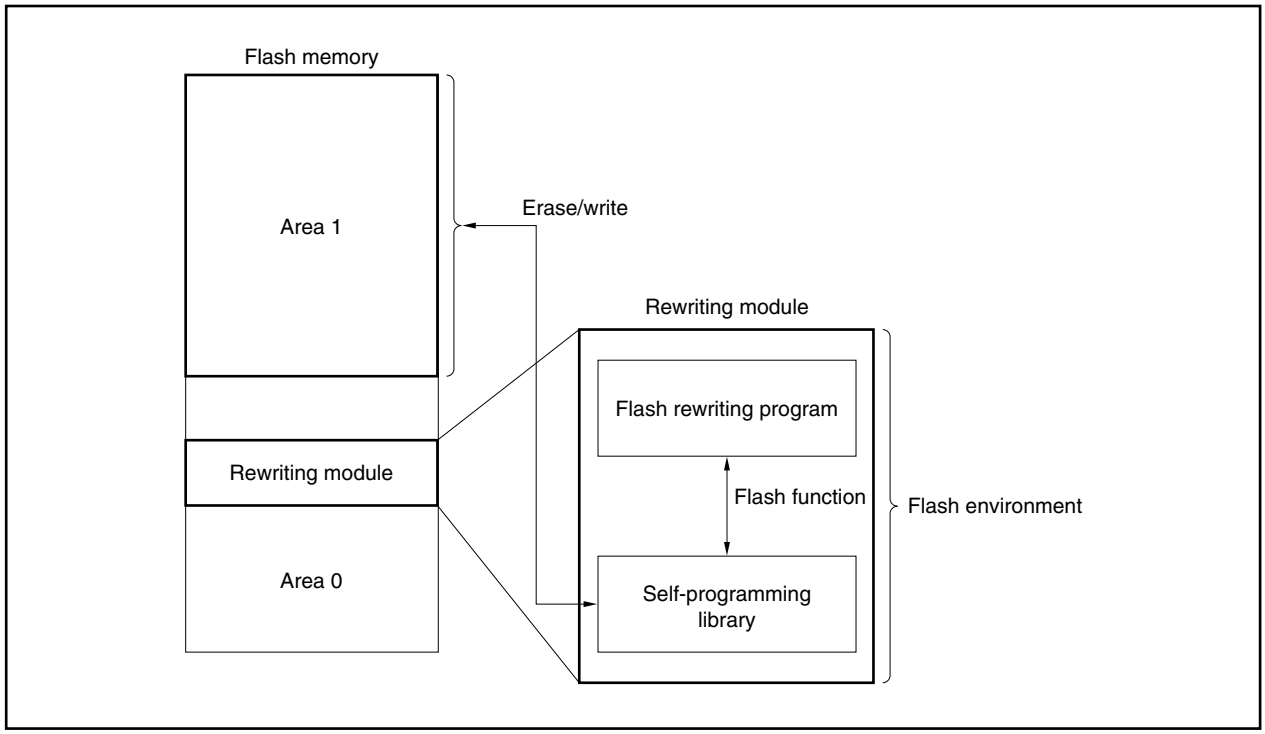
(1) Functional outline

Figure 16-23 outlines the function of the self-programming library. In this figure, a rewriting module is located in area 0 and the data in area 1 is rewritten or erased.

The rewriting module is a user program to rewrite the flash memory. The other areas can also be rewritten by using the flash functions included in this self-programming library. The flash functions expand the entry program in the external memory or internal RAM and call the device internal processing.

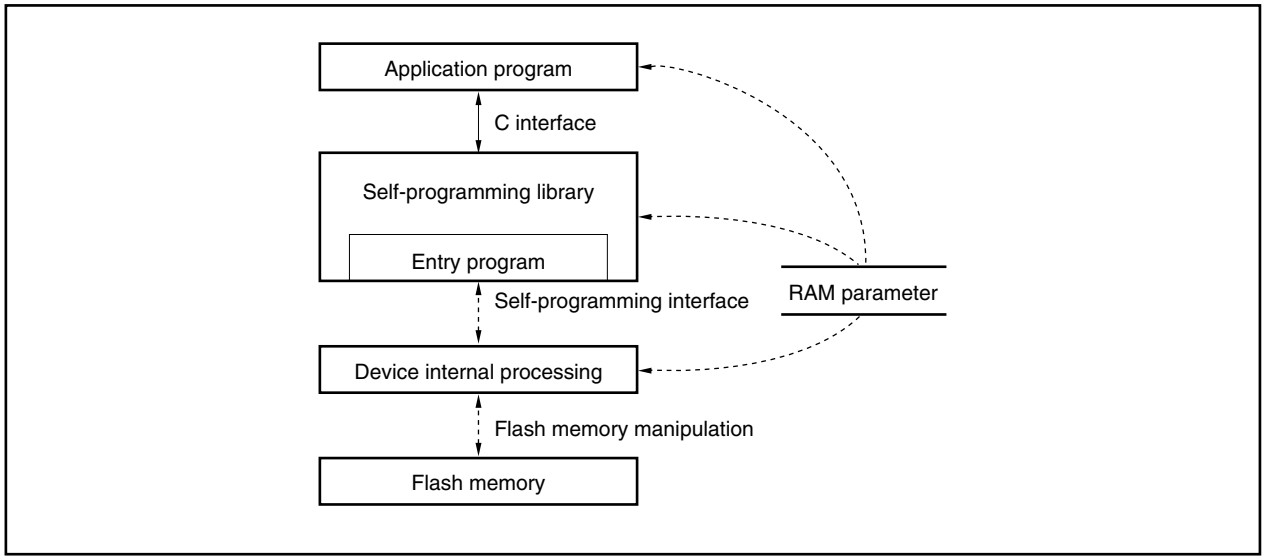
When using the self-programming library, make sure that the hardware conditions, such as the write voltage, and the software conditions, such as interrupts, are satisfied.

Figure 16-23. Functional Outline of Self-Programming Library



The configuration of the self-programming library is outlined below.

Figure 16-24. Outline of Self-Programming Library Configuration



Absolute Maximum Ratings (T_A = 25°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +4.6	V
	V_{PP}	**Notes 1, 2**	–0.5 to +8.5	V
	AV_{DD}		–0.5 to +4.6	V
	BV_{DD}		–0.5 to +4.6	V
	V_{SS}		–0.5 to +0.5	V
	AV_{SS}		–0.5 to +0.5	V
	BV_{SS}		–0.5 to +0.5	V
Input voltage	V_{I1}	**Note 3**, P114, $\overline{\text{RESET}}$	–0.5 to V_{DD} + 0.5 Note 6	V
	V_{I2}	**Note 4**	–0.5 to BV_{DD} + 0.5 Note 6	V
Clock input voltage	V_K	X1, XT1, XT2, V_{DD} = 2.7 to 3.6 V	–0.5 to V_{DD} + 1.0 Note 6	V
Analog input voltage	V_{IAN}	**Note 5** (AV_{DD})	–0.5 to AV_{DD} + 0.5 Note 6	V
Analog reference input voltage	AV_{REF}	AV_{REF}	–0.5 to AV_{DD} + 0.5 Note 6	V
Output current, low	I_{OL}	Per pin	4.0	mA
		Total for P00 to P07, P10 to P15, P20 to P25	25	mA
		Total for P26, P27, P30 to P37, P100 to P107, P110 to P113	25	mA
		Total for P40 to P47, P90 to P96, P120, CLKOUT	25	mA
		Total for P50 to P57, P60 to P65	25	mA
Output current, high	I_{OH}	Per pin	–4.0	mA
		Total for P00 to P07, P10 to P15, P20 to P25	–25	mA
		Total for P26, P27, P30 to P37, P100 to P107, P110 to P113	–25	mA
		Total for P40 to P47, P90 to P96, P120, CLKOUT	–25	mA
		Total for P50 to P57, P60 to P65	–25	mA
Output voltage	V_{O1}	**Note 3**, V_{DD} = 2.7 to 3.6 V	–0.5 to V_{DD} + 0.5 Note 6	V
	V_{O2}	**Note 4**, CLKOUT, BV_{DD} = 2.7 to 3.6 V	–0.5 to BV_{DD} + 0.5 Note 6	V
Operating ambient temperature	T_A	Normal operating mode	–40 to +85	°C
		Flash memory programming mode: 100 times guaranteed Note 7	0 to 85	°C
		Flash memory programming mode: 20 times guaranteed	10 to 40	°C
Storage temperature	T_{stg}	**Note 8**	–65 to +150	°C
		Note 1	–40 to +125	°C

Notes 1. μPD70F3015B, 70F3015BY, 70F3017A, and 70F3017AY only

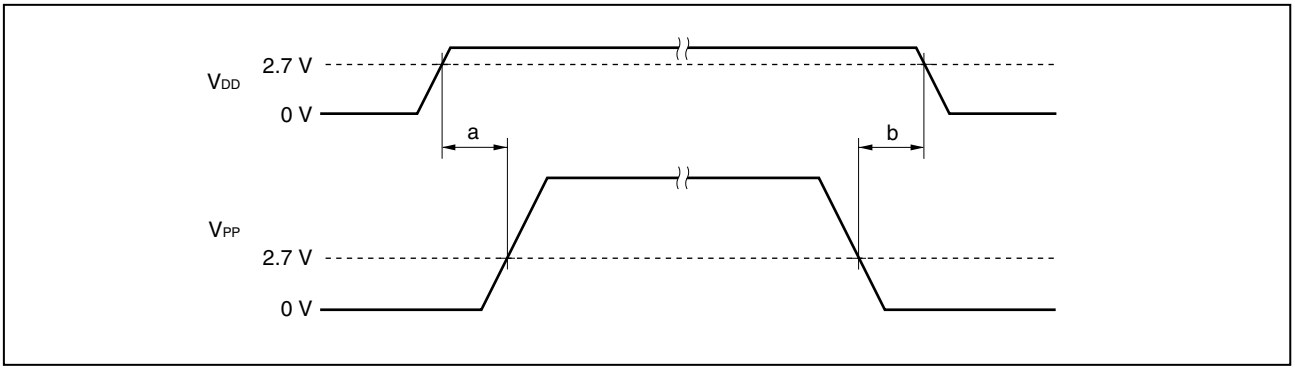
2. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when programming flash memory.

- **When supply voltage rises**

  V_{PP} must exceed V_{DD} 10 μs or more after V_{DD} reached the lower-limit value (2.7 V) of the operating voltage range (see "a" in the figure below).

- **When supply voltage drops**

  V_{DD} must be lowered 10 μs or more after V_{PP} falls below the lower-limit value (2.7 V) of the operating voltage range of V_{DD} (see "b" in the figure below).



Notes 3. P00 to P07, P10 to P15, P20 to P27, P30 to P37, P100 to P107, P110 to P113, P120, and their alternate-function pins.

4. P40 to P47, P50 to P57, P60 to P65, P90 to P96, and their alternate-function pins.

5. P70 to P77, P80 to P83, and their alternate-function pins.

6. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

7. The versions that guarantee 20 flash memory rewrites can be distinguished from the versions that guarantee 100 flash memory rewrites according to the product or the lot number stamped on the package (xxxx indicates the four-digit number or symbol for internal management).

- μPD70F3015B, 70F3015BY: Only products that guarantee 100 rewrites

- μPD70F3017A, 70F3017AY

	20 Rewrites Guaranteed	100 Rewrites Guaranteed
Lot No.	0135Mxxxx or earlier	0136Mxxxx or later
Flash memory rewrite count	20 rewrites	100 rewrites
Flash memory rewrite temperature	10 to 40°C	0 to 85°C

- About lot No.

01 36 M4001

- M4001 → Internal management number
- 36 → Product week code number (Example: 36th week)
- 01 → Product year code number (Example: 2001)

8. μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, and 703017AY only.

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-connector 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} = BV_{DD} = V_{SS} = AV_{SS} = BV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz Unmeasured pins returned to 0 V			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

Operating Conditions

(1) Operating frequency, operating voltage

Internal Operation Clock Frequency	Supply Voltage (V_{DD})
2 MHz ≤ f_{XX} ≤ 17 MHz	2.7 to 3.6 V
2 MHz ≤ f_{XX} ≤ 20 MHz	3.0 to 3.6 V
f_{XT} = 32.768 kHz	2.7 to 3.6 V

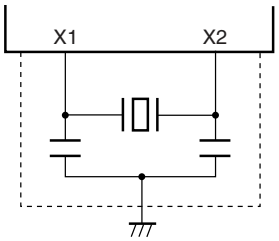
(2) CPU Operating frequency

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
CPU operating frequency	f_{CPU}	Operation with main clock	V_{DD} = 2.7 to 3.6 V	0.25		17	MHz
			V_{DD} = 3.0 to 3.6 V	0.25		20	MHz
		Operation with subclock	V_{DD} = 2.7 to 3.6 V		32.768		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_{XX}	V_{DD} = 2.7 to 3.6 V	2		17	MHz
		V_{DD} = 3.0 to 3.6 V	2		20	MHz
Oscillation stabilization time		Upon reset release		$2^{19}/f_{XX}$		s
		Upon STOP mode release		**Note**		s

Note The TYP value differs depending on the setting of the oscillation stabilization time select register (OSTS).

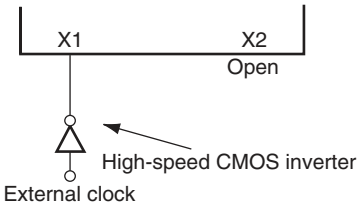
Caution Ensure that the duty of oscillation waveform is between 45% and 55%.

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 required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

(b) External clock input



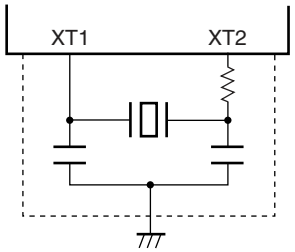
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_{XX}	V_{DD} = 2.7 to 3.6 V	2		17	MHz
		V_{DD} = 3.0 to 3.6 V	2		20	MHz

Cautions 1. Connect the high-speed CMOS inverter as close as possible to the X1 pin.

2. Sufficiently evaluate the matching between the V850/SA1 and the high-speed CMOS inverter.

(2) Subclock oscillator (T_A = –40 to +85°C)

(a) Connection of crystal resonator



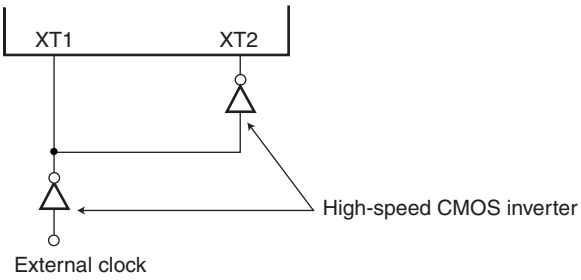
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{XT}	V_{DD} = 2.7 to 3.6 V	32	32.768	35	kHz
Oscillation stabilization time				10		s

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 required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

(b) External clock input



Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_{XT}	V_{DD} = 2.7 to 3.6 V	32	32.768	35	kHz

Cautions 1. Connect the high-speed CMOS inverter as close as possible to the XT2 pin.

2. Sufficiently evaluate the matching between the V850/SA1 and the high-speed CMOS inverter.

DC Characteristics

(1) Operating Conditions (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	Pins other than below		0.7V_{DD}		V_{DD}	V
	V_{IH2}	**Note 1**		0.7AV_{DD}		AV_{DD}	V
	V_{IH3}	**Note 2**		0.75V_{DD}		V_{DD}	V
	V_{IH4}	X1, XT1 (P114), XT2		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	Pins other than below		V_{SS}		0.3V_{DD}	V
	V_{IL2}	**Note 1**		AV_{SS}		0.3AV_{DD}	V
	V_{IL3}	**Note 2**		V_{SS}		0.2V_{DD}	V
	V_{IL4}	X1, XT1 (P114), XT2		V_{SS}		0.2V_{DD}	V
Output voltage, high	V_{OH1}	**Note 3**	I_{OH} = –3 mA	0.8V_{DD}			V
	V_{OH2}	**Note 4**	I_{OH} = –1 mA	0.8V_{DD}			V
Output voltage, low	V_{OL1}	**Note 3**	I_{OL} = 1.6 mA			0.4	V
	V_{OL2}	**Note 4** (Except pins P10 and P12)	I_{OL} = 1.6 mA			0.4	V
	V_{OL3}	P10, P12	I_{OL} = 3 mA			0.4	V
V_{PP} supply voltage[Note 5]	V_{PP1}	Normal operation		0		0.2V_{DD}	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD} = AV_{DD} = BV_{DD}	Pins other than below			5	μA
	I_{LIH2}		X1, XT1, XT2			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Pins other than below			–5	μA
	I_{LIL2}		X1, XT1, XT2			–20	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD} = AV_{DD} = BV_{DD}				5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–5	μA
Pull-up resistance	R_L	V_{IN} = 0 V		10	30	100	kΩ

Notes 1. P70 to P77, P80 to P83, and their alternate-function pins.

