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

V850/SB1™, V850/SB2™

32-Bit Single-Chip Microcontroller

Hardware

μPD703031A	μPD703030B	μPD703034A	μPD703034B
μPD703031AY	μPD703030BY	μPD703034AY	μPD703034BY
μPD703032A	μPD703031B	μPD703035A	μPD703035B
μPD703032AY	μPD703031BY	μPD703035AY	μPD703035BY
μPD703033A	μPD703032B	μPD703037A	μPD703036H
μPD703033AY	μPD703032BY	μPD703037AY	μPD703036HY
μPD70F3032A	μPD703033B	μPD70F3035A	μPD703037H
μPD70F3032AY	μPD703033BY	μPD70F3035AY	μPD703037HY
μPD70F3033A	μPD70F3030B	μPD70F3037A	μPD70F3035B
μPD70F3033AY	μPD70F3030BY	μPD70F3037AY	μPD70F3035BY
	μPD70F3032B		μPD70F3036H
	μPD70F3032BY		μPD70F3036HY
	μPD70F3033B		μPD70F3037H
	μPD70F3033BY		μPD70F3037HY

Document No. U13850EJ6V0UD00 (6th edition)
Date Published February 2003 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① **PRECAUTION AGAINST ESD FOR SEMICONDUCTORS**

Note:

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 once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build 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 bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② **HANDLING OF UNUSED INPUT PINS FOR CMOS**

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ **STATUS BEFORE INITIALIZATION OF MOS DEVICES**

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Purchase of NEC I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

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.

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.

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60

- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
  Fax: 01-30-67 58 99

- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
  Fax: 02-66 75 42 99

- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
  Fax: 040-244 45 80

- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
  Fax: 08-63 80 388

- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133
  Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138
Fax: 021-6841-1137

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311
Fax: 6250-3583

J02.11

Major Revisions in This Edition (1/3)

Page	Description
Throughout	• Addition of the following products. μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY, 703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY, 703037H, 703037HY, 70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY, 70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY • Deletion of the following products. μPD703030A, 703030AY, 703036A, 703036AY
p. 63	Addition of description on minimum instruction execution time in **1.5.1**
p. 63	Addition of description on instruction set in **1.5.1**
p. 73	Addition of description in **Table 2-1 Pin I/O Buffer Power Supplies**
p. 80	Modification of description and addition of **Notes** in **Table 2-3 Operating States of Pins in Each Operating Mode**
p. 87	Addition of description in **2.3 (9) (b) (i) $\overline{\text{LBEN}}$**
p. 92	Modification of P23 I/O circuit type and description on P33 in **2.4 Pin I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins**
p. 96	Addition of description on minimum instruction execution time in **3.1**
p. 100	Modification of description and addition of **Note** in **3.2.2 (2) Program status word (PSW)**
p. 110	Addition of **3.4.5 (2) (a) V850/SB1 (μPD703031B, 703031BY), V850/SB2 (μPD703034B, 703034BY**)
pp. 119, 124, 125	Modification of **Note** and addition of registers in **3.4.8 Peripheral I/O registers**
p. 126	Addition of description in **3.4.9 Specific registers**
p. 126	Modification of **[Description example]** in **3.4.9 Specific registers**
p. 126	Modification of **Caution 2** in **3.4.9 Specific registers**
p. 127	Addition of **Remarks** in **3.4.9 (2) (b) Reset conditions (PRERR = 1)**
p. 129	Addition of **Note** and **Caution** in **4.2.2 (1) System control register (SYC)**
p. 158	Addition of **Remark** in **5.3.3 Priorities of maskable interrupts**
p. 162	Addition of **Caution 2** in **5.3.4 Interrupt control register (xxICn)**
p. 165	Addition of **Caution** in **5.3.5 In-service priority register (ISPR)**
p. 165	Addition of **Remark** in **5.3.6 ID flag**
p. 176	Addition of **Remark** in **5.6.2 (2) To generate exception in service program**
p. 178	Addition of **5.8.1 Interrupt request valid timing after EI instruction**
p. 179	Addition of **5.9 Interrupt Control Register Bit Manipulation Instructions During DMA Transfer**
p. 184	Addition of description in **Cautions** in **6.3.1 (1) Processor clock control register (PCC)**
p. 185	Modification of description in **6.3.1 (1) (b) Example of subclock operation → main clock operation setup**
p. 186	Modification of description in **6.3.1 (2) Power save control register (PSC)**
pp. 190, 191	Addition and deletion of description in **Table 6-1 Operating Statuses in HALT Mode**
p. 192	Modification of description in **Table 6-2 Operating Statuses in IDLE Mode**
p. 194	Addition of description in **6.4.4 (1) Settings and operating states**
p. 194	Modification of description in **Table 6-3 Operating Statuses in Software STOP Mode**

Major Revisions in This Edition (2/3)

Pages	Description
p. 197	Addition of **6.6 (1) While an instruction is being executed on internal ROM**
p. 198	Addition of **6.6 (2) While an instruction is being executed on external ROM**
p. 206	Addition of description in **Caution** in **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**
p. 207	Addition of description in **Caution** in **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**
p. 213	Modification of description in **Figure 7-5 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**
p. 214	Addition of **Figure 7-6 Configuration of PPG Output**
p. 214	Addition of **Figure 7-7 PPG Output Operation Timing**
p. 215	Modification of description in **Figure 7-8 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**
p. 217	Modification of description in **Figure 7-11 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**
p. 219	Modification of description in **Figure 7-14 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**
p. 221	Modification of description in **Figure 7-17 Timing of Pulse Width Measurement by Restarting (with Rising Edge Specified)**
p. 222	Modification of description in **Figure 7-18 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**
p. 227	Modification of description in **Caution** in **7.2.6 (2) One-shot pulse output with external trigger**
p. 231	Modification of description in **7.2.7 (6) (a) One-shot pulse output by software**
p. 231	Modification of description in **7.2.7 (6) (b) One-shot pulse output with external trigger**
p. 234	Addition of **7.3.1 Outline**
p. 242	Change of **Figure 7-32 Timing of Interval Timer Operation (3/3)**
p. 246	Addition of description to **Remarks** in **Figure 7-34 Square Wave Output Operation Timing**
p. 248	Addition of description to **Remarks** in **Figure 7-35 Timing of PWM Output**
p. 253	Addition of registers and **Caution** in **Figure 8-1 Block Diagram of Watch Timer**
p. 254	Addition of registers and **Note** in **Table 8-2 Configuration of Watch Timer**
p. 255	Addition of description and **Caution** in **8.3 Watch Timer Control Register**
p. 256	Addition of **8.3 (2) Watch timer high-speed clock selection register (WTNHC)**
p. 257	Addition of description in **8.3 (3) Watch timer clock selection register (WTNCS)**
p. 263	Addition of **Caution** in **9.3 (2) Watchdog timer clock selection register (WDCS)**
p. 268	Addition of description in **10.2 (2) 3-wire serial I/O mode (fixed to MSB first)**
p. 280	Modification of **Caution** in **10.3.2 (1) IIC control registers 0, 1 (IICC0, IICC1)**
p. 288	Addition of **Caution** in **10.3.2 (3) IIC clock selection registers 0, 1 (IICCL0, IICCL1)**
p. 334	Addition of **10.4 I^{2}C Bus (B and H Versions)**
p. 402	Addition of description to **Cautions** in **10.5.2 (1) Asynchronous serial interface mode registers 0, 1 (ASIM0, ASIM1)**
p. 405	Addition of description to **Cautions** in **10.5.2 (4) Baud rate generator mode control registers n0, n1 (BRGMCn0, BRGMCn1)**
p. 406	Addition of description to **Cautions** in **Figure 10-45 ASIMn Setting (Operation Stop Mode)**
p. 407	Addition of description to **Cautions** in **Figure 10-46 ASIMn Setting (Asynchronous Serial Interface Mode)**
p. 410	Addition of description to **Cautions** in **Figure 10-49 BRGMCn0 and BRGMCn1 Settings (Asynchronous Serial Interface Mode)**

Major Revisions in This Edition (3/3)

Pages	Description
p. 437	Addition of **Caution** in **11.3 (2) Analog input channel specification register (ADS)**
p. 447	Addition of **11.7 How to Read A/D Converter Characteristics Table**
p. 452	Addition of **12.3 Configuration**
p. 455	Addition of **12.4 (2) (a) V850/SB1 (μPD703031B, 703031BY), V850/SB2 (μPD703034B, 703034BY)**
p. 460	Addition of **Caution** in **12.4 (5) DMA channel control registers 0 to 5 (DCHC0 to DCHC5)**
p. 462	Addition of **12.5 Operation**
p. 463	Addition of **12.6 Cautions**
p. 466	Addition of **13.2 Features**
p. 468	Addition of **13.3 (2) Output latch**
p. 471	Addition of description in **13.5 Usage**
p. 473	Addition of **13.7 (3)**
p. 474	Addition of description in **Table 14-1 Pin I/O Buffer Power Supplies**
p. 501	Addition of **Caution** in **14.2.8 (1) Function of P9 pins**
p. 504	Addition of **Caution** in **14.2.9 (1) Function of P10 pins**
p. 511, 512	Addition of description in **Table 14-12 Setting When Port Pin is Used as Alternate Function**
p. 515	Addition of **14.4 Port Function Operation**
p. 517	Addition of description in **16.1 Outline**
p. 517	Addition of description in **Figure 16-1 Regulator**
p. 524	Addition of **18.1.1 (2) V850/SB1 (μPD70F3030B, 70F3030BY), V850/SB2 (μPD70F3036H, 70F3036HY)**
p. 526	Addition of **Figure 18-1 Wiring Example of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GC-8EU)**
p. 527	Addition of **Table 18-1 Table for Wiring of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GC-8EU)**
p. 528	Addition of **Figure 18-2 Wiring Example of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GF-3BA)**
p. 529	Addition of **Table 18-2 Table for Wiring of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GF-3BA)**
p. 549	Modification of description in **Table 19-5 Acknowledge Signal Output Condition of Control Field**
p. 560	Addition of register to **Table 19-7 Internal Registers of IEBus Controller**
p. 582	Addition of **Remark** in **19.3.2 (13) IEBus clock selection register (IECLK)**
p. 583	Addition of **19.3.2 (14) IEBus high-speed clock selection register (IEHCLK)**
p. 600	Addition of **CHAPTER 20 ELECTRICAL SPECIFICATIONS**
p. 635	Addition of **CHAPTER 21 PACKAGE DRAWINGS**
p. 637	Addition of **CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**
p. 642	Addition of **APPENDIX A NOTES ON TARGET SYSTEM DESIGN**
p. 658	Modification of **APPENDIX D INDEX**
p. 664	Addition of **APPENDIX E REVISION HISTORY**

The mark ★ shows major revised points.

INTRODUCTION

Readers This manual is intended for users who wish to understand the functions of the V850/SB1 and V850/SB2 and design application systems using the V850/SB1 or V850/SB2.

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

Organization The V850/SB1, V850/SB2 User's Manual is divided into two parts: hardware (this manual) and architecture (V850 Series™ Architecture User's Manual).

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
• IEBus controller (V850/SB2 only)	• Pipeline operation
• Electrical specifications	

How to Read 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 Architecture User's Manual** available separately.

To know the electrical specifications of the V850/SB1 and V850/SB2:
→ Refer to **CHAPTER 20 ELECTRICAL SPECIFICATIONS**.

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/SB1 and V850/SB2:
→ Read this manual in accordance with the **CONTENTS**.

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

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

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

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

Caution: Information requiring particular attention

Remark: Supplementary information

Number 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.

Related documents for V850/SB1 and V850/SB2

Document Name	Document No.
V850 Series Architecture User's Manual	U10243E
V850/SB1, V850/SB2 Hardware User's Manual	This manual

Related documents for development tool (user's manual)

Document Name		Document No.
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 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 (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/SB1 and V850/SB2 are products in the NEC Electronics V850 Series of single-chip microcontrollers designed for low power operation.

1.1 General

The V850/SB1 and V850/SB2 are 32-bit single-chip microcontrollers that include the V850 Series CPU core, and peripheral functions such as ROM/RAM, a timer/counter, a serial interface, an A/D converter, a timer, and DMA controller.

Based on the V850/SA1™, the V850/SB1 and V850/SB2 feature various additions, including 3 to 5 V I/O interface support, and ROM correction. For V850/SB2, based on the V850/SB1™, the peripheral functions of automobile LAN (IEBus™ (Inter Equipment Bus™)) are added. In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850/SB1 and V850/SB2 have 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/SB1 and V850/SB2 enable the realization of extremely high cost-performance for applications that require low power consumption, such as audio equipment, car audio systems, and VCRs.

Table 1-1 shows the outlines of the V850/SB1 and V850/SB2 product lineup.

★

Table 1-1. Product Lineup of V850/SB1

Product Name		On-Chip I^2C	ROM		RAM Size	Package	On-Chip IEBus
Commercial Name	Part Number		Type	Size			
V850/SB1	μPD703031A	No	Mask ROM	128 KB	12 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	No
	μPD703031AY	Yes					
	μPD703033A	No	Mask ROM	256 KB	16 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD70F3033A		Flash memory				
	μPD703033AY	Yes	Mask ROM				
	μPD70F3033AY		Flash memory				
	μPD703032A	No	Mask ROM	512 KB	24 KB	100-pin QFP (14 × 20)	
	μPD70F3032A		Flash memory				
	μPD703032AY	Yes	Mask ROM				
	μPD70F3032AY		Flash memory				
	μPD703031B	No	Mask ROM	128 KB	8 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD703031BY	Yes					
	μPD703033B	No	Mask ROM	256 KB	16 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD70F3033B		Flash memory				
	μPD703033BY	Yes	Mask ROM				
	μPD70F3033BY		Flash memory				
	μPD703030B	No	Mask ROM	384 KB	24 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD70F3030B		Flash memory				
	μPD703030BY	Yes	Mask ROM				
	μPD70F3030BY		Flash memory				
	μPD703032B	No	Mask ROM	512 KB	24 KB	100-pin QFP (14 × 20)	
	μPD70F3032B		Flash memory				
	μPD703032BY	Yes	Mask ROM				
	μPD70F3032BY		Flash memory				

★ **Table 1-2. Product Lineup of V850/SB2**

Product Name		On-Chip I^2C	ROM		RAM Size	Package	On-Chip IEBus
Commercial Name	Part Number		Type	Size			
V850/SB2	μPD703034A	No	Mask ROM	128 KB	12 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	Yes
	μPD703034AY	Yes					
	μPD703035A	No	Mask ROM	256 KB	16 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD70F3035A		Flash memory				
	μPD703035AY	Yes	Mask ROM				
	μPD70F3035AY		Flash memory				
	μPD703037A	No	Mask ROM	512 KB	24 KB	100-pin QFP (14 × 20)	
	μPD70F3037A		Flash memory				
	μPD703037AY	Yes	Mask ROM				
	μPD70F3037AY		Flash memory				
	μPD703034B	No	Mask ROM	128 KB	8 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD703034BY	Yes					
	μPD703035B	No	Mask ROM	256 KB	16 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD70F3035B		Flash memory				
	μPD703035BY	Yes	Mask ROM				
	μPD70F3035BY		Flash memory				
	μPD703036H	No	Mask ROM	384 KB	24 KB	100-pin QFP (14 × 20)/ 100-pin LQFP (14 × 14)	
	μPD70F3036H		Flash memory				
	μPD703036HY	Yes	Mask ROM				
	μPD70F3036HY		Flash memory				
	μPD703037H	No	Mask ROM	512 KB	24 KB	100-pin QFP (14 × 20)	
	μPD70F3037H		Flash memory				
	μPD703037HY	Yes	Mask ROM				
	μPD70F3037HY		Flash memory				

★ The part numbers of the V850/SB1 and V850/SB2 are described as follows in this manual.

- A versions
  A versions of the V850/SB1: μPD703031A, 703031AY, 703032A, 703032AY, 703033A, 703033AY, 70F3032A, 70F3032AY, 70F3033A, 70F3033AY
  A versions of the V850/SB2: μPD703034A, 703034AY, 703035A, 703035AY, 703037A, 703037AY, 70F3035A, 70F3035AY, 70F3037A, 70F3037AY
- B versions, H versions
  B versions of the V850/SB1: μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY, 70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY
  B and H versions of the V850/SB2: μPD703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY, 703037H, 703037HY, 70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY

- Flash memory versions
  Flash memory versions of the V850/SB1: μPD70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY, 70F3033A, 70F3033AY, 70F3033B, 70F3033BY
  Flash memory versions of the V850/SB2: μPD70F3035A, 70F3035AY, 70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037A, 70F3037AY, 70F3037H, 70F3037HY
- Mask ROM versions
  Mask ROM versions of the V850/SB1: μPD703030B, 703030BY, 703031A, 703031AY, 703031B, 703031BY, 703032A, 703032AY, 703032B, 703032BY, 703033A, 703033AY, 703033B, 703033BY
  Mask ROM versions of the V850/SB2: μPD703034A, 703034AY, 703034B, 703034BY, 703035A, 703035AY, 703035B, 703035BY, 703036H, 703036HY, 703037A, 703037AY, 703037H, 703037HY
- Y versions (with on-chip I^2C)
  Y versions of the V850/SB1 (with on-chip I^2C): μPD703030BY, 703031AY, 703031BY, 703032AY, 703032BY, 703033AY, 703033BY, 70F3030BY, 70F3032AY, 70F3032BY, 70F3033AY, 70F3033BY
  Y versions of the V850/SB2 (with on-chip I^2C): μPD703034AY, 703034BY, 703035AY, 703035BY, 703036HY, 703037AY, 703037HY, 70F3035AY, 70F3035BY, 70F3036HY, 70F3037AY, 70F3037BY

1.2 V850/SB1 (A Versions)

1.2.1 Features (V850/SB1 (A versions))

○ Number of instructions: 74	
○ Minimum instruction execution time	
	50 ns (operating at 20 MHz, external power supply 5 V, regulator output 3.3 V)
○ General-purpose registers	32 bits × 32 registers
○ Instruction set	Signed multiplication (16 × 16 → 32): 100 ns (operating at 20 MHz) (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 3 V to 5 V interface enabled Bus hold function External wait function
○ Internal memory	μPD703031A, 703031AY (mask ROM: 128 KB/RAM: 12 KB) μPD703033A, 703033AY (mask ROM: 256 KB/RAM: 16 KB) μPD703032A, 703032AY (mask ROM: 512 KB/RAM: 24 KB) μPD70F3033A, 70F3033AY (flash memory: 256 KB/RAM: 16 KB) μPD70F3032A, 70F3032AY (flash memory: 512 KB/RAM: 24 KB)
○ Interrupts and exceptions	Non-maskable interrupts: 2 sources Maskable interrupts: 37 sources (μPD703031A, 703032A, 703033A, 70F3032A, 70F3033A) 38 sources (μPD703031AY, 703032AY, 703033AY, 70F3032AY, 70F3033AY) Software exceptions: 32 sources Exception trap: 1 source
○ I/O lines	Total: 83 (12 input ports and 71 I/O ports) 3 V to 5 V interface enabled
○ Timer/counter	16-bit timer: 2 channels (PWM output) 8-bit timer: 6 channels (4 PWM outputs, cascade connection enabled)
○ Watch timer	When operating under subclock or main clock: 1 channel Operation using the subclock or main clock is also possible in the IDLE mode.
○ Watchdog timer	1 channel