2. P00 to P07, P10, P12, P13, P15, P20, P22, P23, P25 to P27, P30 to P33, P36, P37, RESET, and their alternate-function pins.

3. CLKOUT, P40 to P47, P50 to P57, P60 to P65, P90 to P96, P120, and their alternate-function pins.

4. P00 to P07, P10 to P15, P20 to P27, P30 to P37, P100 to P107, P110 to P113, and their alternate-function pins.

5. μPD70F3015B, 70F3015BY, 70F3017A, 70F3017AY only.

(1) Operating conditions (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V) (2/2)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current[Note]	μPD703014A, μPD703014AY, μPD703014B, μPD703014BY, μPD703015A, μPD703015AY, μPD703015B, μPD703015BY, μPD703017A, μPD703017AY	I_{DD1}	Normal operation	f_{XX} = 17 MHz All peripheral functions operating		17	30	mA
		I_{DD2}	HALT mode	f_{XX} = 17 MHz All peripheral functions operating		8	20	mA
		I_{DD3}	IDLE mode	f_{XX} = 17 MHz Watch timer operating		1	4	mA
		I_{DD4}	Software STOP mode (subclock operation @f_{XT} = 32.768 kHz, watch timer operating)			8	60	μA
			Software STOP mode (subclock stopped (XT1 = V_{SS}))			1	60	μA
		I_{DD5}	Subclock normal operation mode f_{XT} = 32.768 kHz (main clock stopped)			40	140	μA
		I_{DD6}	Subclock IDLE mode f_{XT} = 32.768 kHz (main clock stopped, watch timer operating)			8	60	μA
	μPD70F3015B, μPD70F3015BY, μPD70F3017A, μPD70F3017AY	I_{DD1}	Normal operation	f_{XX} = 17 MHz All peripheral functions operating		30	60	mA
		I_{DD2}	HALT mode	f_{XX} = 17 MHz All peripheral functions operating		10	25	mA
		I_{DD3}	IDLE mode	f_{XX} = 17 MHz Watch timer operating		4	8	mA
		I_{DD4}	Software STOP mode (subclock operating@f_{XT}= 32.768 kHz, watch timer operating)			10	100	μA
			Software STOP mode (subclock, stopped (XT1 = V_{SS}))			2	100	μA
		I_{DD5}	Subclock normal operation mode f_{XT} = 32.768 kHz (main clock stopped)			250	600	μA
		I_{DD6}	Subclock IDLE mode f_{XT} = 32.768 kHz (main clock stopped, watch timer operating)			130	360	μA

Note The TYP. value of V_{DD} is 3.3 V. The current consumed by the output buffer is not included.

(2) Operating Conditions (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	Pins other than below		0.7V_{DD}		V_{DD}	V
	V_{IH2}	**Note 1**		0.7AV_{DD}		AV_{DD}	V
	V_{IH3}	**Note 2**		0.75V_{DD}		V_{DD}	V
	V_{IH4}	X1, XT1 (P114), XT2		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	Pins other than below		V_{SS}		0.3V_{DD}	V
	V_{IL2}	**Note 1**		AV_{SS}		0.3AV_{DD}	V
	V_{IL3}	**Note 2**		V_{SS}		0.2V_{DD}	V
	V_{IL4}	X1, XT1 (P114), XT2		V_{SS}		0.2V_{DD}	V
Output voltage, high	V_{OH1}	**Note 3**	I_{OH} = –3 mA	0.8V_{DD}			V
	V_{OH2}	**Note 4**	I_{OH} = –1 mA	0.8V_{DD}			V
Output voltage, low	V_{OL1}	**Note 3**	I_{OL} = 1.6 mA			0.4	V
	V_{OL2}	**Note 4** (Except pins P10 and P12)	I_{OL} = 1.6 mA			0.4	V
	V_{OL3}	P10, P12	I_{OL} = 3 mA			0.4	V
V_{PP} supply voltage[Note 5]	V_{PP1}	Normal operation		0		0.2V_{DD}	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD} = AV_{DD} = BV_{DD}	Pins other than below			5	μA
	I_{LIH2}		X1, XT1, XT2			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Pins other than below			–5	μA
	I_{LIL2}		X1, XT1, XT2			–20	μA
Output leakage current, high	I_{LOH1}	V_O = V_{DD} = AV_{DD} = BV_{DD}				5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–5	μA
Pull-up resistance	R_L	V_{IN} = 0 V		10	30	100	kΩ

Notes 1. P70 to P77, P80 to P83, and their alternate-function pins.

2. P00 to P07, P10, P12, P13, P15, P20, P22, P23, P25 to P27, P30 to P33, P36, P37, $\overline{\text{RESET}}$ and their alternate-function pins.

3. CLKOUT, P40 to P47, P50 to P57, P60 to P65, P90 to P96, P120, and their alternate-function pins.

4. P00 to P07, P10 to P15, P20 to P27, P30 to P37, P100 to P107, P110 to P113, and their alternate-function pins.

5. μPD70F3015B, 70F3015BY, 70F3017A, 70F3017AY only.

(2) Operating conditions (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V) (2/2)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current[Note 5]	μPD703014A, μPD703014AY, μPD703014B, μPD703014BY, μPD703015A, μPD703015AY, μPD703015B, μPD703015BY, μPD703017A, μPD703017AY	I_{DD1}	Normal operation	f_{XX} = 20 MHz All peripheral functions operating		20	35	mA
		I_{DD2}	HALT mode	f_{XX} = 20 MHz All peripheral functions operating		9	22	mA
		I_{DD3}	IDLE mode	f_{XX} = 20 MHz Watch timer operating		1.2	4.5	mA
		I_{DD4}	Software STOP mode (subclock operation @f_{XT} = 32.768 kHz, watch timer operating)			8	60	μA
			Software STOP mode (subclock stopped (XT1 = V_{SS}))			1	60	μA
		I_{DD5}	Subclock normal operation mode f_{XT} = 32.768 kHz (main clock stopped)			40	140	μA
		I_{DD6}	Subclock IDLE mode f_{XT} = 32.768 kHz (main clock stopped, watch timer operating)			8	60	μA
	μPD70F3015B, μPD70F3015BY, μPD70F3017A, μPD70F3017AY	I_{DD1}	Normal operation	f_{XX} = 20 MHz All peripheral functions operating		32	64	mA
		I_{DD2}	HALT mode	f_{XX} = 20 MHz All peripheral functions operating		11	26	mA
		I_{DD3}	IDLE mode	f_{XX} = 20 MHz Watch timer operating		4.5	9	mA
		I_{DD4}	Software STOP mode (subclock operating@f_{XT} = 32.768 kHz, watch timer operating)			10	100	μA
			Software STOP mode (subclock stopped (XT1 = V_{SS}))			2	100	μA
		I_{DD5}	Subclock normal operation mode f_{XT} = 32.768 kHz (main clock stopped)			250	600	μA
		I_{DD6}	Subclock IDLE mode f_{XT} = 32.768 kHz (main clock stopped, watch timer operating)			130	360	μA

Note The TYP. value of V_{DD} is 3.3 V. The current consumed by the output buffer is not included.

Data Retention Characteristics (T_A = –40 to +85°C, V_{SS} = AV_{SS} = BV_{SS} = 0 V)

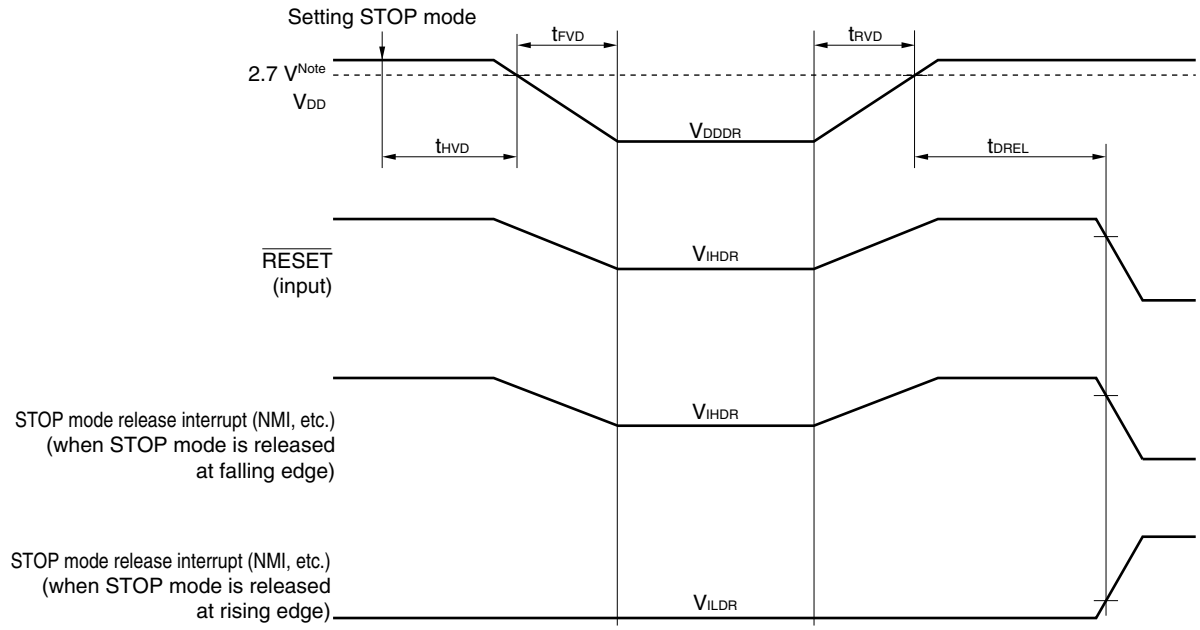
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode		1.8		3.6	V
Data retention current	I_{DDDR}	V_{DD} = V_{DDDR}, XT1 = V_{SS}	**Note 1**		1	60	μA
			Note 2		2	100	μA
Supply voltage rise time	t_{RVD}			200			μs
Supply voltage fall time	t_{FVD}			200			μs
Supply voltage hold time (from STOP mode setting)	t_{HVD}			0			ms
STOP mode release signal input time	t_{DREL}			0			ms
Data retention high-level input voltage	V_{IHDR}	All input ports		V_{IHn}		V_{DDDR}	V
Data retention low-level input voltage	V_{ILDR}	All input ports		0		V_{ILn}	V

Notes 1. μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, 703017AY only

2. μPD70F3015B, 70F3015BY, 70F3017A, 70F3017AY only

Remarks 1. TYP. values are reference values for when T_A = 25°C.

2. n = 1 to 4



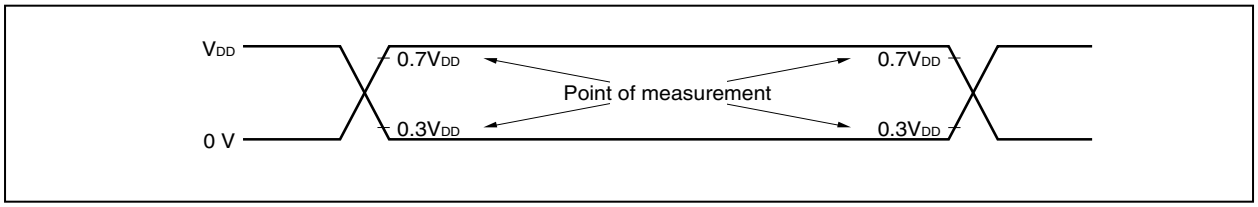
Note V_{DD} = 2.7 V indicates the minimum operating voltage of the V850/SA1 (when f_{XX} = 17 MHz).

Caution Shifting to STOP mode and restoring from STOP mode must be performed at V_{DD} = 2.7 V min. (f_{XX} = 17 MHz) and V_{DD} = 3.0 V min. (f_{XX} = 20 MHz), respectively.

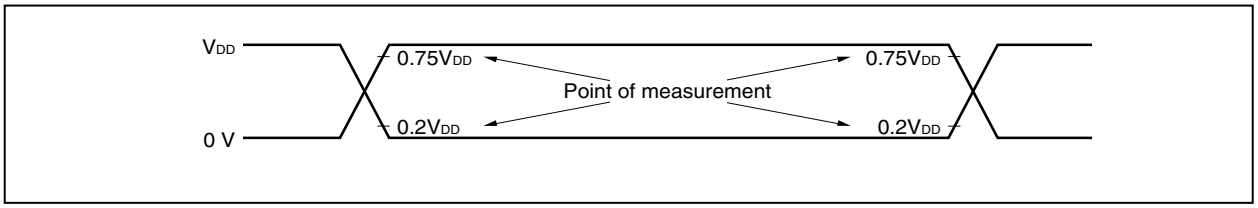
AC Characteristics

AC test input measurement points

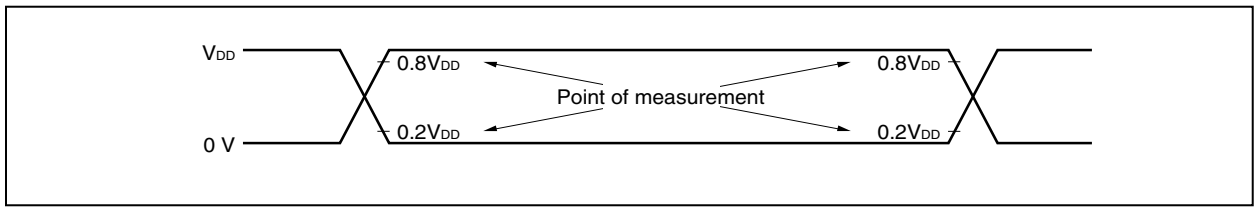
(1) P11, P14, P21, P24, P34, P35, P40 to P47, P50 to P57, P60 to P65, P90 to P96, P100 to P107, P110 to P113, P120, and their alternate-function pins



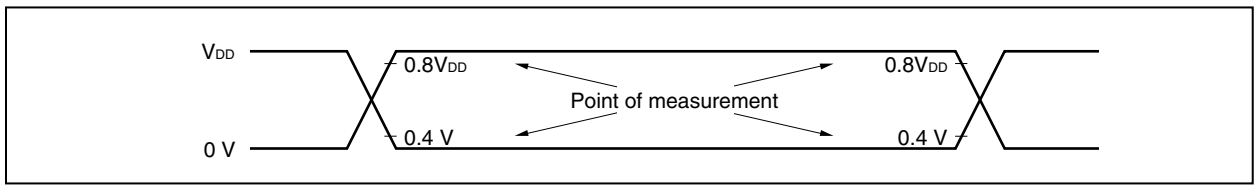
(2) P00 to P07, P10, P12, P13, P15, P20, P22, P23, P25 to P27, P30 to P33, P36, P37, $\overline{\text{RESET}}$, and their alternate-function pins



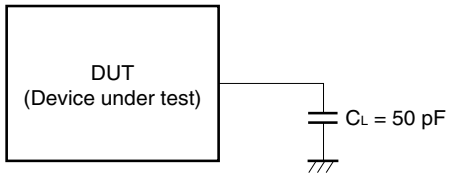
(3) X1, XT1 (P114), XT2



AC test output measurement points



Load conditions



Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring 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} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
X1 input cycle	t_{CYX}	<1>		58.8	500	ns
XT1 input cycle				28.5	31.2	μs
X1 input high-level width	t_{WXH}	<2>		26.4		ns
XT1 input high-level width				12.8		μs
X1 input low-level width	t_{WXL}	<3>		26.4		ns
XT1 input low-level width				12.8		μs
X1, XT1 input rise time	t_{XR}	<4>			0.5 (t_{CYX} – t_{WXH} – t_{WXL})	ns
X1, XT1 input fall time	t_{XF}	<5>			0.5 (t_{CYX} – t_{WXH} – t_{WXL})	ns
CLKOUT output cycle	t_{CYK}	<6>		58.8 ns	31.2 μs	
CLKOUT high-level width	t_{WKH}	<7>		0.4t_{CYK} – 10		ns
CLKOUT low-level width	t_{WKL}	<8>		0.4t_{CYK} – 10		ns
CLKOUT rise time	t_{KR}	<9>			10	ns
CLKOUT fall time	t_{KF}	<10>			10	ns

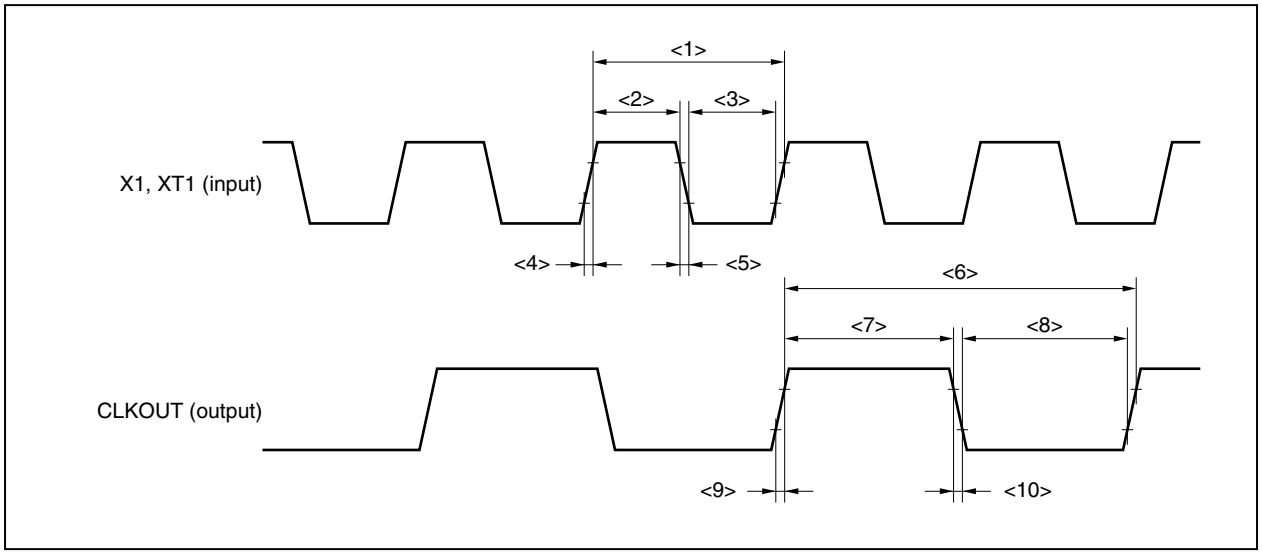
Remark Ensure that the duty is between 45% and 55%.

(2) Operating Conditions (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
X1 input cycle	t_{CYX}	<1>		50.0	500	ns
XT1 input cycle				28.5	31.2	μs
X1 input high-level width	t_{WXH}	<2>		22.5		ns
XT1 input high-level width				12.8		μs
X1 input low-level width	t_{WXL}	<3>		22.5		ns
XT1 input low-level width				12.8		μs
X1, XT1 input rise time	t_{XR}	<4>			0.5 (t_{CYX} – t_{WXH} – t_{WXL})	ns
X1, XT1 input fall time	t_{XF}	<5>			0.5 (t_{CYX} – t_{WXH} – t_{WXL})	ns
CLKOUT output cycle	t_{CYK}	<6>		50.0 ns	31.2 μs	
CLKOUT high-level width	t_{WKH}	<7>		0.4t_{CYK} – 10		ns
CLKOUT low-level width	t_{WKL}	<8>		0.4t_{CYK} – 10		ns
CLKOUT rise time	t_{KR}	<9>			10	ns
CLKOUT fall time	t_{KF}	<10>			10	ns

Remark Ensure that the duty is between 45% and 55%.

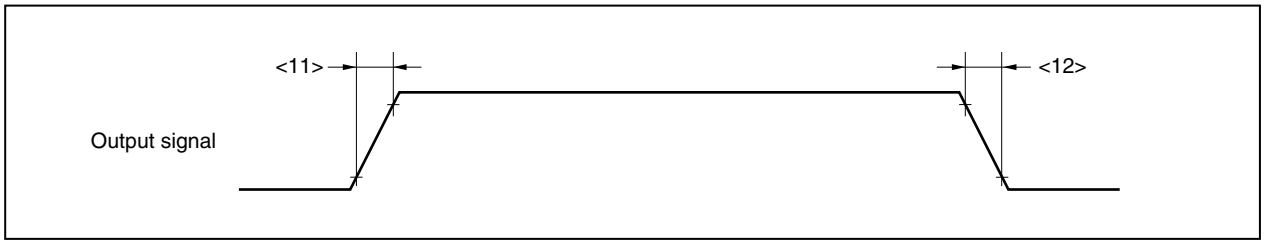
Clock Timing



Timing of pins other than CLKOUT, ports 4, 5, 6, and 9

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{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 (CLKOUT Asynchronous)

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	t_{SAST}	<13>		0.5T – 15		ns
Address hold time (from ASTB↓)	t_{HSTA}	<14>		0.5T – 15		ns
Address float delay time from $\overline{DSTB}$↓	t_{FDA}	<15>			2	ns
Data input setup time from address	t_{SAID}	<16>			(2 + n)T – 25	ns
Data input setup time from $\overline{DSTB}$↓	t_{SDID}	<17>			(1 + n)T – 25	ns
Delay time from ASTB↓ to $\overline{DSTB}$↓	t_{DSTD}	<18>		0.5T – 15		ns
Data input hold time (from $\overline{DSTB}$↑)	t_{HDID}	<19>		0		ns
Address output time from $\overline{DSTB}$↑	t_{DDA}	<20>		(1 + i)T – 15		ns
Delay time from $\overline{DSTB}$↑ to ASTB↑	t_{DDST1}	<21>		0.5T – 15		ns
Delay time from $\overline{DSTB}$↑ to ASTB↓	t_{DDST2}	<22>		(1.5 + i)T – 15		ns
$\overline{DSTB}$ low-level width	t_{WDL}	<23>		(1 + n)T – 15		ns
ASTB high-level width	t_{WSTH}	<24>		T – 15		ns
Data output time from $\overline{DSTB}$↓	t_{DDOD}	<25>			15	ns
Data output setup time (to $\overline{DSTB}$↑)	t_{SODD}	<26>		(1 + n)T – 20		ns
Data output hold time (from $\overline{DSTB}$↑)	t_{HDOD}	<27>		T – 15		ns
$\overline{WAIT}$ setup time (to address)	t_{SAWT1}	<28>	n ≥ 1		1.5T – 25	ns
	t_{SAWT2}	<29>	n ≥ 1		(1.5 + n)T – 25	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT1}	<30>	n ≥ 1	(0.5 + n)T		ns
	t_{HAWT2}	<31>	n ≥ 1	(1.5 + n)T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{SSTWT1}	<32>	n ≥ 1		T – 25	ns
	t_{SSTWT2}	<33>	n ≥ 1		(1 + n)T – 25	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT1}	<34>	n ≥ 1	nT		ns
	t_{HSTWT2}	<35>	n ≥ 1	(1 + n)T		ns
$\overline{HLDRQ}$ high-level width	t_{WHQH}	<36>		T + 10		ns
$\overline{HLDAK}$ low-level width	t_{WHAL}	<37>		T – 15		ns
Bus output delay time from $\overline{HLDAK}$↑	t_{DHAC}	<38>		0		ns
Delay time from $\overline{HLDRQ}$↓ to $\overline{HLDAK}$↓	t_{DHQHA1}	<39>			(2n + 7.5)T + 25	ns
Delay time from $\overline{HLDRQ}$↑ to $\overline{HLDAK}$↑	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. i: Number of idle states inserted after the read cycle (0 or 1).

4. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

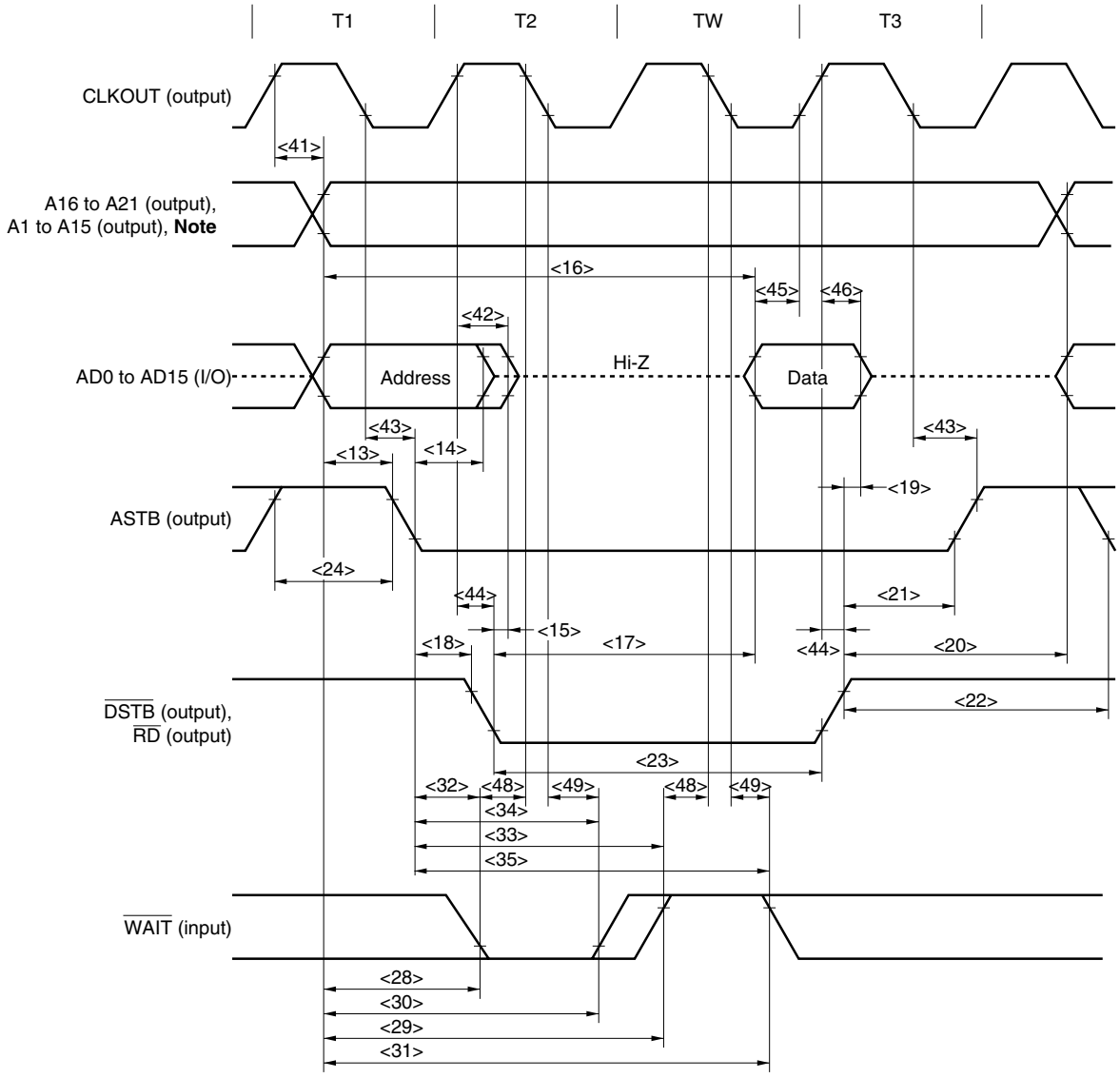
Bus Timing (CLKOUT Synchronous)

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{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>		–12	7	ns
Delay time from CLKOUT↓ to ASTB	t_{DKST}	<43>		–12	7	ns
Delay time from CLKOUT↑ to $\overline{DSTB}$	t_{DKD}	<44>		–5	14	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>		15		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT}	<49>		5		ns
$\overline{HLDRQ}$ setup time (to CLKOUT↓)	t_{SHQK}	<50>		15		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

Remark The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

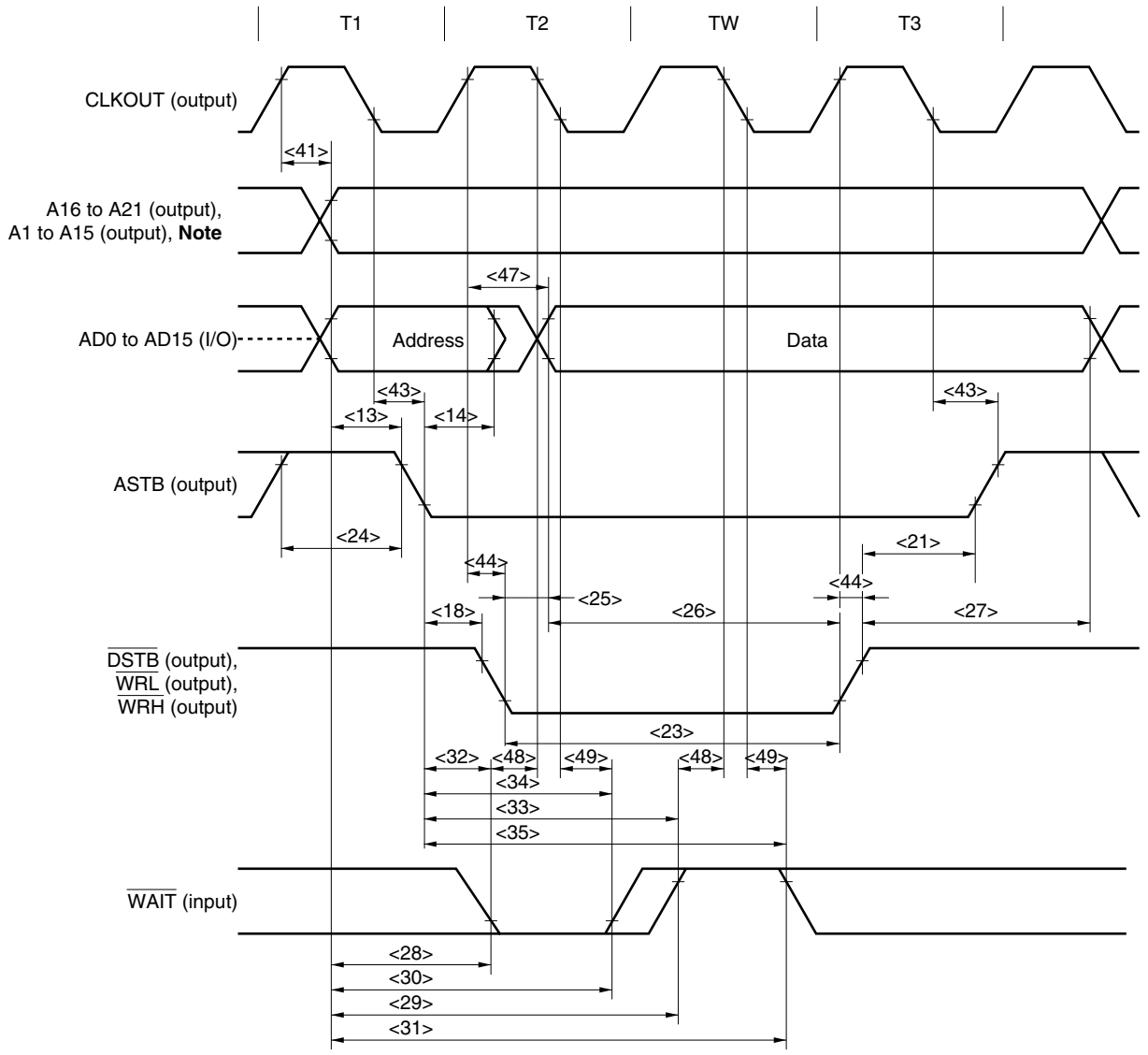
Read Cycle (CLKOUT Synchronous/Asynchronous, 1 Wait)



Note R/$\overline{W}$ (output), $\overline{UBEN}$ (output), $\overline{LBEN}$ (output)

Remark $\overline{WRL}$ and $\overline{WRH}$ are high level.