○ Serial interface (SIO)	Asynchronous serial interface (UART) Clocked serial interface (CSI) I^2C bus interface (I^2C) (only for μPD703031AY, 703032AY, 703033AY, 70F3032AY, and 70F3033AY) 8-/16-bit variable-length serial interface CSI/UART: 2 channels CSI/I^2C: 2 channels CSI (8-/16-bit valuable): 1 channel Dedicated baud rate generator: 3 channels
○ A/D converter	10-bit resolution: 12 channels
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels
○ Real-time output port (RTP)	8 bits × 1 channel or 4 bits × 2 channels
○ ROM correction	Modifiable 4 points
○ Regulator	4.0 V to 5.5 V input → internal 3.3 V
○ Key return function	4 to 8 selecting enabled, falling edge fixed
○ Clock generator	During main clock or subclock operation 5-level CPU clock (including sub operations)
○ Power-saving functions	HALT/IDLE/STOP modes
○ Package	100-pin plastic LQFP (fine pitch, 14 × 14) 100-pin plastic QFP (14 × 20)
○ CMOS structure	All static circuits

1.2.2 Application fields (V850/SB1 (A versions))

AV equipment

Example: Audio, car audio equipment, VCR, and TV.

1.2.3 Ordering information (V850/SB1 (A versions))

Part Number	Package	Internal ROM
μPD703031AGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703031AGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703031AYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703031AYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703033AGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703033AGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703033AYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703033AYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703032AGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD703032AYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD70F3033AGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3033AGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3033AYGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3033AYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3032AGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)
μPD70F3032AYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)

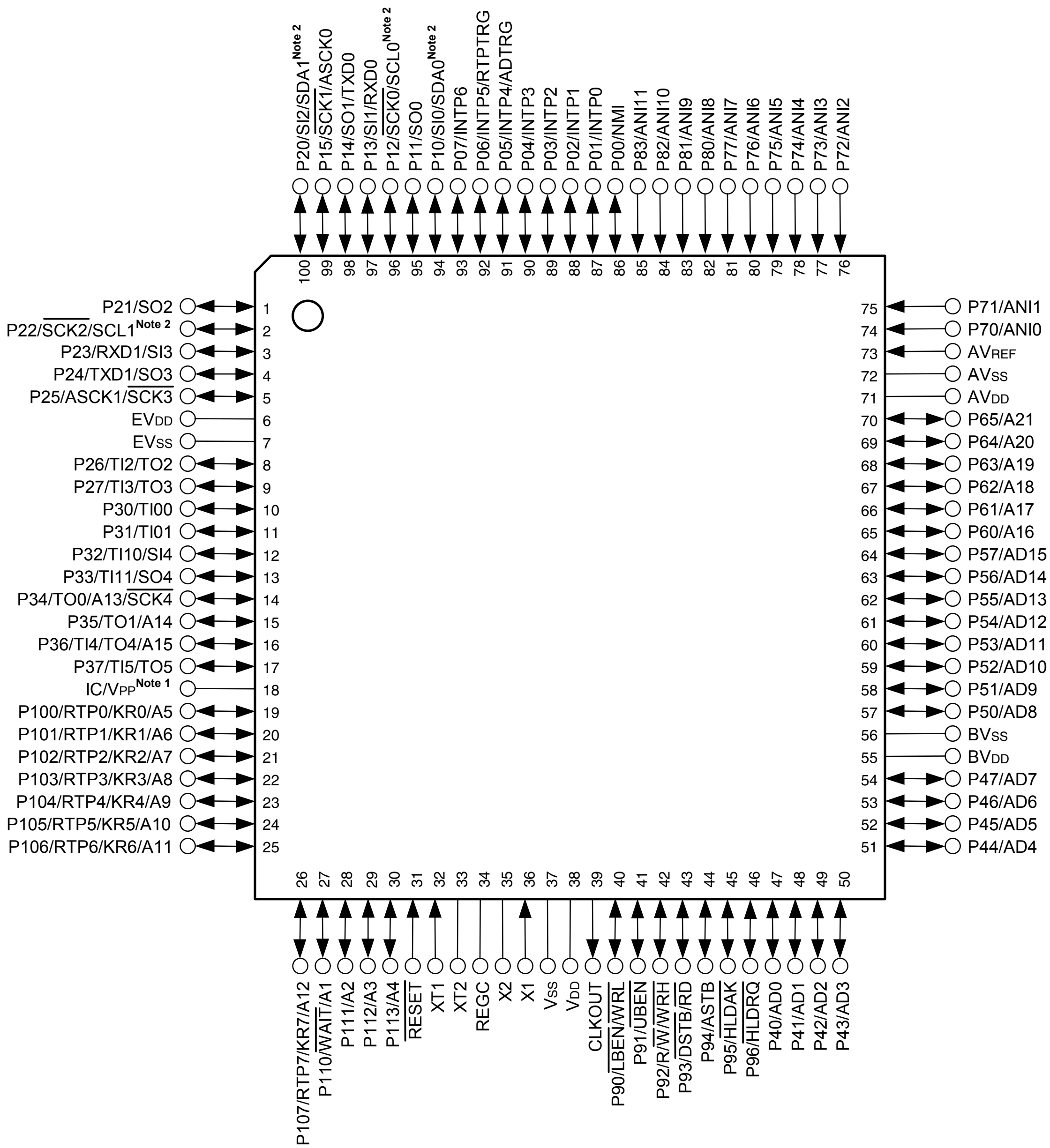
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.2.4 Pin configuration (top view) (V850/SB1 (A versions))

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

- μPD703031AGC-xxx-8EU
- μPD703031AYGC-xxx-8EU
- μPD703033AGC-xxx-8EU
- μPD703033AYGC-xxx-8EU
- μPD70F3033AGC-8EU
- μPD70F3033AYGC-8EU

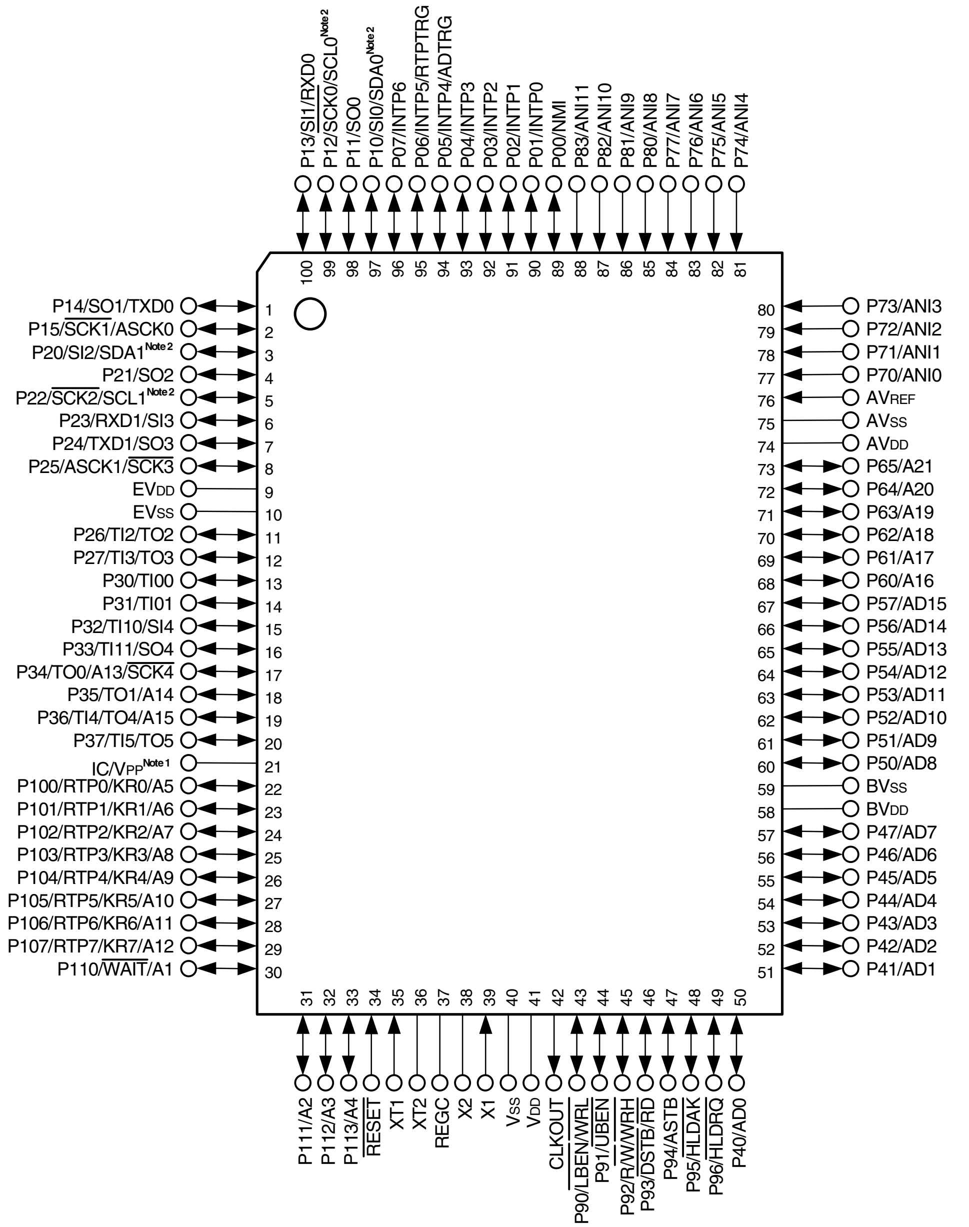


Notes 1. IC (μPD703031A, 703031AY, 703033A, 703033AY): Connect directly to V_{SS}.
V_{PP} (μPD70F3033A, 70F3033AY): Connect to V_{SS} in normal operation mode.