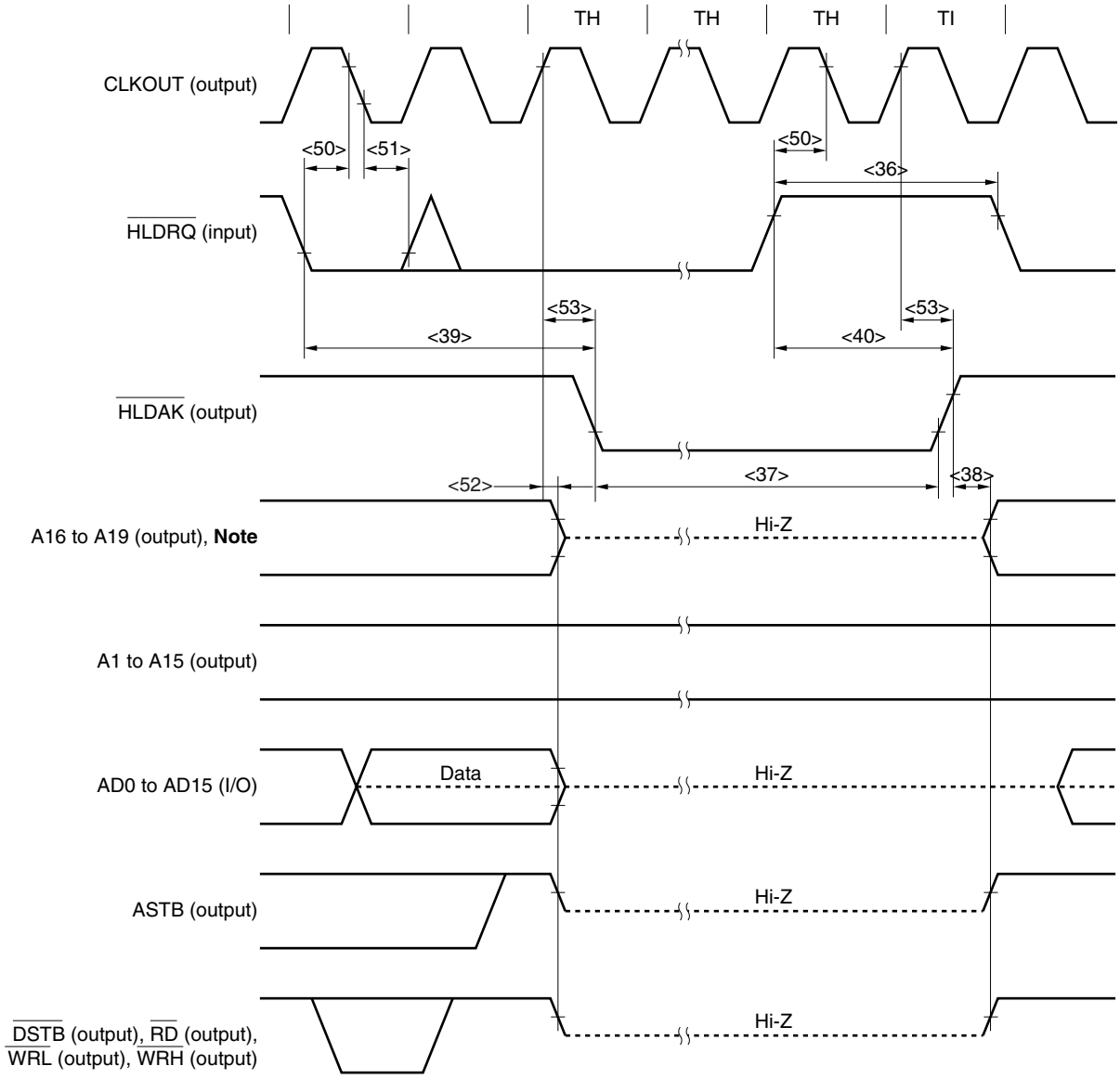
Write Cycle (CLKOUT Synchronous/Asynchronous, 1 Wait)



Note R/$\overline{W}$ (output), $\overline{UBEN}$ (output), $\overline{LBEN}$ (output)

Remark $\overline{RD}$ is high level.

Bus Hold



Note R/W (output), UBEN (output), LBEN (output)

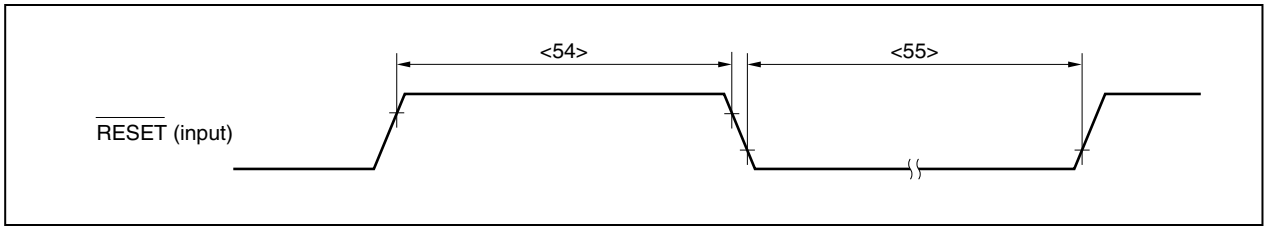
Reset/Interrupt Timing

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

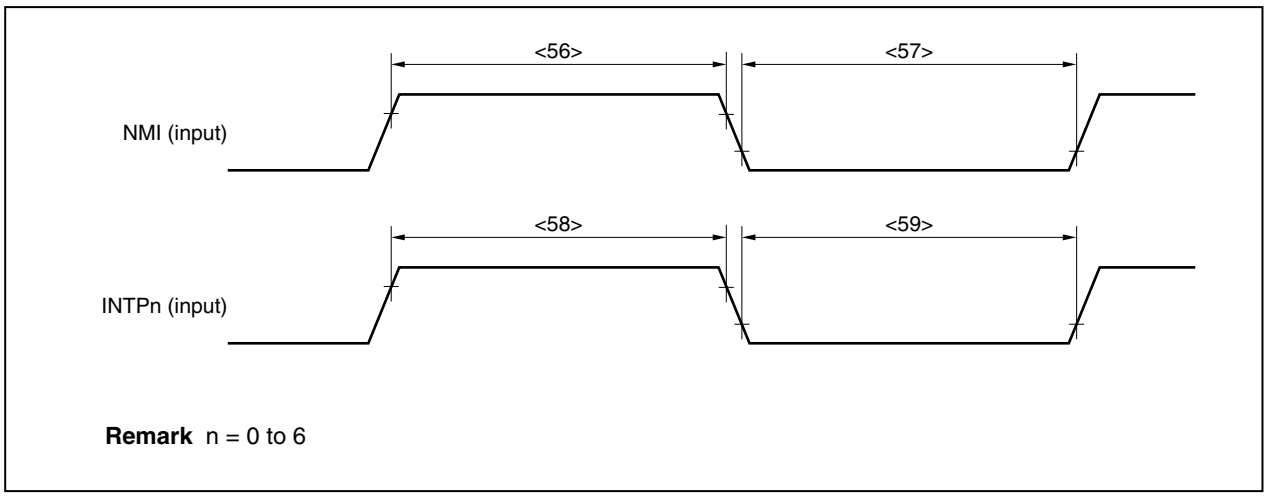
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<54>		500		ns
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<55>		500		ns
NMI high-level width	t_{WNIH}	<56>		500		ns
NMI low-level width	t_{WNIL}	<57>		500		ns
INTPn high-level width	t_{WITH}	<58>	n = 0 to 3 (analog noise elimination)	500		ns
			n = 4 to 6 (digital noise elimination)	3T + 20		ns
INTPn low-level width	t_{WITL}	<59>	n = 0 to 3 (analog noise elimination)	500		ns
			n = 4 to 6 (digital noise elimination)	3T + 20		ns

Remark T = 1/fxx

Reset



Interrupt



Remark n = 0 to 6

TIn Input Timing

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn0, TIn1 high-level width	t_{TIHn}	<60>	n = 0, 1	$2T_{sam}$ + 20[Note]		ns
TIn high-level width			n = 2 to 5	3T + 20		ns
TIn0, TIn1 low-level width	t_{TILn}	<61>	n = 0, 1	$2T_{sam}$ + 20[Note]		ns
TIn low-level width			n = 2 to 5	3T + 20		ns

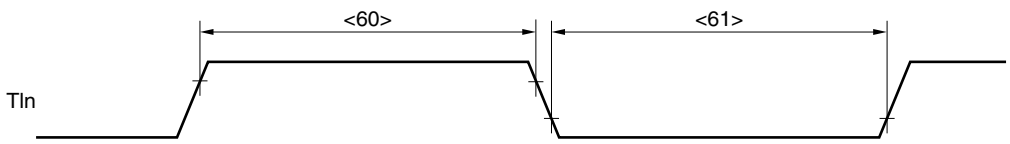
Note T_{sam} (count clock cycle) can be selected as follows by setting the PRMn2 to PRMn0 bits of prescaler mode register n, n1 (PRMn, PRMn1).

When n = 0 (TM0): T_{sam} = 2T, 4T, 16T, 64T, 256T or 1/INTWTI cycle

When n = 1 (TM1): T_{sam} = 2T, 4T, 16T, 32T, 128T, or 256T cycle

However, when the TIn0 valid edge is selected as the count clock, T_{sam} = 2T.

Remark T= 1/fxx



Remark n = 00, 01, 10, 11, 2 to 5

CSI Timing

(1) Master mode (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

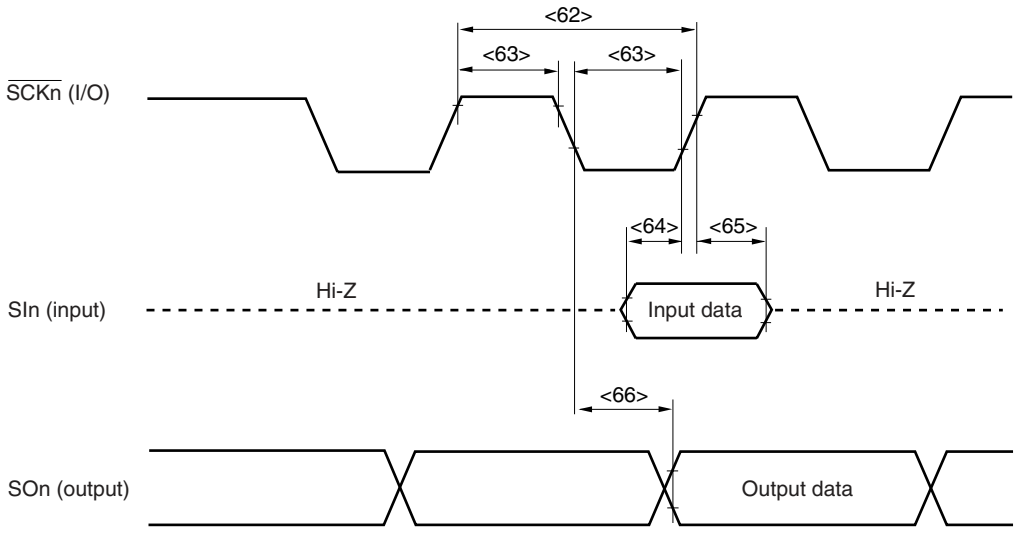
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle time	t_{KCY1}	<62>		400		ns
$\overline{SCKn}$ high-/low-level width	t_{KH1}, t_{KL1}	<63>		140		ns
SIn setup time (to $\overline{SCKn}\uparrow$)	t_{SIK1}	<64>		50		ns
SIn hold time (from $\overline{SCKn}\uparrow$)	t_{KSI1}	<65>		50		ns
Delay time from $\overline{SCKn}\downarrow$ to SOn output	t_{KSO1}	<66>			60	ns

Remark n = 0 to 2

(2) Slave mode (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle time	t_{KCY2}	<62>		400		ns
$\overline{SCKn}$ high-/low-level width	t_{KH2}, t_{KL2}	<63>		140		ns
SIn setup time (to $\overline{SCKn}\uparrow$)	t_{SIK2}	<64>		50		ns
SIn hold time (from $\overline{SCKn}\uparrow$)	t_{KSI2}	<65>		50		ns
Delay time from $\overline{SCKn}\downarrow$ to SOn output	t_{KSO2}	<66>			60	ns

Remark n = 0 to 2

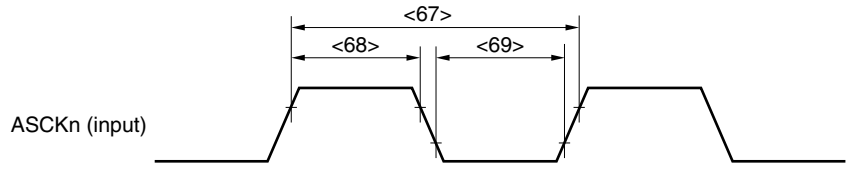


Remark n = 0 to 2

UART Timing (T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
ASCKn cycle time	t_{KCY13}	<67>		200		ns
ASCKn high-level width	t_{KH13}	<68>		80		ns
ASCKn low-level width	t_{KL13}	<69>		80		ns

Remark n = 0 or 1



Remark n = 0 or 1

I^2C Bus Mode (μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, 70F3017AY only)

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = 2.7 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V, C_L = 50 pF)

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}	<70>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<71>	4.0	–	0.6	–	μs
SCL clock low-level width		t_{LOW}	<72>	4.7	–	1.3	–	μs
SCL clock high-level width		t_{HIGH}	<73>	4.0	–	0.6	–	μs
Setup time for start/restart condition		$t_{SU:STA}$	<74>	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	<75>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<76>	250	–	100[Note 4]	–	ns
SDA and SCL signal rise time		t_R	<77>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA and SCL signal fall time		t_F	<78>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<79>	4.0	–	0.6	–	μs
Width of spike pulse suppressed by input filter		t_{SP}	<80>	–	–	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 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.

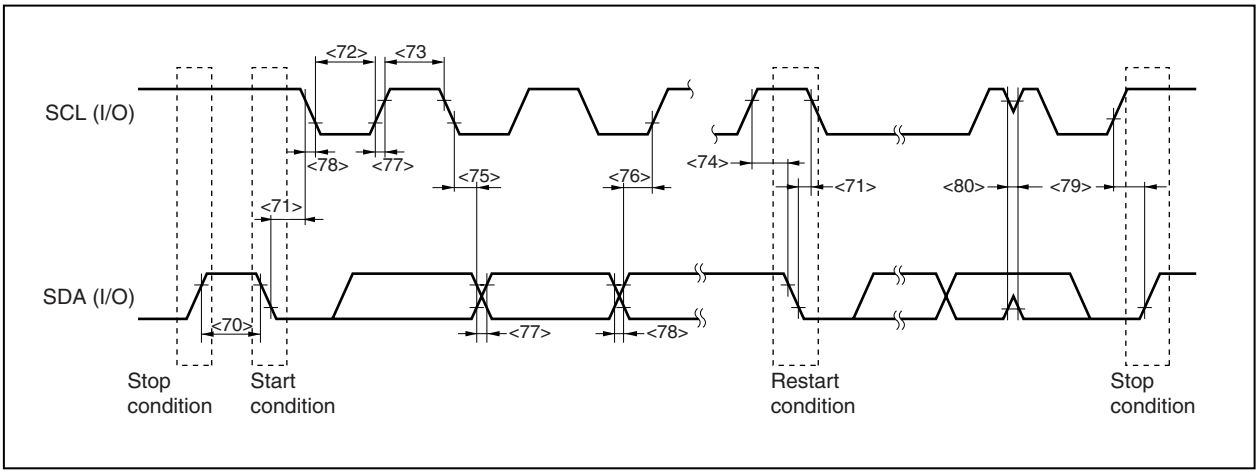
4. The high-speed mode I^2C bus can be used in the normal-mode I^2C bus system. In this case, set the high-speed mode I^2C bus so that it meets the following conditions.

- 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 following data bit to the SDA line prior to the SCL line release ($t_{Rmax.} + t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Normal mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

Remark The maximum operating frequency of the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and μPD70F3017AY is f_{XX} = 17 MHz.