2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703031AY, 703033AY, and 70F3033AY.

100-pin plastic QFP (14 × 20)

- μPD703031AGF-xxx-3BA
- μPD703031AYGF-xxx-3BA
- μPD703032AGF-xxx-3BA
- μPD703032AYGF-xxx-3BA
- μPD703033AGF-xxx-3BA
- μPD703033AYGF-xxx-3BA
- μPD70F3032AGF-3BA
- μPD70F3032AYGF-3BA
- μPD70F3033AGF-3BA
- μPD70F3033AYGF-3BA



Notes 1. IC (μPD703031A, 703031AY, 703032A, 703032AY, 703033A, 703033AY):
Connect directly to VSS.
VPP (μPD70F3032A, 70F3032AY, 70F3033A, 70F3033AY):
Connect to VSS in normal operation mode.

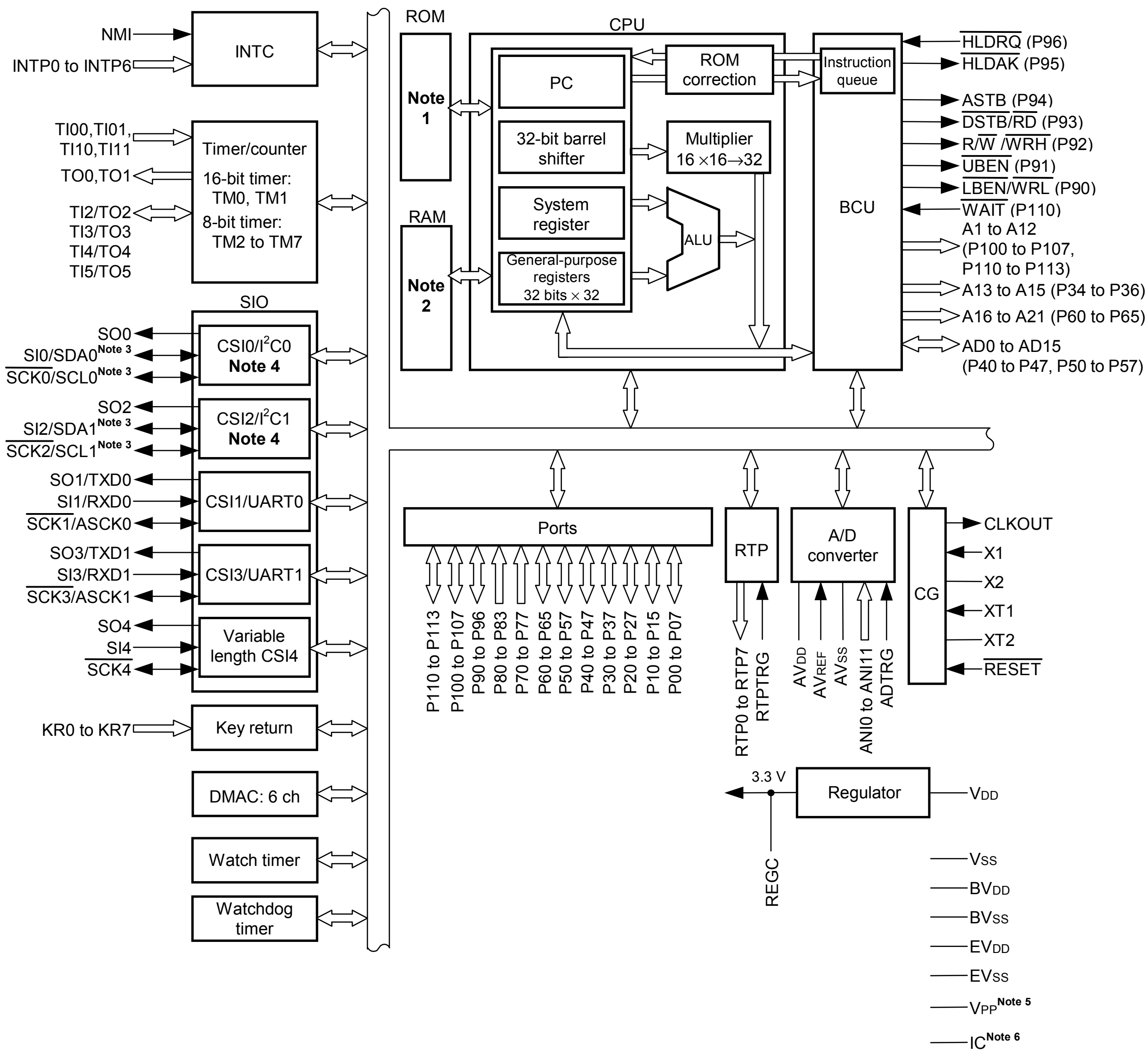
2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703031AY, 703032AY, 703033AY, 70F3032AY, and 70F3033AY.

Pin names (V850/SB1 (A versions))

A1 to A21:	Address bus	P70 to P77:	Port 7
AD0 to AD15:	Address/data bus	P80 to P83:	Port 8
ADTRG:	A/D trigger input	P90 to P96:	Port 9
ANI0 to ANI11:	Analog input	P100 to P107:	Port 10
ASCK0, ASCK1:	Asynchronous serial clock	P110 to P113:	Port 11
ASTB:	Address strobe	$\overline{\text{RD}}$:	Read
AV_{DD}:	Analog V_{DD}	REGC:	Regulator control
AV_{REF}:	Analog reference voltage	$\overline{\text{RESET}}$:	Reset
AV_{SS}:	Analog V_{SS}	RTP0 to RTP7:	Real-time output port
BV_{DD}:	Power supply for bus interface	RTPTRG:	RTP trigger
BV_{SS}:	Ground for bus interface	R/$\overline{\text{W}}$:	Read/write status
CLKOUT:	Clock output	RXD0, RXD1:	Receive data
$\overline{\text{DSTB}}$:	Data strobe	$\overline{\text{SCK0}}$ to $\overline{\text{SCK4}}$:	Serial clock
EV_{DD}:	Power supply for port	SCL0, SCL1:	Serial clock
EV_{SS}:	Ground for port	SDA0, SDA1:	Serial data
$\overline{\text{HLDAK}}$:	Hold acknowledge	SI0 to SI4:	Serial input
$\overline{\text{HLDRQ}}$:	Hold request	SO0 to SO4:	Serial output
IC:	Internally connected	TI00, TI01, TI10,	
INTP0 to INTP6:	Interrupt request from peripherals	TI11, TI2 to TI5:	Timer input
KR0 to KR7:	Key return	TO0 to TO5:	Timer output
$\overline{\text{LBEN}}$:	Lower byte enable	TXD0,TXD1:	Transmit data
NMI:	Non-maskable interrupt request	$\overline{\text{UBEN}}$:	Upper byte enable
P00 to P07:	Port 0	V_{DD}:	Power supply
P10 to P15:	Port 1	V_{PP}:	Programming power supply
P20 to P27:	Port 2	V_{SS}:	Ground
P30 to P37:	Port 3	$\overline{\text{WAIT}}$:	Wait
P40 to P47:	Port 4	$\overline{\text{WRH}}$:	Write strobe high level data
P50 to P57:	Port 5	$\overline{\text{WRL}}$:	Write strobe low level data
P60 to P65:	Port 6	X1, X2:	Crystal for main clock
		XT1, XT2:	Crystal for subclock

1.2.5 Function blocks (V850/SB1 (A versions))

(1) Internal block diagram



Notes 1. μPD703031A, 703031AY: 128 KB (mask ROM)
μPD703033A, 703033AY: 256 KB (mask ROM)
μPD703032A, 703032AY: 512 KB (mask ROM)
μPD70F3033A, 70F3033AY: 256 KB (flash memory)
μPD70F3032A, 70F3032AY: 512 KB (flash memory)

2. μPD703031A, 703031AY: 12 KB
μ PD703033A, 703033AY, 70F3033A, 70F3033AY: 16 KB
μPD703032A, 703032AY, 70F3032A, 70F3032AY: 24 KB

3. SDA0, SDA1, SCL0, and SCL1 pins are available only in the μPD703031AY, 703032AY, 703033AY, 70F3032AY, and 70F3033AY.

4. I^{2}C function is available only in the μPD703031AY, 703032AY, 703033AY, 70F3032AY, and 70F3033AY.

5. μPD70F3032A, 70F3032AY, 70F3033A, 70F3033AY

6. μPD703031A, 703031AY, 703032A, 703032AY, 703033A, 703033AY

(2) Internal units

(a) 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 barrel shifter (32 bits) help accelerate processing of complex instructions.

(b) 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.

(c) ROM

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

The ROM capacity varies depending on the product. The ROM capacity of each product is shown below.

μPD703031A, 703031AY:	128 KB (mask ROM)
μPD703033A, 703033AY:	256 KB (mask ROM)
μPD70F3033A, 70F3033AY:	256 KB (flash memory)
μPD703032A, 703032AY:	512 KB (mask ROM)
μPD70F3032A, 70F3032AY:	512 KB (flash memory)

ROM can be accessed by the CPU in one clock cycle during instruction fetch.

(d) RAM

The RAM capacity and mapping addresses vary depending on the product. The RAM capacity of each product is shown below.

μPD703031A, 703031AY:	12 KB (mapping starts at FFFFC000H)
μPD703033A, 703033AY, 70F3033A, 70F3033AY:	16 KB (mapping starts at FFFFB000H)
μPD703032A, 703032AY, 70F3032A, 70F3032AY:	24 KB (mapping starts at FFFF9000H)

RAM can be accessed by the CPU in one clock cycle during data access.

(e) 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 multiplexed servicing control can be performed for interrupt sources.

(f) Clock generator (CG)

The clock generator includes two types of oscillators; each for main clock (f_{XX}) and for 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}).

(g) Timer/counter

A two-channel 16-bit timer/event counter, a four-channel 8-bit timer/event counter, and a two-channel 8-bit interval timer 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 to enable use as a 16-bit timer.

The two-channel 8-bit interval timer can be connected via a cascade to enable to be used as a 16-bit timer.

(h) Watch timer

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

(i) Watchdog timer

A watchdog timer is equipped to detect inadvertent 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.

(j) Serial interface (SIO)

The V850/SB1 includes three kinds of serial interfaces: asynchronous serial interfaces (UART0, UART1), clocked serial interfaces (CSI0 to CSI3), and an 8-/16-bit variable-length serial interface (CSI4). These plus the I^{2}C bus interfaces (I^{2}C0, I^{2}C1) comprise five channels. Two of these channels are switchable between the UART and CSI and another two switchable between CSI and I^{2}C.

For UART0 and UART1, data is transferred via the TXD0, TXD1, RXD0, and RXD1 pins.

For CSI0 to CSI3, data is transferred via the SO0 to SO3, SI0 to SI3, and $\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$ pins.

For CSI4, data is transferred via the SO4, SI4, and $\overline{\text{SCK4}}$ pins.

For I^{2}C0 and I^{2}C1, data is transferred via the SDA0, SDA1, SCL0, and SCL1 pins.

I^{2}C0 and I^{2}C1 are equipped only in the μPD703031AY, 703032AY, 703033AY, 70F3032AY, and 70F3033AY.

For UART and CSI4, a dedicated baud rate generator is equipped.

(k) A/D converter

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

(l) DMA controller

A six-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.

(m) Real-time output port (RTP)

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

(n) 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, serial interface
Port 4	8-bit I/O		External address/data bus
Port 5	8-bit I/O		
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, key return input
Port 11	4-bit I/O		Wait control, external address bus

★

1.3 V850/SB1 (B Versions)

1.3.1 Features (V850/SB1 (B versions))

○ Number of instructions: 74	
○ Minimum instruction execution time	
	50 ns (@ 20 MHz operation, external power supply 5 V, regulator output 3.3 V operation)
○ 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
	3 V to 5 V interface enabled
	Bus hold function
	External wait function
○ Internal memory	μPD703031B, 703031BY (mask ROM: 128 KB/RAM: 8 KB)
	μPD703033B, 703033BY (mask ROM: 256 KB/RAM: 16 KB)
	μPD703030B, 703030BY (mask ROM: 384 KB/RAM: 24 KB)
	μPD703032B, 703032BY (mask ROM: 512 KB/RAM: 24 KB)
	μPD70F3033B, 70F3033BY (flash memory: 256 KB/RAM: 16 KB)
	μPD70F3030B, 70F3030BY (flash memory: 384 KB/RAM: 24 KB)
	μPD70F3032B, 70F3032BY (flash memory: 512 KB/RAM: 24 KB)
○ Interrupts and exceptions	Non-maskable interrupts: 2 sources
	Maskable interrupts: 37 sources (μPD703030B, 703031B, 703032B, 703033B, 70F3030B, 70F3032B, 70F3033B)
	38 sources (μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3030BY, 70F3032BY, 70F3033BY)
	Software exceptions: 32 sources
	Exception trap: 1 source
○ I/O lines	Total: 83 (12 input ports and 71 I/O ports)
	3 V to 5 V interface enabled
○ Timer/counter	16-bit timer: 2 channels (PWM output)
	8-bit timer: 6 channels (4 PWM outputs, cascade connection enabled)
○ Watch timer	When operating under subclock or main clock: 1 channel
	Operation using the subclock or main clock is also possible in the IDLE mode.
○ Watchdog timer	1 channel

○ Serial interface (SIO)	Asynchronous serial interface (UART) Clocked serial interface (CSI) I^2C bus interface (I^2C) (only for μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3030BY, 70F3032BY, and 70F3033BY) 8-/16-bit variable-length serial interface CSI/UART: 2 channels CSI/I^2C: 2 channels CSI (8-/16-bit valuable): 1 channel Dedicated baud rate generator: 3 channels
○ A/D converter	10-bit resolution: 12 channels
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels
○ Real-time output port (RTP)	8 bits × 1 channel or 4 bits × 2 channels
○ ROM correction	Modifiable 4 points
○ Regulator	4.0 V to 5.5 V input → internal 3.3 V
○ Key return function	4 to 8 selecting enabled, falling edge fixed
○ Clock generator	During main clock or subclock operation 5-level CPU clock (including sub operations)
○ Power-saving functions	HALT/IDLE/STOP modes
○ Package	100-pin plastic LQFP (fine pitch, 14 × 14) 100-pin plastic QFP (14 × 20)
○ CMOS structure	All static circuits

1.3.2 Application fields (V850/SB1 (B versions))

AV equipment

Example: Audio, car audio equipment, VCR, and TV.

1.3.3 Ordering information (V850/SB1 (B versions))

Part Number	Package	Internal ROM
μPD703031BGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703031BGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703031BYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703031BYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703033BGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703033BGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703033BYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703033BYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703030BGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (384 KB)
μPD703030BGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (384 KB)
μPD703030BYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (384 KB)
μPD703030BYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (384 KB)
μPD703032BGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD703032BYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD70F3033BGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3033BGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3033BYGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3033BYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3030BGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (384 KB)
μPD70F3030BGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (384 KB)
μPD70F3030BYGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (384 KB)
μPD70F3030BYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (384 KB)
μPD70F3032BGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)
μPD70F3032BYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)

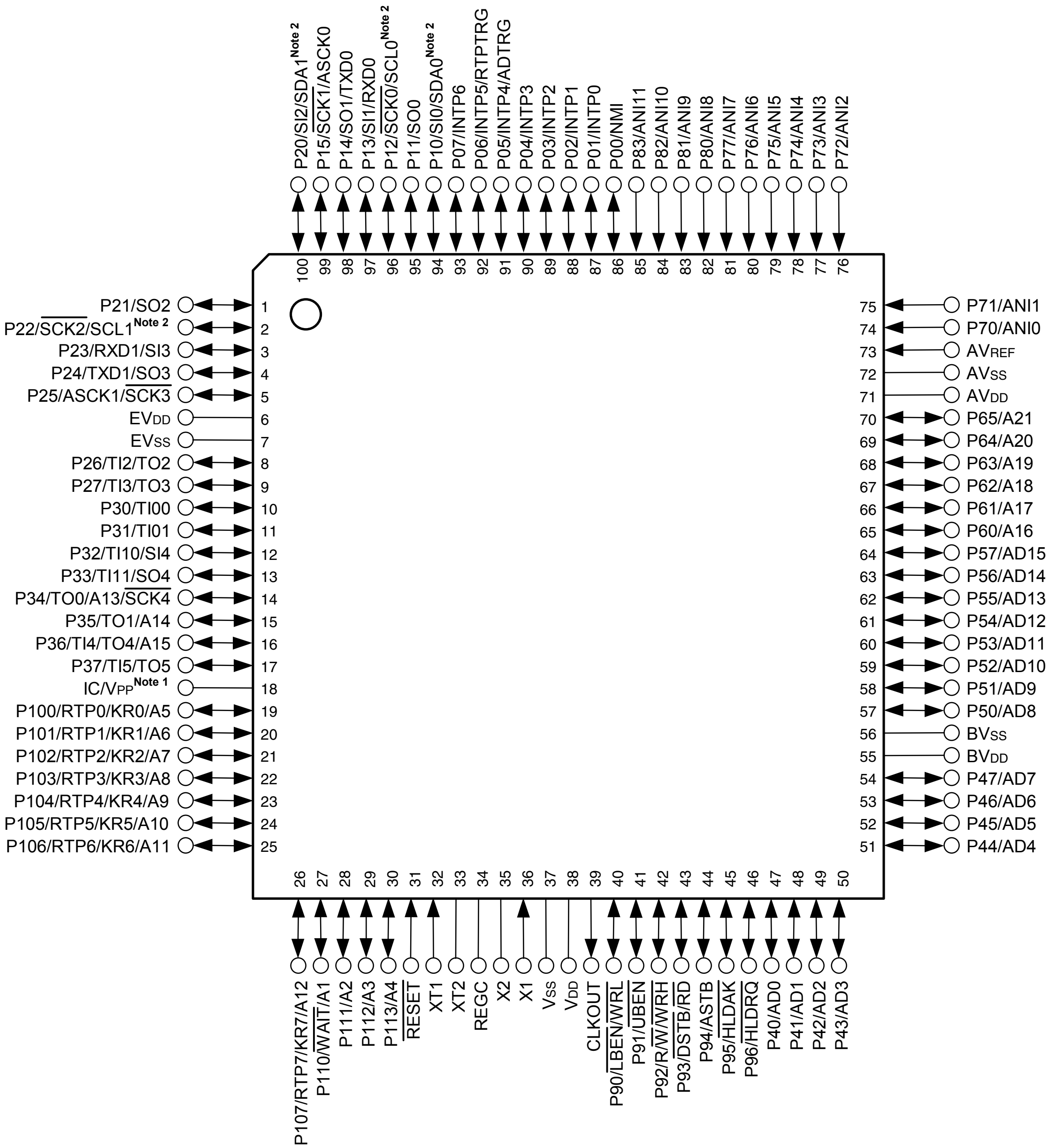
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.3.4 Pin configuration (top view) (V850/SB1 (B versions))