I²C Bus Mode (μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, 70F3017AY only)



A/D Converter

(T_A = –40 to +85°C, V_{DD} = AV_{DD} = AV_{REF} = 2.7 to 3.6 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]					±0.8	%FSR
Conversion time	t_{CONV}		5		100	μs
Zero-scale error[Note 1]					±0.4	%FSR
Full-scale error[Note 1]					±0.4	%FSR
Integral linearity error[Note 2]					±4	LSB
Differential linearity error[Note 2]					±4	LSB
Analog reference voltage	AV_{REF}	$AV_{REF} = AV_{DD}$	2.7		3.6	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF}	V
AV_{REF} current	AI_{REF}			360	500	μA
AV_{DD} power supply current	AI_{DD}			1	3	mA

Notes 1. Excluding quantization error (±0.05% FSR).

2. Excluding quantization error (±0.5 LSB)

Remark FSR: Full Scale Range
LSB: Least Significant Bit

Flash Memory Programming Mode
(μPD70F3015B, 70F3015BY, 70F3017A, 70F3017AY only)

Write/erase characteristics (T_A = 0 to 85°C, V_{DD} = AV_{DD} = BV_{DD} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = BV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} supply voltage	V_{PP2}	During flash memory programming	7.5	7.8	8.1	V
V_{DD} supply current	I_{DD}	When V_{PP} = V_{PP2}, f_{XX} = 20 MHz			67	mA
V_{PP} supply current	I_{PP}	V_{PP} = V_{PP2}, 0.1 s after erasure			100	mA
Step erase time	t_{ER}	**Note 1**		0.2		s
Overall erase time per area[Note 2]	t_{ERA}	When step erase time = 0.2 s, **Note 3**			20	s/area
Write-back time	t_{WB}	**Note 4**		1		ms
Number of write-backs per write-back command	C_{WB}	When write-back time = 1 ms, **Note 5**			300	Count/write-back command
Number of erase/write-backs	C_{ERWB}				16	Count
Step writing time	t_{WR}	**Note 6**		20		μs
Overall writing time per word	t_{WRW}	When step writing time = 20 μs (1 word = 4 bytes), **Note 7**	20		200	μs/word
Number of rewrites per area[Note 2]	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Notes 8, 9**		100		Count/area
				20		Count/area

Notes 1. The recommended setting value of the step erase time is 0.2 s.

2. No areas are included in the μPD70F3015B and 70F3015BY.
The areas the μPD70F3017A and 70F3017AY are as follows.
Area 0 = 000000H to 01FFFFH
Area 1 = 020000H to 03FFFFH

3. The prewrite time prior to erasure and the erase verify time (write-back time) are not included.

4. The recommended setting value of the write-back time is 1 ms.

5. Write-back is executed once by the issuance of the write-back command. Therefore, the retry count must be the maximum value minus the number of commands issued.

6. The recommended setting value of the step writing time is 20 μs.

7. 20 μs is added to the actual writing time per word. The internal verify time during and after the writing is not included.

8. When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)
Shipped product ⟶ P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

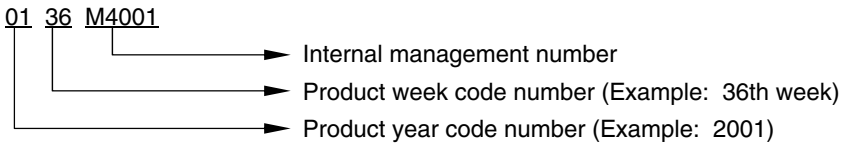
Notes 9. The versions that guarantee 20 rewrites can be distinguished from the versions that guarantee 100 rewrites according to the product or the lot number stamped on the package (xxxx indicates the four-digit number or symbol for internal management).

- μPD70F3015B,70F3015BY: Only products that guarantee 100 rewrites
  (Rewrite temperature: 0 to 85°C)

- μPD70F3017A, 70F3017AY

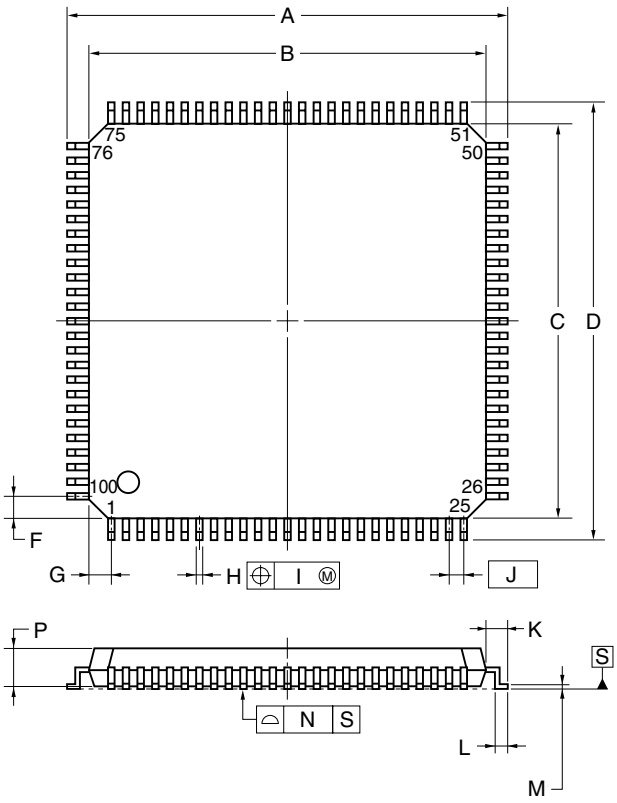
	20 Rewrites Guaranteed	100 Rewrites Guaranteed
Lot No.	0135Mxxxx or earlier	0136Mxxxx or later
Flash memory rewrite count	20 rewrites	100 rewrites
Flash memory rewrite temperature	10 to 40°C	0 to 85°C

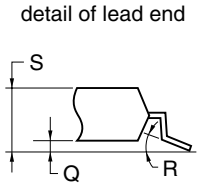
- About lot No.



Remark When the PG-FP3 and PG-FP4 are used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)





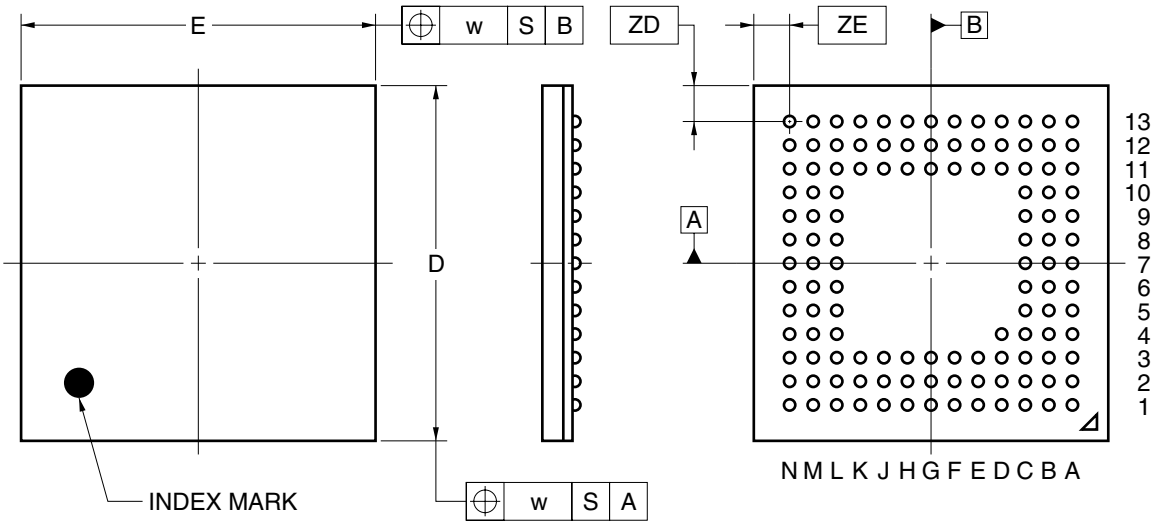
NOTE

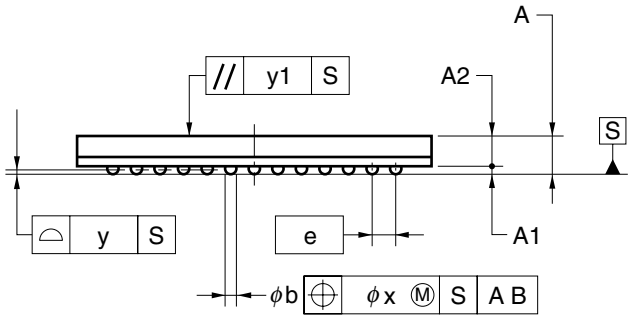
Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

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 19 RECOMMENDED SOLDERING CONDITIONS

The V850/SA1 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 19-1. Surface Mounting Type Soldering Conditions (1/4)

(1) PD703014BGC- -8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD703014BYGC- -8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD703015BGC- -8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD703015BYGC- -8EU: 100-pin plastic LQFP (fine pitch) (14 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less	VP15-00-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

(2) PD703017AGC- -8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD703017AYGC- -8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3015BGC-8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3015BYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-107-2
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).

Table 19-1. Surface Mounting Type Soldering Conditions (2/4)

(3) PD70F3017AGC-8EU: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3017AYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-103-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-103-2
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).

(4) PD703014AF1- -EA6: 121-pin plastic FBGA (12 12)
PD703014AYF1- -EA6: 121-pin plastic FBGA (12 12)
PD703015AF1- -EA6: 121-pin plastic FBGA (12 12)
PD703015AYF1- -EA6: 121-pin plastic FBGA (12 12)
PD703017AF1- -EA6: 121-pin plastic FBGA (12 12)
PD703017AYF1- -EA6: 121-pin plastic FBGA (12 12)
PD70F3017AF1-EA6: 121-pin plastic FBGA (12 12)
PD70F3017AYF1-EA6: 121-pin plastic FBGA (12 12)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Two times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Two times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 hours)	VP15-107-2

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).

Table 19-1. Surface Mounting Type Soldering Conditions (3/4)

(5) PD703015BGC-xxx-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD703015BYGC-xxx-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD703017AYGC-xxx-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3015BGC-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3017AGC-8EU-A: 100-pin plastic LQFP (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 20 to 72 hours)	IR60-203-3
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.

(6) PD703014BGC-xxx-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD703014BYGC-xxx-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD703017AGC-xxx-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3015BYGC-8EU-A: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3017AYGC-8EU-A: 100-pin plastic LQFP (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 19-1. Surface Mounting Type Soldering Conditions (4/4)

(7) PD703014AF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703014AYF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703014BF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703015AF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703015AYF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703015BF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703017AF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703017AYF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD70F3015BF1-EA6-A: 121-pin plastic FBGA (12 12)
PD70F3017AF1-EA6-A: 121-pin plastic FBGA (12 12)
PD70F3017AYF1-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.

(8) PD703014BYF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD703015BYF1- -EA6-A: 121-pin plastic FBGA (12 12)
PD70F3015BYF1-EA6-A: 121-pin plastic FBGA (12 12)

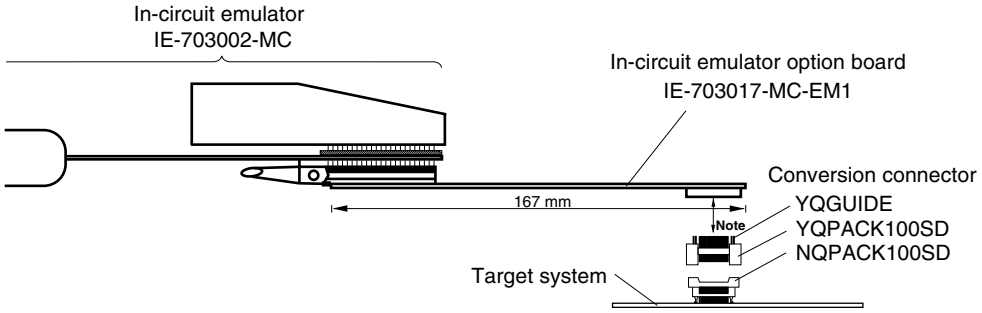
Undefined

★ APPENDIX A NOTES ON TARGET SYSTEM DESIGN

The following shows a diagram of the connection conditions between the in-circuit emulator option board and conversion connector. Design your system making allowances for conditions such as the shape of parts mounted on the target system as shown below.

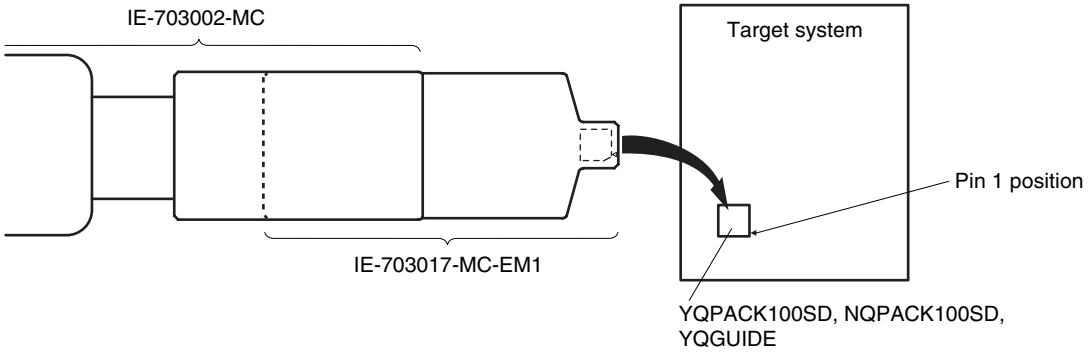
Figure A-1. 100-Pin Plastic LQFP (Fine Pitch) (14 14)

Side view



Note YQSOCKET100SDN (included with IE-703002-MC) can be inserted here to adjust the height (height: 3.2 mm).

Top view



Connection condition diagram

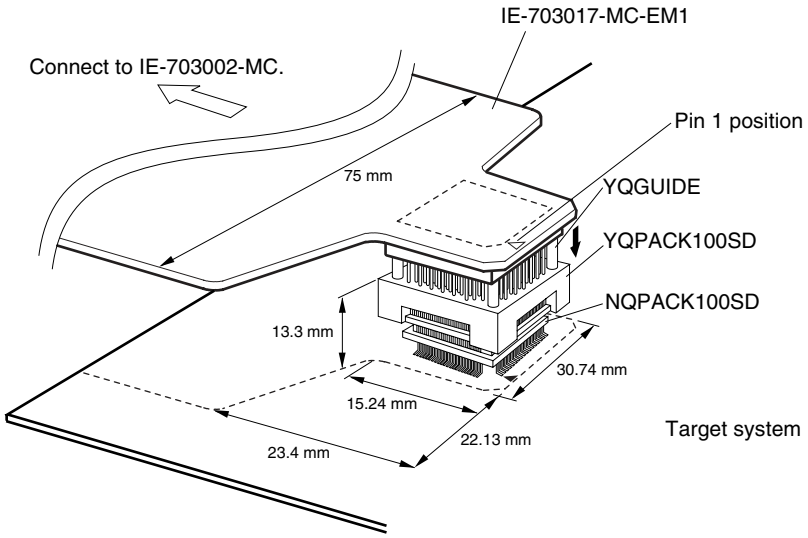
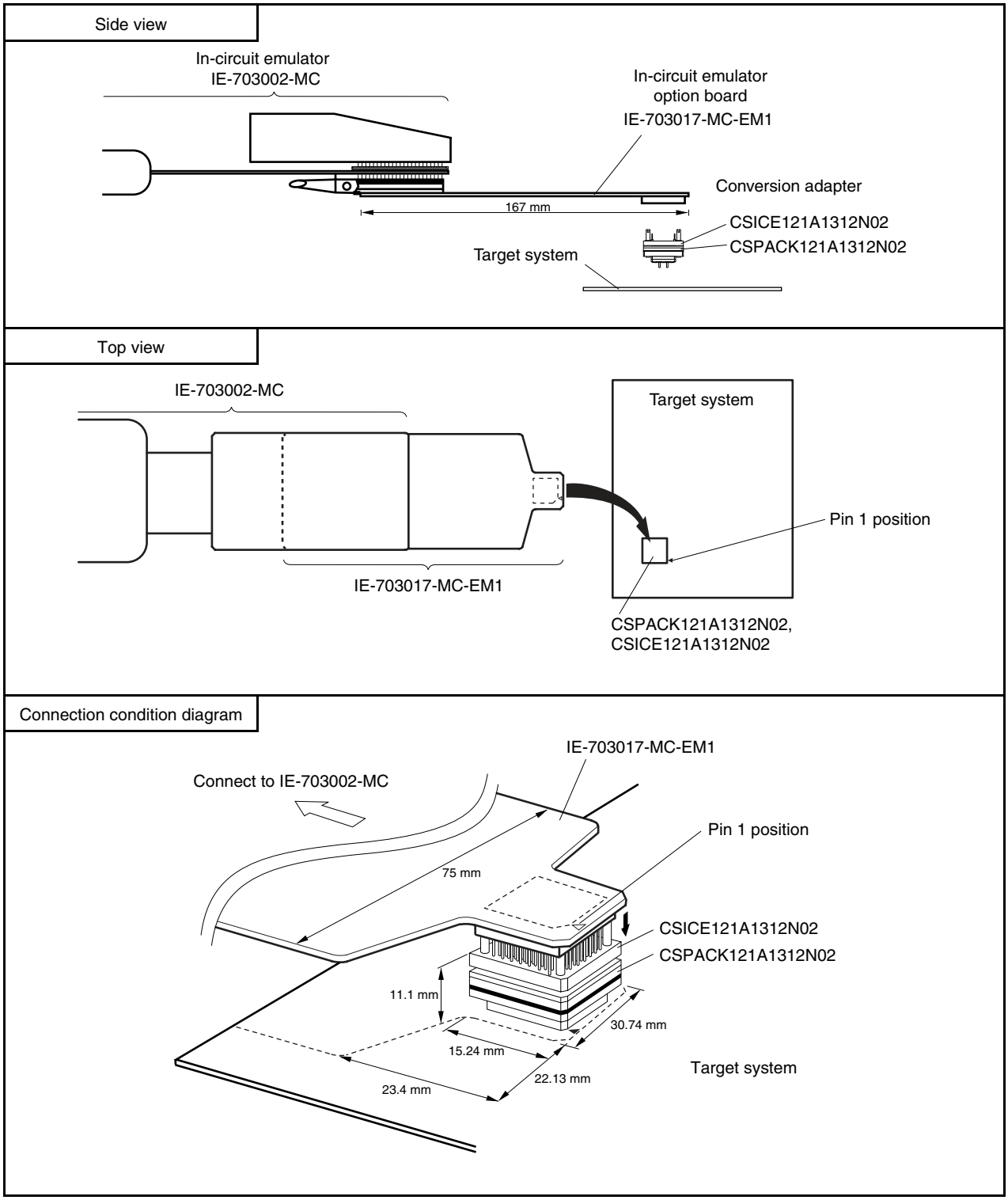