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

- μPD703030BGC-xxx-8EU
- μPD703030BYGC-xxx-8EU
- μPD703031BGC-xxx-8EU
- μPD703031BYGC-xxx-8EU
- μPD703033BGC-xxx-8EU
- μPD703033BYGC-xxx-8EU
- μPD70F3030BGC-8EU
- μPD70F3030BYGC-8EU
- μPD70F3033BGC-8EU
- μPD70F3033BYGC-8EU

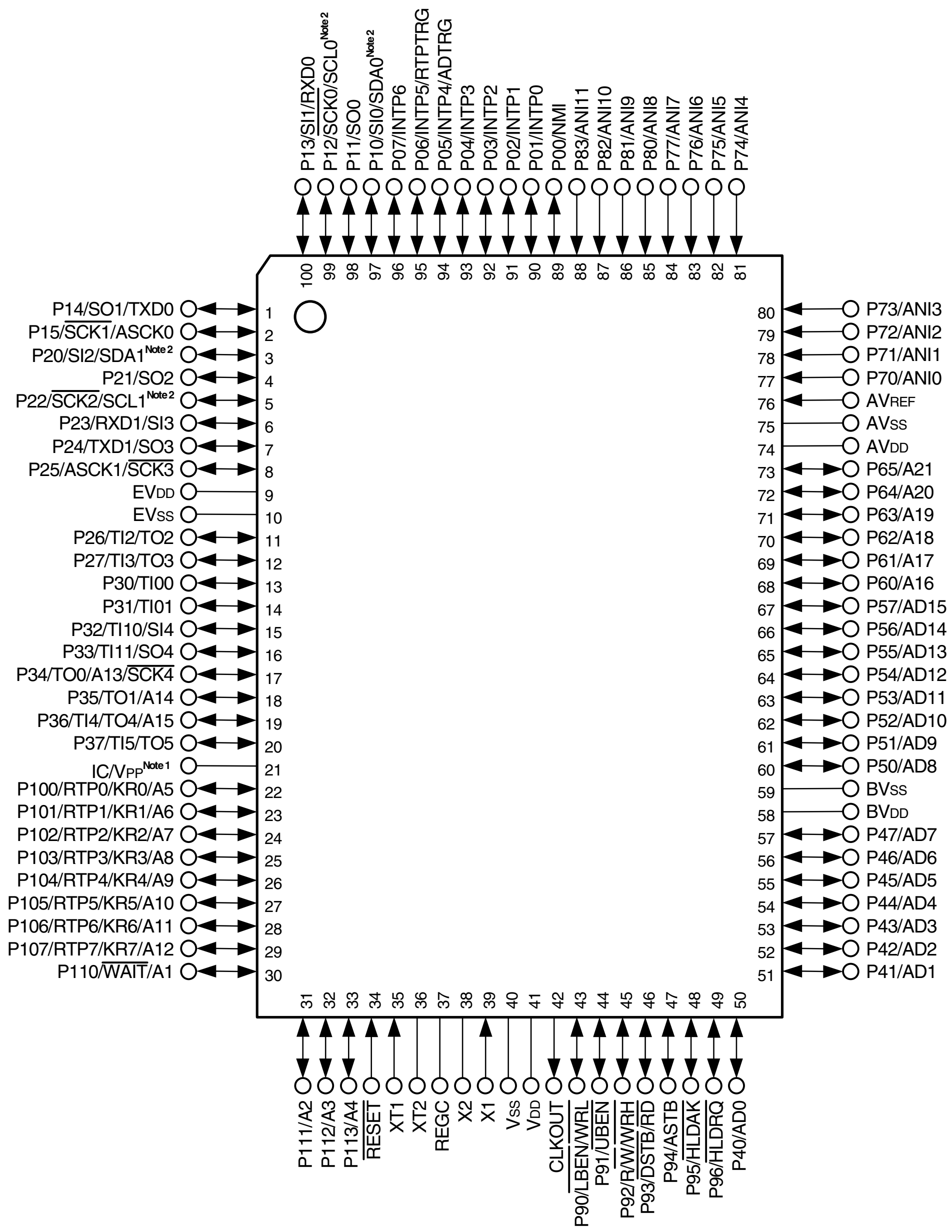


Notes 1. IC (μPD703030B, 703030BY, 703031B, 703031BY, 703033B, 703033BY): Connect directly to V_{SS}.
V_{PP} (μPD70F3030B, 70F3030BY, 70F3033B, 70F3033BY): Connect to V_{SS} in normal operation mode.

2. SCL0, SCL1, SDA0, and SDA1 are available only in the 703030BY, μPD703031BY, 703033BY, 70F3030BY, and 70F3033BY.

100-pin plastic QFP (14 × 20)

- μPD703030BGF-xxx-3BA
- μPD703030BYGF-xxx-3BA
- μPD703031BGF-xxx-3BA
- μPD703031BYGF-xxx-3BA
- μPD703032BGF-xxx-3BA
- μPD703032BYGF-xxx-3BA
- μPD703033BGF-xxx-3BA
- μPD703033BYGF-xxx-3BA
- μPD70F3030BGF-3BA
- μPD70F3030BYGF-3BA
- μPD70F3032BGF-3BA
- μPD70F3032BYGF-3BA
- μPD70F3033BGF-3BA
- μPD70F3033BYGF-3BA



Notes 1. IC (μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY): Connect directly to VSS.

VPP (μPD70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY): Connect to VSS in normal operation mode.

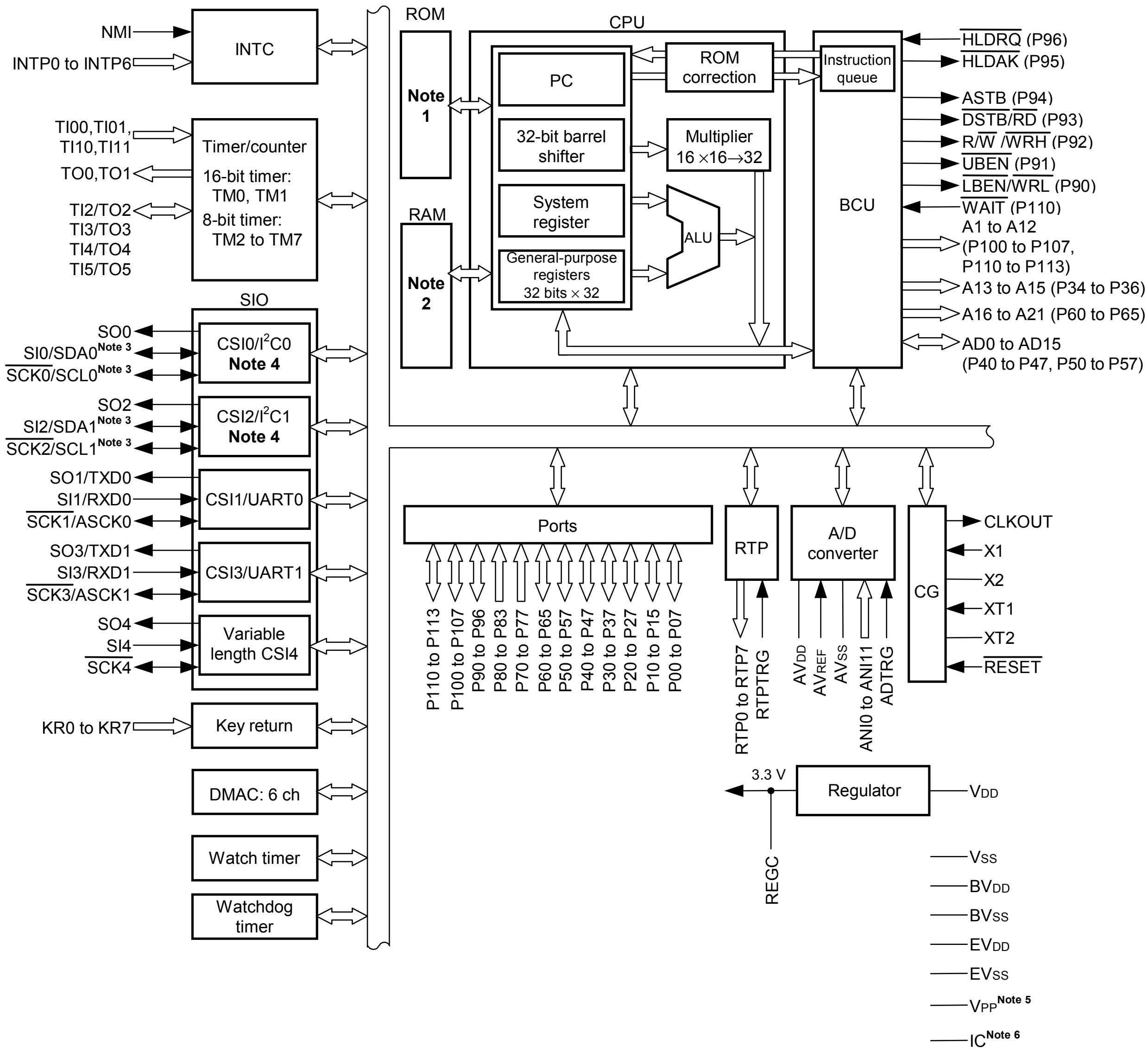
2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3030BY, 70F3032BY, and 70F3033BY.

Pin names (V850/SB1 (B versions))

A1 to A21:	Address bus
AD0 to AD15:	Address/data bus
ADTRG:	A/D trigger input
ANI0 to ANI11:	Analog input
ASCK0, ASCK1:	Asynchronous serial clock
ASTB:	Address strobe
AV_{DD}:	Analog V_{DD}
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog V_{SS}
BV_{DD}:	Power supply for bus interface
BV_{SS}:	Ground for bus interface
CLKOUT:	Clock output
$\overline{\text{DSTB}}$:	Data strobe
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
$\overline{\text{HLDAK}}$:	Hold acknowledge
$\overline{\text{HLDRQ}}$:	Hold request
IC:	Internally connected
INTP0 to INTP6:	Interrupt request from peripherals
KR0 to KR7:	Key return
$\overline{\text{LBEN}}$:	Lower byte enable
NMI:	Non-maskable interrupt request
P00 to P07:	Port 0
P10 to P15:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P65:	Port 6
P70 to P77:	Port 7
P80 to P83:	Port 8
P90 to P96:	Port 9
P100 to P107:	Port 10
P110 to P113:	Port 11
$\overline{\text{RD}}$:	Read
REGC:	Regulator control
$\overline{\text{RESET}}$:	Reset
RTP0 to RTP7:	Real-time output port
RTPTRG:	RTP trigger
R/$\overline{\text{W}}$:	Read/write status
RXD0, RXD1:	Receive data
$\overline{\text{SCK0}}$ to $\overline{\text{SCK4}}$:	Serial clock
SCL0, SCL1:	Serial clock
SDA0, SDA1:	Serial data
SI0 to SI4:	Serial input
SO0 to SO4:	Serial output
TI00, TI01, TI10, TI11, TI2 to TI5:	Timer input
TO0 to TO5:	Timer output
TXD0,TXD1:	Transmit data
$\overline{\text{UBEN}}$:	Upper byte enable
V_{DD}:	Power supply
V_{PP}:	Programming power supply
V_{SS}:	Ground
$\overline{\text{WAIT}}$:	Wait
$\overline{\text{WRH}}$:	Write strobe high level data
$\overline{\text{WRL}}$:	Write strobe low level data
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for subclock

1.3.5 Function blocks (V850/SB1 (B versions))

(1) Internal block diagram



Notes 1. μPD703031B, 703031BY: 128 KB (mask ROM)
μPD703033B, 703033BY: 256 KB (mask ROM)
μPD703030B, 703030BY: 384 K (mask ROM)
μPD703032B, 703032BY: 512 K (mask ROM)
μPD70F3033B, 70F3033BY: 256 K (flash memory)
μPD70F3030B, 70F3030BY: 384 K (flash memory)
μPD70F3032B, 70F3032BY: 512 K (flash memory)

2. μPD703031B, 703031BY: 8 KB
μPD703033B, 703033BY, 70F3033B, 70F3033BY: 16 KB
μPD703030B, 703030BY, 703032B, 703032BY,: 24 KB
70F3030B, 70F3030BY, 70F3032B, 70F3032BY

3. SDA0, SDA1, SCL0, and SCL1 pins are available only in the μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3032BY, 70F3030BY, and 70F3033BY

4. I^{2}C function is available only in the μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3030BY, 70F3032BY, and 70F3033BY

5. μPD70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY

6. μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY

(2) Internal units

(a) 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 $\times$ 16 bits $\rightarrow$ 32 bits) and the barrel shifter (32 bits) help accelerate processing of complex instructions.

(b) 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.

(c) ROM

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

The ROM capacity varies depending on the product. The ROM capacity of each product is shown below.

μPD703031B, 703031BY:	128 KB (mask ROM)
μPD703033B, 703033BY:	256 KB (mask ROM)
μPD70F3033B, 70F3033BY:	256 KB (flash memory)
μPD703030B, 703030BY:	384 KB (mask ROM)
μPD70F3030B, 70F3030BY:	384 KB (flash memory)
μPD703032B, 703032BY:	512 KB (mask ROM)
μPD70F3032B, 70F3032BY:	512 KB (flash memory)

ROM can be accessed by the CPU in one clock cycle during instruction fetch.

(d) RAM

The RAM capacity and mapping addresses vary depending on the product. The RAM capacity of each product is shown below.

μPD703031B, 703031BY:	8 KB (mapping starts at FFFFD000H)
μPD703033B, 703033BY, 70F3033B, 70F3033BY:	16 KB (mapping starts at FFFFB000H)
μPD703030B, 703030BY, 70F3030B, 70F3030BY, 703032B, 703032BY, 70F3032B, 70F3032BY:	24 KB (mapping starts at FFFF9000H)

RAM can be accessed by the CPU in one clock cycle during data access.

(e) 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 multiplexed servicing control can be performed for interrupt sources.

(f) Clock generator (CG)

The clock generator includes two types of oscillators; each for main clock (f_{XX}) and for 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}).

(g) Timer/counter

A two-channel 16-bit timer/event counter, a four-channel 8-bit timer/event counter, and a two-channel 8-bit interval timer 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 to enable use as a 16-bit timer.

The two-channel 8-bit interval timer can be connected via a cascade to enable to be used as a 16-bit timer.

(h) Watch timer

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

(i) Watchdog timer

A watchdog timer is equipped to detect inadvertent 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.

(j) Serial interface (SIO)

The V850/SB1 includes three kinds of serial interfaces: asynchronous serial interfaces (UART0, UART1), clocked serial interfaces (CSI0 to CSI3), and an 8-/16-bit variable-length serial interface (CSI4). These plus the I^2C bus interfaces (I^2C0, I^2C1) comprise five channels. Two of these channels are switchable between the UART and CSI and another two switchable between CSI and I^2C.

For UART0 and UART1, data is transferred via the TXD0, TXD1, RXD0, and RXD1 pins.

For CSI0 to CSI3, data is transferred via the SO0 to SO3, SI0 to SI3, and $\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$ pins.

For CSI4, data is transferred via the SO4, SI4, and $\overline{\text{SCK4}}$ pins.

For I^2C0 and I^2C1, data is transferred via the SDA0, SDA1, SCL0, and SCL1 pins.

I^2C0 and I^2C1 are equipped only in the μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3030BY, 70F3032BY, and 70F3033BY.

For UART and CSI4, a dedicated baud rate generator is equipped.

(k) A/D converter

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

(l) DMA controller

A six-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.

(m) Real-time output port (RTP)

The RTP is a real-time output function that transfers preset 8-bit data to an output latch when an external trigger signal occurs or when there is a match signal in a timer compare register. It can also be used for 4-bit × 2 channels.

(n) 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, serial interface
Port 4	8-bit I/O		External address/data bus
Port 5	8-bit I/O		
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, key return input
Port 11	4-bit I/O		Wait control, external address bus

1.4 V850/SB2 (A Versions)

1.4.1 Features (V850/SB2 (A versions))

○ Number of instructions: 74	
○ Minimum instruction execution time	79 ns (operating at 12.58 MHz, external power supply 5 V, regulator output 3.0 V)
○ General-purpose registers	32 bits × 32 registers
○ Instruction set	Signed multiplication (16 × 16 → 32): 158 ns (operating at 12.58 MHz) (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 3 V to 5 V interface enabled Bus hold function External wait function
○ Internal memory	μPD703034A, 703034AY (mask ROM: 128 KB/RAM: 12 KB) μPD703035A, 703035AY (mask ROM: 256 KB/RAM: 16 KB) μPD703037A, 703037AY (mask ROM: 512 KB/RAM: 24 KB) μPD70F3035A, 70F3035AY (flash memory: 256 KB/RAM: 16 KB) μPD70F3037A, 70F3037AY (flash memory: 512 KB/RAM: 24 KB)
○ Interrupts and exceptions	Non-maskable interrupts: 2 sources Maskable interrupts: 39 sources (μPD703034A, 703035A, 703037A, 70F3035A, 70F3037A) 40 sources (μPD703034AY, 703035AY, 703037AY, 70F3035AY, 70F3037AY) Software exceptions: 32 sources Exception trap: 1 source
○ I/O lines	Total: 83 (12 input ports and 71 I/O ports) 3 V to 5 V interface enabled
○ Timer/counter	16-bit timer: 2 channels (PWM output) 8-bit timer: 6 channels (four PWM outputs, cascade connection enabled)
○ Watch timer	When operating under subclock or main clock: 1 channel Operation using the subclock or main clock is also possible in the IDLE mode.
○ Watchdog timer	1 channel
○ Serial interface (SIO)	Asynchronous serial interface (UART) Clocked serial interface (CSI)

	I^2C bus interface (I^2C)
	(only for μPD703034AY, 703035AY, 703037AY, 70F3035AY, and 70F3037AY)
	8-/16-bit variable-length serial interface
	CSI/UART: 2 channels
	CSI/I^2C: 2 channels
	CSI (8-/16-bit valuable): 1 channel
	Dedicated baud rate generator: 3 channels
○ A/D converter	10-bit resolution: 12 channels
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels
○ Real-time output port (RTP)	8 bits × 1 channel or 4 bits × 2 channels
○ ROM correction	Modifiable 4 points
○ Regulator	4.0 V to 5.5 V input → internal 3.0 V
○ Key return function	4 to 8 selecting enabled, falling edge fixed
○ Clock generator	During main clock or subclock operation
	5-level CPU clock (including sub operations)
○ Power-saving functions	HALT/IDLE/STOP modes
○ IEBus controller	1 ch
○ Package	100-pin plastic LQFP (fine pitch, 14 × 14)
	100-pin plastic QFP (14 × 20)
○ CMOS structure	All static circuits

1.4.2 Application fields (V850/SB2 (A versions))

AV equipment

Example: Audio, car audio equipment, VCR, and TV.