Figure A-2. 121-Pin Plastic FBGA (12 12)



APPENDIX B REGISTER INDEX

(1/5)

Symbol	Name	Unit	Page
ADCR	A/D conversion result register	ADC	310
ADCRH	A/D conversion result register H	ADC	310
ADIC	Interrupt control register	INTC	122 to 124
ADM	A/D converter mode register	ADC	312
ADS	Analog input channel specification register	ADC	315
ASIM0	Asynchronous serial interface mode register 0	UART	288
ASIM1	Asynchronous serial interface mode register 1	UART	288
ASIS0	Asynchronous serial interface status register 0	UART	290
ASIS1	Asynchronous serial interface status register 1	UART	290
BCC	Bus cycle control register	BCU	95
BRGC0	Baud rate generator control register 0	BRG	291
BRGC1	Baud rate generator control register 1	BRG	291
BRGMC0	Baud rate generator mode control register 0	BRG	292
BRGMC01	Baud rate generator mode control register 01	BRG	292
BRGMC1	Baud rate generator mode control register 1	BRG	293
CR00	Capture/compare register 00	RPU	156
CR01	Capture/compare register 01	RPU	157
CR10	Capture/compare register 10	RPU	156
CR11	Capture/compare register 11	RPU	157
CR20	8-bit compare register 2	RPU	191
CR23	16-bit compare register 23 (when TM2 and TM3 are connected in cascade)	RPU	205
CR30	8-bit compare register 3	RPU	191
CR40	8-bit compare register 4	RPU	191
CR45	16-bit compare register 45 (when TM4 and TM5 are connected in cascade)	RPU	205
CR50	8-bit compare register 5	RPU	191
CRC0	Capture/compare control register 0	RPU	160
CRC1	Capture/compare control register 1	RPU	160
CSIC0	Interrupt control register	INTC	122 to 124
CSIC1	Interrupt control register	INTC	122 to 124
CSIC2	Interrupt control register	INTC	122 to 124
CSIM0	Serial operation mode register 0	CSI	223
CSIM1	Serial operation mode register 1	CSI	223
CSIM2	Serial operation mode register 2	CSI	223

(2/5)

Symbol	Name	Unit	Page
CSIS0	Serial clock select register 0	CSI	223
CSIS1	Serial clock select register 1	CSI	223
CSIS2	Serial clock select register 2	CSI	223
DBC0	DMA byte counter register 0	DMAC	335
DBC1	DMA byte counter register 1	DMAC	335
DBC2	DMA byte counter register 2	DMAC	335
DCHC0	DMA channel control register 0	DMAC	336
DCHC1	DMA channel control register 1	DMAC	336
DCHC2	DMA channel control register 2	DMAC	336
DIOA0	DMA peripheral I/O address register 0	DMAC	332
DIOA1	DMA peripheral I/O address register 1	DMAC	332
DIOA2	DMA peripheral I/O address register 2	DMAC	332
DMAIC0	Interrupt control register	INTC	122 to 124
DMAIC1	Interrupt control register	INTC	122 to 124
DMAIC2	Interrupt control register	INTC	122 to 124
DRA0	DMA internal RAM address register 0	DMAC	332
DRA1	DMA internal RAM address register 1	DMAC	332
DRA2	DMA internal RAM address register 2	DMAC	332
DWC	Data wait control register	BCU	93
ECR	Interrupt source register	CPU	62
EGN0	Falling edge specification register	INTC	114, 351
EGP0	Rising edge specification register	INTC	114, 351
EIPC	Interrupt status saving register	CPU	62
EIPSW	Interrupt status saving register	CPU	62
FEPC	NMI status saving register	CPU	62
FEPSW	NMI status saving register	CPU	62
★ FLPMC	Flash memory programming mode control register	CPU	423
IIC0	IIC shift register 0	I^2C	242
IICC0	IIC control register 0	I^2C	234
IICCL0	IIC clock select register 0	I^2C	241
IICS0	IIC status register 0	I^2C	238
IICX0	IIC function expansion register 0	I^2C	241
ISPR	In-service priority register	INTC	125
MAM	Memory address output mode register	Port	78
MM	Memory expansion mode register	Port	77
OSTS	Oscillation stabilization time select register	WDT	143, 215, 220
P0	Port 0	Port	348
P1	Port 1	Port	353

(3/5)

Symbol	Name	Unit	Page
P10	Port 10	Port	383
P11	Port 11	Port	387
P12	Port 12	Port	390
P2	Port 2	Port	359
P3	Port 3	Port	367
P4	Port 4	Port	372
P5	Port 5	Port	372
P6	Port 6	Port	375
P7	Port 7	Port	377
P8	Port 8	Port	377
P9	Port 9	Port	379
PCC	Processor clock control register	CG	140
PF1	Port 1 function register	Port	355
PF10	Port 10 function register	Port	385
PF2	Port 2 function register	Port	361
PIC0	Interrupt control register	INTC	122 to 124
PIC1	Interrupt control register	INTC	122 to 124
PIC2	Interrupt control register	INTC	122 to 124
PIC3	Interrupt control register	INTC	122 to 124
PIC4	Interrupt control register	INTC	122 to 124
PIC5	Interrupt control register	INTC	122 to 124
PIC6	Interrupt control register	INTC	122 to 124
PM0	Port 0 mode register	Port	350
PM1	Port 1 mode register	Port	354
PM10	Port 10 mode register	Port	384
PM11	Port 11 mode register	Port	388
PM12	Port 12 mode register	Port	391
PM2	Port 2 mode register	Port	360
PM3	Port 3 mode register	Port	368
PM4	Port 4 mode register	Port	373
PM5	Port 5 mode register	Port	373
PM6	Port 6 mode register	Port	376
PM9	Port 9 mode register	Port	380
PMC12	Port 12 mode control register	Port	391
PRCMD	Command register	CG	88
PRM0	Prescaler mode register 0	RPU	162
PRM01	Prescaler mode register 01	RPU	162

(4/5)

Symbol	Name	Unit	Page
PRM1	Prescaler mode register 1	RPU	164
PRM11	Prescaler mode register 11	RPU	164
PSC	Power save control register	CG	142
PSW	Program status word	CPU	63
PU0	Pull-up resistor option register 0	Port	350
PU1	Pull-up resistor option register 1	Port	355
PU10	Pull-up resistor option register 10	Port	385
PU11	Pull-up resistor option register 11	Port	388
PU2	Pull-up resistor option register 2	Port	361
PU3	Pull-up resistor option register 3	Port	368
RTBH	Real-time output buffer register H	RPU	342
RTBL	Real-time output buffer register L	RPU	342
RTPC	Real-time output port control register	RPU	344
RTPM	Real-time output port mode register	RPU	343
RX0	Receive shift register 0	UART	287
RX1	Receive shift register 1	UART	287
RXB0	Receive buffer register 0	UART	287
RXB1	Receive buffer register 1	UART	287
SAR	Successive approximation register	ADC	310
SERIC0	Interrupt control register	INTC	122 to 124
SERIC1	Interrupt control register	INTC	122 to 124
SIO0	Serial I/O shift register 0	CSI	222
SIO1	Serial I/O shift register 1	CSI	222
SIO2	Serial I/O shift register 2	CSI	222
SRIC1	Interrupt control register	INTC	122 to 124
STIC0	Interrupt control register	INTC	122 to 124
STIC1	Interrupt control register	INTC	122 to 124
SVA0	Slave address register 0	I^2C	242
SYC	System control register	BCU	90
SYS	System status register	CG	88
TCL2	Timer clock select register 2	RPU	193
TCL21	Timer clock select register 21	RPU	193
TCL3	Timer clock select register 3	RPU	193
TCL31	Timer clock select register 31	RPU	193
TCL4	Timer clock select register 4	RPU	193
TCL41	Timer clock select register 41	RPU	193
TCL5	Timer clock select register 5	RPU	193

(5/5)

Symbol	Name	Unit	Page
TCL51	Timer clock select register 51	RPU	193
TM0	16-bit timer register 0	RPU	155
TM1	16-bit timer register 1	RPU	155
TM2	8-bit counter 2	RPU	191
TM23	16-bit counter 23 (when TM2 and TM3 are connected in cascade)	RPU	205
TM3	8-bit counter 3	RPU	191
TM4	8-bit counter 4	RPU	191
TM45	16-bit counter 45 (when TM4 and TM5 are connected in cascade)	RPU	205
TM5	8-bit counter 5	RPU	191
TMC0	16-bit timer mode control register 0	RPU	158
TMC1	16-bit timer mode control register 1	RPU	158
TMC2	8-bit timer mode control register 2	RPU	195
TMC3	8-bit timer mode control register 3	RPU	195
TMC4	8-bit timer mode control register 4	RPU	195
TMC5	8-bit timer mode control register 5	RPU	195
TMIC00	Interrupt control register	INTC	122 to 124
TMIC01	Interrupt control register	INTC	122 to 124
TMIC10	Interrupt control register	INTC	122 to 124
TMIC11	Interrupt control register	INTC	122 to 124
TMIC2	Interrupt control register	INTC	122 to 124
TMIC3	Interrupt control register	INTC	122 to 124
TMIC4	Interrupt control register	INTC	122 to 124
TMIC5	Interrupt control register	INTC	122 to 124
TOC0	16-bit timer output control register 0	RPU	161
TOC1	16-bit timer output control register 1	RPU	161
TXS0	Transmit shift register 0	UART	287
TXS1	Transmit shift register 1	UART	287
WDCS	Watchdog timer clock select register	WDT	216
WDTIC	Interrupt control register	INTC	122 to 124
WDTM	Watchdog timer mode register	WDT	126, 217
WTIC	Interrupt control register	INTC	122 to 124
WTIIC	Interrupt control register	INTC	122 to 124
WTM	Watch timer mode control register	WT	210

APPENDIX C LIST OF INSTRUCTION SETS

- **How to read instruction set list**

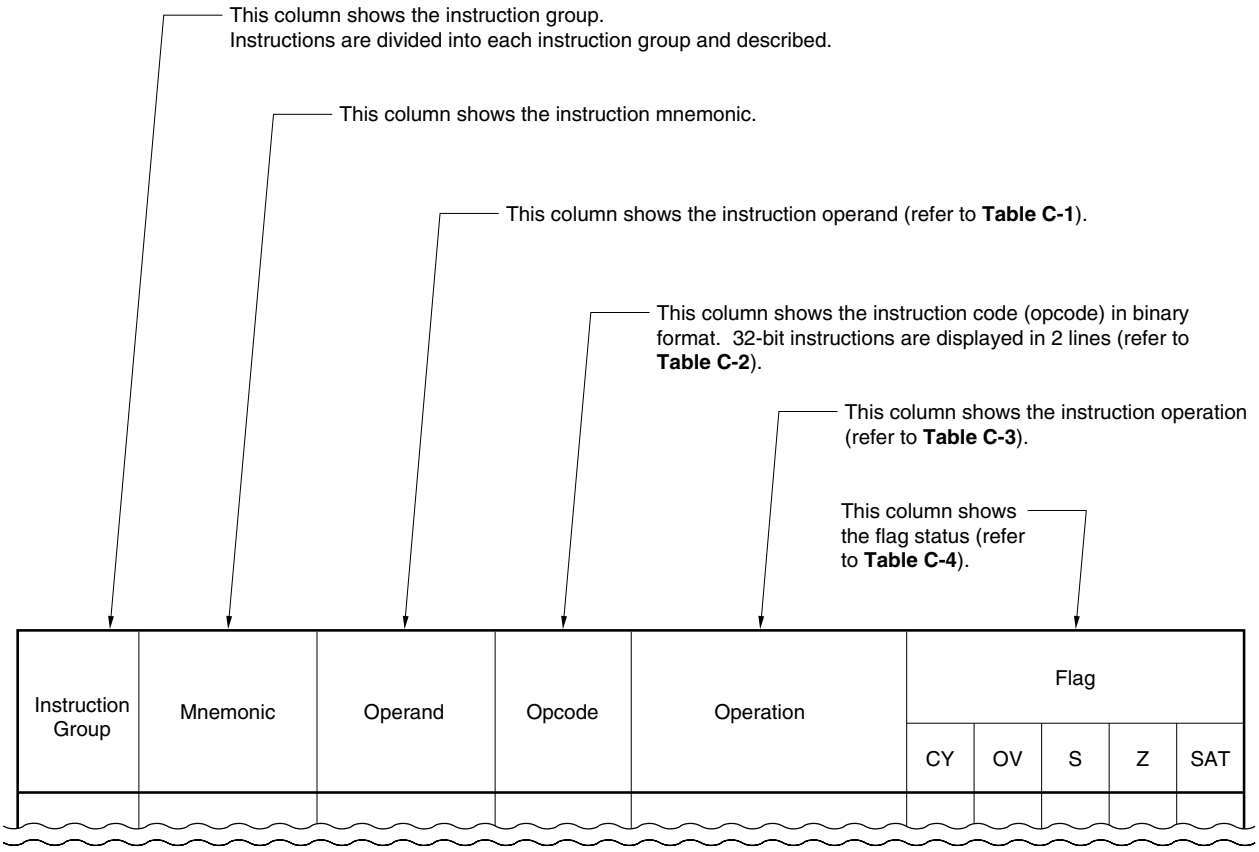


Table C-1. Symbols in Operand Description

Symbol	Description
reg1	General-purpose register (r0 to r31): Used as source register
reg2	General-purpose register (r0 to r31): Mainly used as destination register
ep	Element pointer (r30)
bit#3	3-bit data for bit number specification
imm×	×-bit immediate data
disp×	×-bit displacement
regID	System register number
vector	5-bit data that specifies trap vector number (00H to 1FH)
cccc	4-bit data that indicates condition code

Table C-2. Symbols Used for Opcode

Symbol	Description
R	1-bit data of code that specifies reg1 or regID
r	1-bit data of code that specifies reg2
d	1-bit data of displacement
i	1-bit data of immediate data
cccc	4-bit data that indicates condition code
bbb	3-bit data that specifies bit number

Table C-3. Symbols Used for Operation Description

Symbol	Description
←	Assignment
GR[ ]	General-purpose register
SR[ ]	System register
zero-extend (n)	Zero-extends n to word length.
sign-extend (n)	Sign-extends n to word length.
load-memory (a,b)	Reads data of size b from address a.
store-memory (a,b,c)	Writes data b of size c to address a.
load-memory-bit (a,b)	Reads bit b from address a.
store-memory-bit (a,b,c)	Writes c to bit b of address a.
saturated (n)	Performs saturated processing of n. (n is 2's complement). Result of calculation of n: If n is n ≥ 7FFFFFFFH as result of calculation, 7FFFFFFFH. If n is n ≤ 80000000H as result of calculation, 80000000H.
result	Reflects result to a flag.
Byte	Byte (8 bits)
Halfword	Halfword (16 bits)
Word	Word (32 bits)
+	Add
−	Subtract
||	Bit concatenation
×	Multiply
÷	Divide
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum
NOT	Logical negate
logically shift left by	Logical left shift
logically shift right by	Logical right shift
arithmetically shift right by	Arithmetic right shift

Table C-4. Symbols Used for Flag Operation

Symbol	Description
(blank)	Not affected
0	Cleared to 0
×	Set or cleared according to result
R	Previously saved value is restored

Table C-5. Condition Codes

Condition Name (cond)	Condition Code (cccc)	Conditional Expression	Description
V	0000	OV = 1	Overflow
NV	1000	OV = 0	No overflow
C/L	0001	CY = 1	Carry Lower (Less than)
NC/NL	1001	CY = 0	No carry No lower (Greater than or equal)
Z/E	0010	Z = 1	Zero Equal
NZ/NE	1010	Z = 0	Not zero Not equal
NH	0011	(CY OR Z) = 1	Not higher (Less than or equal)
H	1011	(CY OR Z) = 0	Higher (Greater than)
N	0100	S = 1	Negative
P	1100	S = 0	Positive
T	0101	–	Always (unconditional)
SA	1101	SAT = 1	Saturated
LT	0110	(S XOR OV) = 1	Less than signed
GE	1110	(S XOR OV) = 0	Greater than or equal signed
LE	0111	((S XOR OV) OR Z) = 1	Less than or equal signed
GT	1111	((S XOR OV) OR Z) = 0	Greater than signed

Instruction Set List (1/4)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Load/store	SLD.B	disp7 [ep], reg2	rrrrr0110ddddddd	adr ← ep + zero-extend (disp7) GR [reg2] ← sign-extend (Load-memory (adr, Byte))					
	SLD.H	disp8 [ep], reg2	rrrrr1000ddddddd **Note 1**	adr ← ep + zero-extend (disp8) GR [reg2] ← sign-extend (Load-memory (adr, Halfword))					
	SLD.W	disp8 [ep], reg2	rrrrr1010ddddddd0 **Note 2**	adr ← ep + zero-extend (disp8) GR [reg2] ← Load-memory (adr, Word)					
	LD.B	disp16 [reg1], reg2	rrrrr111000RRRRR ddddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← sign-extend (Load-memory (adr, Byte))					
	LD.H	disp16 [reg1], reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← sign-extend (Load-memory (adr, Halfword))					
	LD.W	disp16 [reg1], reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← Load-memory (adr, Word))					
	SST.B	reg2, disp7 [ep]	rrrrr0111ddddddd	adr ← ep + zero-extend (disp7) Store-memory (adr, GR [reg2], Byte)					
	SST.H	reg2, disp8 [ep]	rrrrr1001ddddddd **Note 1**	adr ← ep + zero-extend (disp8) Store-memory (adr, GR [reg2], Halfword)					
	SST.W	reg2, disp8 [ep]	rrrrr1010ddddddd1 **Note 2**	adr ← ep + zero-extend (disp8) Store-memory (adr, GR [reg2], Word)					
	ST.B	reg2, disp16 [reg1]	rrrrr111010RRRRR ddddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) Store-memory (adr, GR [reg2], Byte)					
	ST.H	reg2, disp16 [reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) Store-memory (adr, GR [reg2], Halfword)					
	ST.W	reg2, disp16 [reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) Store-memory (adr, GR [reg2], Word)					
Arithmetic operation	MOV	reg1, reg2	rrrrr000000RRRRR	GR [reg2] ← GR [reg1]					
	MOV	imm5, reg2	rrrrr010000iiiii	GR [reg2] ← sign-extend (imm5)					
	MOVHI	imm16, reg1, reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1] + (imm16 || 0^{16})					
	MOVEA	imm16, reg1, reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1] + sign-extend (imm16)					

Notes 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

Instruction Set List (2/4)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Arithmetic operation	ADD	reg1, reg2	rrrrr001110RRRRR	GR [reg2] ← GR [reg2] + GR [reg1]	×	×	×	×	
	ADD	imm5, reg2	rrrrr010010iiiii	GR [reg2] ← GR [reg2] + sign-extend (imm5)	×	×	×	×	
	ADDI	imm16, reg1, reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1] + sign-extend (imm16)	×	×	×	×	
	SUB	reg1, reg2	rrrrr001101RRRRR	GR [reg2] ← GR [reg2] – GR [reg1]	×	×	×	×	
	SUBR	reg1, reg2	rrrrr001100RRRRR	GR [reg2] ← GR [reg1] – GR [reg2]	×	×	×	×	
	MULH	reg1, reg2	rrrrr000111RRRRR	GR [reg2] ← GR [reg2]Note × GR [reg1]Note (Signed multiplication)					
	MULH	imm5, reg2	rrrrr010111iiiii	GR [reg2] ← GR [reg2]Note × sign-extend (imm5) (Signed multiplication)					
	MULHI	imm16, reg1, reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1]Note × imm16 (Signed multiplication)					
	DIVH	reg1, reg2	rrrrr000010RRRRR	GR [reg2] ← GR [reg2] ÷ GR [reg1]Note (Signed division)		×	×	×	
	CMP	reg1, reg2	rrrrr001111RRRRR	result ← GR [reg2] – GR [reg1]	×	×	×	×	
	CMP	imm5, reg2	rrrrr010011iiiii	result ← GR [reg2] – sign-extend (imm5)	×	×	×	×	
	SETF	cccc, reg2	rrrrr1111110cccc 0000000000000000	if conditions are satisfied then GR [reg2] ← 00000001H else GR [reg2] ← 00000000H					
Saturated operation	SATADD	reg1, reg2	rrrrr000110RRRRR	GR [reg2] ← saturated (GR [reg2] + GR [reg1])	×	×	×	×	×
	SATADD	imm5, reg2	rrrrr010001iiiii	GR [reg2] ← saturated (GR [reg2] + sign-extend (imm5))	×	×	×	×	×
	SATSUB	reg1, reg2	rrrrr000101RRRRR	GR [reg2] ← saturated (GR [reg2] – GR [reg1])	×	×	×	×	×
	SATSUBI	imm16, reg1, reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← saturated (GR [reg1] – sign-extend (imm16))	×	×	×	×	×
	SATSUBR	reg1, reg2	rrrrr000100RRRRR	GR [reg2] ← saturated (GR [reg1] – GR [reg2])	×	×	×	×	×
Logic operation	TST	reg1, reg2	rrrrr001011RRRRR	result ← GR [reg2] AND GR [reg1]		0	×	×	
	OR	reg1, reg2	rrrrr001000RRRRR	GR [reg2] ← GR [reg2] OR GR [reg1]		0	×	×	
	ORI	imm16, reg1, reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1] OR zero-extend (imm16)		0	×	×	
	AND	reg1, reg2	rrrrr001010RRRRR	GR [reg2] ← GR [reg2] AND GR [reg1]		0	×	×	
	ANDI	imm16, reg1, reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1] AND zero-extend (imm16)		0	0	×	

Note Only the lower halfword data is valid.

Instruction Set List (3/4)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Logic operation	XOR	reg1, reg2	rrrrr001001RRRRR	GR [reg2] ← GR [reg2] XOR GR [reg1]		0	×	×	
	XORI	imm16, reg1, reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR [reg2] ← GR [reg1] XOR zero-extend (imm16)		0	×	×	
	NOT	reg1, reg2	rrrrr000001RRRRR	GR [reg2] ← NOT (GR [reg1])		0	×	×	
	SHL	reg1, reg2	rrrrr111111RRRRR 0000000011000000	GR [reg2] ← GR [reg2] logically shift left by GR [reg1]	×	0	×	×	
	SHL	imm5, reg2	rrrrr010110iiiii	GR [reg2] ← GR [reg2] logically shift left by zero-extend (imm5)	×	0	×	×	
	SHR	reg1, reg2	rrrrr111111RRRRR 0000000010000000	GR [reg2] ← GR [reg2] logically shift right by GR [reg1]	×	0	×	×	
	SHR	imm5, reg2	rrrrr010100iiiii	GR [reg2] ← GR [reg2] logically shift right by zero-extend (imm5)	×	0	×	×	
	SAR	reg1, reg2	rrrrr111111RRRRR 0000000010100000	GR [reg2] ← GR [reg2] arithmetically shift right by GR [reg1]	×	0	×	×	
	SAR	imm5, reg2	rrrrr010101iiiii	GR [reg2] ← GR [reg2] arithmetically shift right by zero-extend (imm5)	×	0	×	×	
Jump	JMP	[reg1]	00000000011RRRRR	PC ← GR [reg1]					
	JR	disp22	0000011110ddddddd dddddddddddddddd0 **Note 1**	PC ← PC + sign-extend (disp22)					
	JARL	disp22, reg2	rrrrr11110ddddddd dddddddddddddddd0 **Note 1**	GR [reg2] ← PC + 4 PC ← PC + sign-extend (disp22)					
	Bcond	disp9	ddddd1011dddcccc **Note 2**	if conditions are satisfied then PC ← PC + sign-extend (disp9)					
Bit manipulate	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)				×	
	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)				×	
	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)				×	
	TST1	bit#3, disp16 [reg1]	11bbb111110RRRRR dddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3))				×	

Notes 1. ddddddddddddddddddddddd is the higher 21 bits of dip22.

2. dddddddd is the higher 8 bits of disp9.

Instruction Set List (4/4)

Instruction Group	Mnemonic	Operand	Opcode	Operation		Flag				
						CY	OV	S	Z	SAT
Special	LDSR	reg2, regID	rrrrr111111RRRRR 0000000000100000 **Note**	SR [regID] ←GR [reg2]	regID = EIPC, FEPC					
					regID = EIPSW, FEPSW					
					regID = PSW	×	×	×	×	×
	STSR	regID, reg2	rrrrr111111RRRRR 0000000001000000	GR [reg2] ← SR [regID]						
	TRAP	vector	00000111111iiiii 0000000100000000	EIPC ← PC + 4 (Restored PC) EIPSW ← PSW ECR.EICC ← Interrupt code PSW.EP ← 1 PSW.ID ← 1 PC ← 00000040H (vector = 00H to 0FH) 00000050H (vector = 10H to 1FH)						
	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		R	R	R	R	R
	HALT		0000011111100000 0000000100100000	Stops						
	DI		0000011111100000 0000000101100000	PSW.ID ← 1 (Maskable interrupt disabled)						
	EI		1000011111100000 0000000101100000	PSW.ID ← 0 (Maskable interrupt enabled)						
	NOP		0000000000000000	Uses 1 clock cycle without doing anything						

Note The opcode of this instruction uses the field of reg1 even though the source register is shown as reg2 in the above table. Therefore, the meaning of the register specification for mnemonic description and opcode is different from that of the other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

APPENDIX D INDEX

[Q]

[R]

[S]

[T]

[U]

[V]

[W]

[X]

[Z]

★ APPENDIX E REVISION HISTORY

The following table shows the revision history up to this edition. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/7)

Edition	Major Revision from Previous Edition	Applied to:
3rd edition	Deletion of μPD703015, 703015Y, 70F3017, and 70F3017Y Addition of μPD703014A, 703014AY, 703015A, 703015AY, 703017A, 703017AY, 70F3017A, and 70F3017AY	Throughout
	1.2 Features Deletion and addition of products, modification of minimum instruction execution time	**INTRODUCTION**
	1.4 Ordering Information Deletion and addition of products	
	1.5 Pin Configuration Deletion and addition of products	
	1.6.1 Internal block diagram Deletion and addition of products	
	1.6.2 Internal units (3) ROM, **(4) RAM**, **(10) Serial interface (SIO)** Deletion and addition of products	
	2.1 (2) Non-port pins Deletion and addition of products in Note	**CHAPTER 2 PIN FUNCTIONS**
	2.3 (2) (b) (iv) SDA, **(v) SCL** Deletion and addition of products, **(23) V_{PP}**, **(24) IC** Deletion and addition of products	
	2.4 Pin I/O Circuits and Recommended Connection of Unused Pins Deletion of items AV_{DD} and AV_{SS} pins, modification of recommended connection method of AV_{REF} pin	
	3.1 Features Modification of minimum instruction execution time	**CHAPTER 3 CPU FUNCTIONS**
	3.2 CPU Register Set Modification of use of r2	
	3.2.1 (1) General registers Modification of use and operation of r2 and addition of Note 2	
	3.3 (2) Flash memory programming mode Deletion and addition of products	
	Figure 3-9 Memory Map Correction	
	3.4.5 (1) Internal ROM/internal flash memory area Addition	
	Table 3-3 Interrupt/Exception Table Deletion and addition of products in Note	
	3.4.5 (2) Internal RAM area Addition	
	Figure 3-16 External Memory Area (When Expanded to 64 KB, 256 KB, or 1 MB) Modification	
	Figure 3-17 External Memory Area (When Expanded to 4 MB) Modification	
	Figure 3-18 Memory Expansion Mode Register (MM) Addition of Caution	
	3.4.6 (2) Memory address output mode register (MAM) Addition of description	
	Figure 3-20 Application Example of Wrap-Around Modification	
	3.4.8 Peripheral I/O registers Addition of PRM01, PRM11, TCL21, TCL31, TCL41, TCL51, BRGMC01, IICX0, and Note	
	Table 5-1 Interrupt Source List Deletion and addition of products in Note 2	**CHAPTER 5 INTERRUPT/EXCEPTION PROCESSING FUNCTION**

(2/7)

Edition	Major Revision from Previous Edition	Applied to:
3rd edition	**6.1 (1) Main system clock oscillator** Modification of maximum operating frequency and addition of Caution 2	**CHAPTER 6 CLOCK GENERATION FUNCTION**
	Figure 6-2 Processor Clock Control Register (PCC) Modification of bit 7, addition of bit 5 (MFRC) and Caution 3	
	6.3.1 (1) (a) Example of main clock operation → sub clock operation setup, **(b) Example of sub clock operation → main clock operation setup** Addition	
	6.4.1 (3) Software STOP mode Deletion of part of description	
	Table 6-1 Operating Statuses in HALT Mode Correction of items UART0 and UART1, deletion and addition of products in Note	
	Table 6-2 Operating Statuses in IDLE Mode Correction of items UART0 and UART1, deletion and addition of products in Note	
	6.4.4 (1) Settings and operating states Deletion of part of description	
	Table 6-3 Operating Statuses in Software STOP Mode Correction of items UART0 and UART1, deletion and addition of products in Note	
	6.4.4 (2) Cancellation of software STOP mode Deletion of part of description	
	Figure 7-1 Block Diagram of TM0 and TM1 Modification	**CHAPTER 7 TIMER/COUNTER FUNCTION**
	Table 7-1 Configuration of Timers 0 and 1 Addition of prescaler mode registers 01, 11 (PRM01, PRM11)	
	Table 7-4 Valid Edge of TIn0 Pin and Capture Trigger of CRn1 Addition	
	7.1.4 Timer 0, 1 control registers Addition of prescaler mode register n1 (PRMn1)	
	Figure 7-3 Capture/Compare Control Registers 0, 1 (CRC0, CRC1) Addition of Caution 4	
	Figure 7-4 16-Bit Timer Output Control Registers 0, 1 (TOC0, TOC1) Correction of Note and Addition of Caution 4	
	7.1.4 (4) Prescaler mode registers 0, 01 (PRM0, PRM01) Addition of PRM01 register	
	Figure 7-5 Prescaler Mode Register 0 (PRM0) Addition of Note 2	
	Figure 7-6 Prescaler Mode Register 01 (PRM01) Addition	
	7.1.4 (5) Prescaler mode registers 1, 11 (PRM1, PRM11) Addition of PRM11 register	
	Figure 7-7 Prescaler Mode Register 1 (PRM1) Addition of Note 2	
	Figure 7-8 Prescaler Mode Register 11 (PRM11) Addition	
	Figure 7-10 Configuration of Interval Timer Correction	
	7.2.3 (1) Pulse width measurement with free running counter and one capture register Addition of PRM01 and PRM11 registers	
	Figure 7-14 Configuration for Pulse Width Measurement with Free Running Counter Addition of PRM01 and PRM11 registers in Note	
	7.2.3 (2) Measurement of two pulse widths with free running counter Addition of PRM01 and PRM11 registers	
	7.2.3 (3) Pulse width measurement with free running counter and two capture registers Addition of PRM01 and PRM11 registers	
	7.2.3 (4) Pulse width measurement by restarting Addition of PRM01 and PRM11 registers	
	7.2.4 Operation as external event counter Addition of PRM01 and PRM11 registers	

(3/7)

Edition	Major Revision from Previous Edition	Applied to:
3rd edition	**Figure 7-24 Configuration of External Event Counter** Addition of PRM01 and PRM11 registers in Note	**CHAPTER 7 TIMER/COUNTER FUNCTION**
	Figure 7-28 Control Register Settings for One-Shot Pulse Output with Software Trigger Addition of Caution	
	7.2.7 (2) Setting 16-bit capture/compare register, **(7) <2>**, **(8) Conflicting operations**, **(9) Timer operation**, **(10) Capture operation**, **(11) Compare operation**, **(12) Edge detection** Addition	
	7.3.1 Functions Addition of Caution	
	Figure 7-36 Block Diagram of TM2 to TM5 Correction	
	Table 7-5 Configuration of Timers 2 to 5 Addition of timer clock select register n1 (TCLn1)	
	7.3.2 (1) 8-bit counters 2 to 5 (TM2 to TM5) Addition of description	
	7.3.2 (2) 8-bit compare registers 2 to 5 (CR20 to CR50) Addition of description	
	7.3.3 Timer n control register Addition of TCLn1 register	
	7.3.3 (1) Timer clock select registers 2, 21, 3, 31, 4, 41, 5, 51 (TCL2 to TCL5 and TCL21 to TCL51) Addition of TCLn1 register	
	Figure 7-37 TM2, TM3 Timer Clock Select Registers 2, 21, 3, 31 (TCL2, TCL21, TCL3, and TCL31) Addition of TCLn1 register	
	Figure 7-38 TM4, TM5 Timer Clock Select Registers 4, 41, 5, 51 (TCL4, TCL41, TCL5, and TCL51) Addition of TCLn1 register	
	Figure 7-39 8-Bit Timer Mode Control Registers 2 to 5 (TMC2 to TMC5) Addition of Note	
	Figure 7-42 Timing of Square Wave Output Operation Addition	
	7.4.5 (1) Cascade connection (16-bit timer) mode Addition of description	
	8.4.3 Cautions Addition	**CHAPTER 8 WATCH TIMER**
	Figure 9-1 Block Diagram of Watchdog Timer Correction	**CHAPTER 9 WATCHDOG TIMER**
	Table 9-1 Runaway Detection Time of Watchdog Timer Addition of runaway detection time when f_{XX} = 10 MHz, f_{XX} = 2 MHz	
	Table 9-2 Interval Time of Interval Timer Addition of interval time when f_{XX} = 10 MHz, f_{XX} = 2 MHz	
	Figure 9-2 Oscillation Stabilization Time Select Register (OSTS) Addition of oscillation stabilization time when f_{XX} = 10 MHz, f_{XX} = 2 MHz	
	Figure 9-3 Watchdog Timer Clock Select Register (WDCS) Addition of overflow time of watchdog timer/interval timer when f_{XX} = 10 MHz, f_{XX} = 2 MHz	
	Figure 9-4 Watchdog Timer Mode Register (WDTM) Correction of Caution	
	9.4.1 Operating as watchdog timer Deletion of part of description and correction of Caution 1	
	Table 9-4 Runaway Detection Time of Watchdog Timer Addition of runaway detection time when f_{XX} = 10 MHz, f_{XX} = 2 MHz	
	9.4.2 Operating as interval timer Deletion of part of description and correction of Caution 1	
	Table 9-5 Interval Time of Interval Timer Addition of interval time when f_{XX} = 10 MHz, f_{XX} = 2 MHz	

(4/7)

Edition	Major Revision from Previous Edition	Applied to:
3rd edition	**Figure 9-5 Oscillation Stabilization Time Select Register (OSTS)** Addition of oscillation stabilization time when f_{XX} = 10 MHz, f_{XX} = 2 MHz	**CHAPTER 9 WATCHDOG TIMER**
	10.1 Overview Deletion and addition of products in Note	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Figure 10-2 Serial Operation Mode Registers 0 to 2 (CSIM0 to CSIM2) Addition of Notes	
	Figure 10-5 Settings of CSIMn (3-Wire Serial I/O Mode) Addition of Note	
	10.3 I^2C Bus (μPD703014AY, 703015AY, 703017AY, 70F3017AY) Deletion and addition of products	
	Figure 10-7 Block Diagram of I^2C Correction	
	Table 10-2 Configuration of I^2C Addition of IIC function expansion register 0 (IICX0)	
	10.3.2 I^2C control registers Addition of IIC function expansion register 0 (IICX0)	
	Figure 10-9 IIC Control Register 0 (IICC0) (3/4) Addition of condition of STT = 0 and deletion of Note	
	Figure 10-9 IIC Control Register 0 (IICC0) (4/4) Addition of description and deletion of Note 2	
	Figure 10-10 IIC Status Register 0 (IICS0) (2/3) Addition of Note	
	10.3.2 (3) IIC clock select register 0 (IICCL0) Addition of Remark	
	Figure 10-11 IIC Clock Select Register 0 (IICCL0) Addition of operation description of bit 3 (SMC) and bit 2 (DFC)	
	10.3.2 (4) IIC function expansion register 0 (IICX0) Addition	
	Table 10-3 Settings of Transfer Clock Addition of IIC communication frequency when f_{XX} = 20 MHz	
	10.3.4 (4) Acknowledge signal ($\overline{\text{ACK}}$) Addition of description	
	10.3.5 I^2C interrupt request (INTIIC0) Addition of description	
	10.3.6 (4) Wait cancellation method Addition of cancellation method	
	Table 10-7 Wait Periods Modification of wait period when SMC, CL1, CL0 = 010 and 110	
	Figure 10-27 Master Operation Flow Chart Correction	
	Table 10-8 Configuration of UARTn Addition of baud rate generator mode control register 01 (BRGMC01)	
	10.4.2 UARTn control registers Addition of BRGMC01 register	
	10.4.2 (4) Baud rate generator mode control registers 0, 01 (BRGMC0, BRGMC01) Addition of BRGMC01 register	
	Figure 10-35 Baud Rate Generator Mode Control Registers 0, 01 (BRGMC0, BRGMC01) Addition of BRGMC01 register	
	10.4.4 (1) Register settings Addition of BRGMC01 register	
	10.4.4 (2) Generation of baud rate transmit/receive clock using main clock Correction	
	Table 10-9 Relationship Between Main Clock and Baud Rate Correction	
	Figure 11-2 A/D Converter Mode Register (ADM) Correction of conversion time	**CHAPTER 11 A/D CONVERTER**
	Figure 12-2 DMA Internal RAM Address Registers 0 to 2 (DRA0 to DRA2) Addition of Caution	**CHAPTER 12 DMA FUNCTIONS**

(5/7)

Edition	Major Revision from Previous Edition	Applied to:
3rd edition	**Figure 12-3 Correspondence Between DRAn Setup Value and Internal RAM Area** Addition	**CHAPTER 12 DMA FUNCTIONS**
	Figure 12-5 DMA Channel Control Registers 0 to 2 (DCHC0 to DCHC2) Deletion and addition of products in Note 2	
	14.2.1 (4) Block diagram (port 0) Addition	**CHAPTER 14 PORT FUNCTION**
	14.2.2 (3) Block diagrams (port 1) Addition	
	14.2.3 (3) Block diagrams (port 2) Addition	
	14.2.4 (3) Block diagrams (port 3) Addition	
	14.2.5 (1) Functions of P4 and P5 pins Modification of description	
	14.2.5 (3) Block diagram (port 4, port 5) Addition	
	14.2.6 (3) Block diagram (port 6) Addition	
	14.2.7 (2) Block diagram (port 7, port 8) Addition	
	14.2.8 (1) Function of P9 pins Modification of description	
	14.2.8 (3) Block diagrams (port 9) Addition	
	14.2.9 (3) Block diagram (port 10) Addition	
	14.2.10 (3) Block diagrams (port 11) Addition	
	14.2.11 (1) Function of P12 pin Deletion of description	
	14.2.11 (3) Block diagram (port 12) Addition	
	14.3 Setting When Port Pin Is Used for Alternate Function Addition	
	16.1.1 Erase units Addition	**CHAPTER 16 FLASH MEMORY (μPD70F3017A, 70F3017AY)**
	16.4 (3) CSI0 + HS Addition	
	Table 16-1 Signal Generation of Dedicated Flash Programmer (PG-FP3) Addition of CSI0 + HS	
	Table 16-2 Pins Used by Each Serial Interface Addition of CSI0 + HS	
	Table 16-3 List of Communication Systems Addition of CSI0 + HS	
4th edition	Addition of μPD703014B, 703014BY, 703015B, 703015BY, 70F3015B, and 70F3015BY Deletion of μPD703014AGC, 703014AYGC, 703015AGC, and 703015AYGC	Throughout
	Addition of **Table 1-1 List of V850/SA1 Products**	**INTRODUCTION**
	Addition of description to the minimum instruction execution time in **1.2 Features**	
	Deletion and addition of products in **1.4 Ordering Information**	
	Deletion and addition of products in **1.5 Pin Configuration**	
	Deletion of description in **1.6.2 (2) Bus control unit (BCU)**	
	Addition of **Table 2-1 Pin I/O Buffer Power Supplies**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of description in **Table 2-2 Operating States of Pins in Each Operating Mode**	
	Modification of description in **2.3 (7) P60 to P65 (Port 6)**	
	Addition of **2.3 (13) CLKOUT (Clock Out)**	
	Addition and modification of description in **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**	
	Modification of **2.5 Pin I/O Circuits**	

(6/7)

Edition	Major Revision from Previous Edition	Applied to:
4th edition	Addition of description to minimum instruction execution time in **3.1 Features**	**CHAPTER 3 CPU FUNCTIONS**
	Change of description in **3.2.2 (2) Program status word (PSW)**	
	Modification of **Figure 3-16 Recommended Memory Map**	
	Addition of description in **3.4.8 Peripheral I/O registers**	
	Addition and modification of description in **3.4.9 Specific registers**	
	Addition of description in **5.2.4 Noise elimination of external interrupt request input pin**	**CHAPTER 5 INTERRUPT/EXCEPTION PROCESSING FUNCTION**
	Addition of description in **5.2.5 Edge detection function of external interrupt request input pin**	
	Addition to **Cautions** in **5.3.4 Interrupt control register (xxICn)**	
	Addition of **Caution** in **5.3.5 In-service priority register (ISPR)**	
	Addition of **5.8.1 Interrupt request valid timing after EI instruction**	
	Addition of **5.9 Bit Manipulation Instruction of Interrupt Control Register During DMA Transfer**	
	Modification of description in **6.1 (1) Main clock oscillator**	**CHAPTER 6 CLOCK GENERATION FUNCTION**
	Modification of description in **6.1 (2) Subclock oscillator**	
	Modification of **Figure 6-1 Clock Generator**	
	Addition to **Notes** in **6.3.1 (1) Processor clock control register (PCC)**	
	Modification of description in **6.3.1 (1) (b) Example of subclock operation →main clock operation setup**	
	Addition to **Notes** and **Cautions** in **6.3.1 (2) Power save control register (PSC)**	
	Modification of description in **6.4.4 (1) Settings and operating states**	
	Addition of **6.6 Notes on Power Save Function**	
	Modification of **Caution** in **7.1.3 (2) Capture/compare registers 00, 10 (CR00, CR10)**	**CHAPTER 7 TIMER/COUNTER FUNCTION**
	Modification of **Caution** in **7.1.3 (3) Capture/compare registers 01, 11 (CR01, CR11)**	
	Change of **Figure 7-27 Data Hold Timing of Capture Register**	
	Addition of **7.2.7 (6) (c) One-shot output function**	
	Addition of **7.3.1 Outline**	
	Change of **Caution** in **7.3.4 (2) 8-bit timer mode control registers 2 to 5 (TMC2 to TMC5)**	
	Modification of description in **10.3.2 (3) IIC clock select register 0 (IICCL0), IIC function expansion register 0 (IICX0)**	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Addition of **Figures 10-25** to **10-29**	
	Modification in **11.3 (1) A/D converter mode register (ADM)**	**CHAPTER 11 A/D CONVERTER**
	Addition of **Table 11-2 A/D Conversion Time Selection**	
	Addition of **11.6 How to Read A/D Converter Characteristics Table**	
	Change of description in **12.1 Functions**	**CHAPTER 12 DMA FUNCTIONS**
	Deletion of **12.2 Transfer Completion Interrupt Request** and addition of **12.2 Features**	

(7/7)

Edition	Major Revision from Previous Edition	Applied to:
4th edition	Addition of **12.3 Configuration**	**CHAPTER 12 DMA FUNCTIONS**
	Addition of **Figure 12-2 Correspondence Between DRAn Setting Value and Internal RAM (4 KB)**	
	Addition of **Figure 12-3 Correspondence Between DRAn Setting Value and Internal RAM (8 KB)**	
	Addition of **12.5 Operations**	
	Addition of **12.6 Cautions**	
	Addition of **13.2 Features**	**CHAPTER 13 REAL-TIME OUTPUT FUNCTION (RTO)**
	Addition of **13.3 (2) Output latch**	
	Modification of description in **13.5 Usage**	
	Addition of description in **13.7 Cautions**	
	Addition of **Table 14-1 pin I/O Buffer Power Supplies**	**CHAPTER 14 PORT FUNCTION**
	Addition of **Caution** in **14.2.8 (1) Function of P9 pins**	
	Addition of **14.4 Operation of Port Function**	
	Addition of **Caution** in **CHAPTER 16 FLASH MEMORY**	**CHAPTER 16 FLASH MEMORY (μPD70F3017A, 70F3017AY)**
	Change of description in **16.1.1 Erasing unit**	
	Addition of **Figure 16-1 Wiring Example of V850/SA1 Flash Writing Adapter (FA100GC-8EU)**	
	Addition of **Table 16-1 Wiring Table of V850/SA1 Flash Writing Adapter (FA100GC-8EU)**	
	Addition of **Figure 16-2 Wiring Example of V850/SA1 Flash Writing Adapter (FA121F1-EA6) Wiring Example**	
	Addition of **Table 16-2 Wiring Table of V850/SA1 Flash Writing Adapter (FA121F1-EA6)**	
	Addition of **16.7 Flash Memory Programming by Self-Programming**	
	Addition of **CHAPTER 17 ELECTRICAL SPECIFICATIONS**	**CHAPTER 17 ELECTRICAL SPECIFICATIONS**
	Addition of **CHAPTER 18 PACKAGE DRAWINGS**	**CHAPTER 18 PACKAGE DRAWINGS**
	Addition of **CHAPTER 19 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 19 RECOMMENDED SOLDERING CONDITIONS**
	Addition of **APPENDIX A NOTES ON TARGET SYSTEM DESIGN**	**APPENDIX A NOTES ON TARGET SYSTEM DESIGN**
	Addition of description in **APPENDIX B REGISTER INDEX**	**APPENDIX B REGISTER INDEX**
	Addition of **APPENDIX E REVISION HISTORY**	**APPENDIX E REVISION HISTORY**