1.4.3 Ordering information (V850/SB2 (A versions))

Part Number	Package	Internal ROM
μPD703034AGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703034AGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703034AYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703034AYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703035AGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703035AGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703035AYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703035AYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703037AGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD703037AYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD70F3035AGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3035AGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3035AYGF-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3035AYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3037AGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)
μPD70F3037AYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)

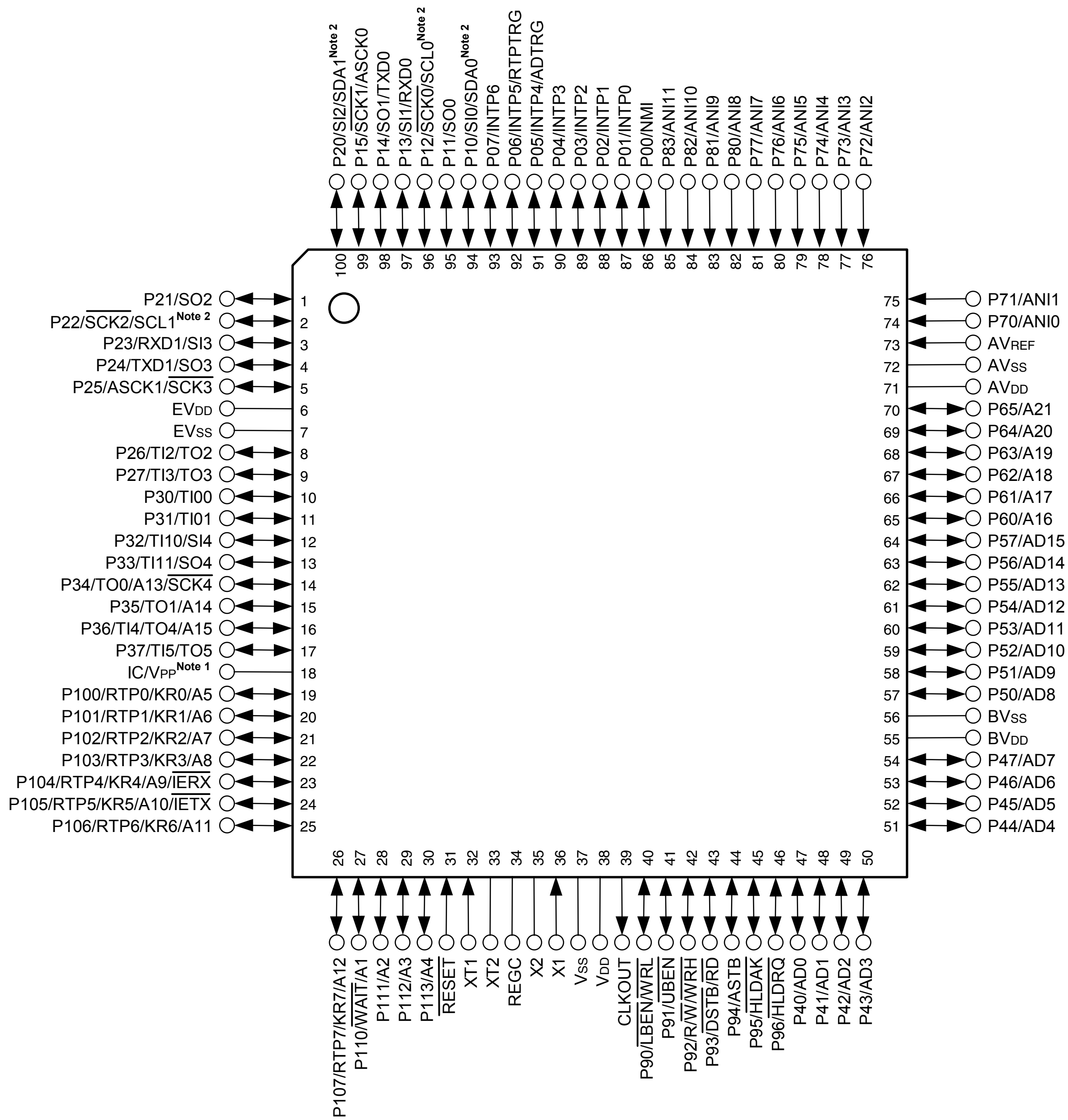
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.4.4 Pin configuration (top view) (V850/SB2 (A versions))

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

- μPD703034AGC-xxx-8EU
- μPD703034AYGC-xxx-8EU
- μPD703035AGC-xxx-8EU
- μPD703035AYGC-xxx-8EU
- μPD70F3035AGC-8EU
- μPD70F3035AYGC-8EU

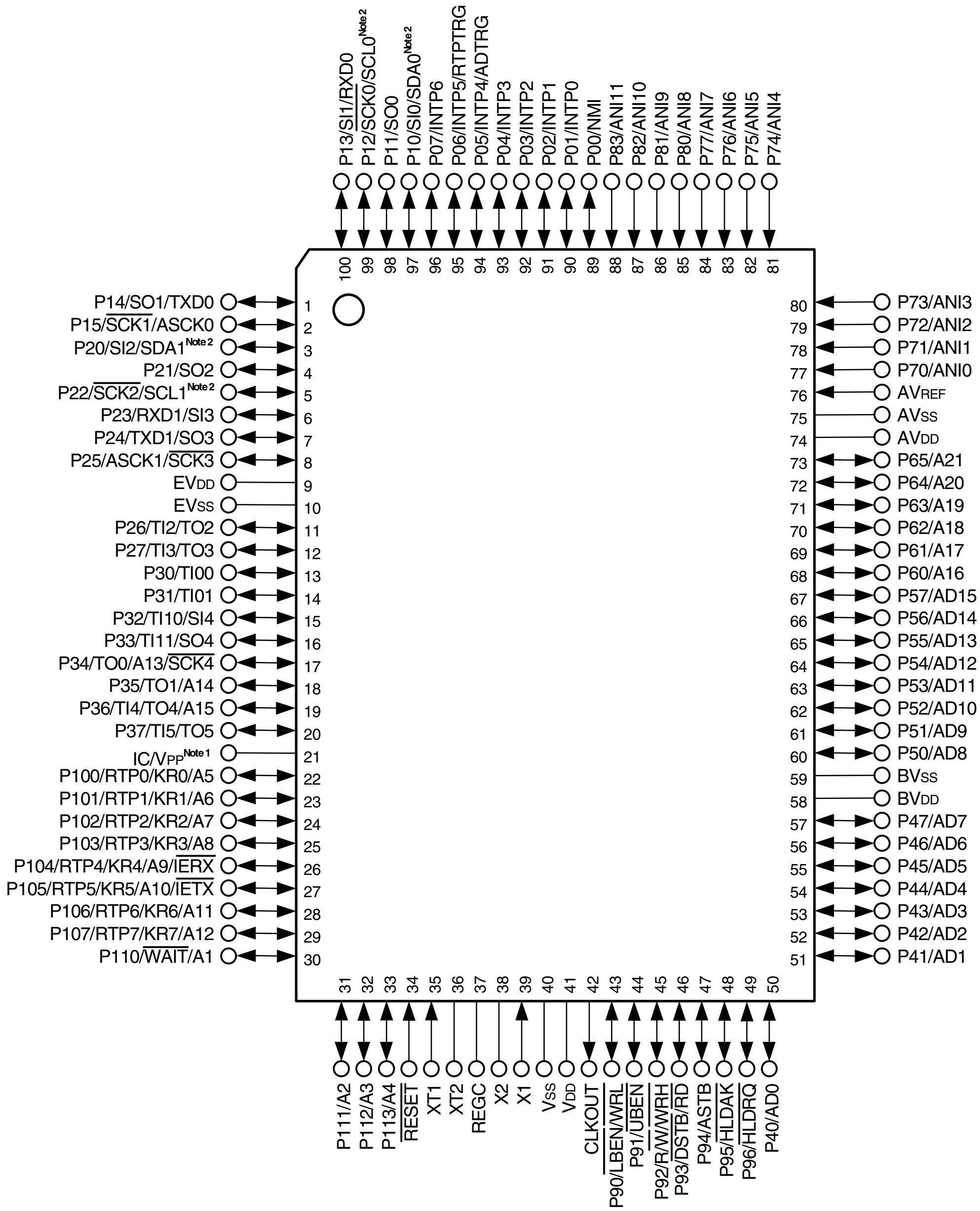


Notes 1. IC (μPD703034A, 703034AY, 703035A, 703035AY): Connect directly to V_{SS}.
V_{PP} (μPD70F3035A, 70F3035AY): Connect to V_{SS} in normal operation mode.

2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703034AY, 703035AY, and 70F3035AY.

100-pin plastic QFP (14 × 20)

- μPD703034AGF-xxx-3BA
- μPD703034AYGF-xxx-3BA
- μPD703035AGF-xxx-3BA
- μPD703035AYGF-xxx-3BA
- μPD703037AGF-xxx-3BA
- μPD703037AYGF-xxx-3BA
- μPD70F3035AGF-3BA
- μPD70F3035AYGF-3BA
- μPD70F3037AGF-3BA
- μPD70F3037AYGF-3BA



Notes 1. IC (μPD703034A, 703034AY, 703035A, 703035AY, 703037A, 703037AY): Connect directly to V_{SS}.
V_{PP} (μPD70F3035A, 70F3035AY, 70F3037A, 70F3037AY):
Connect to V_{SS} in normal operation mode.

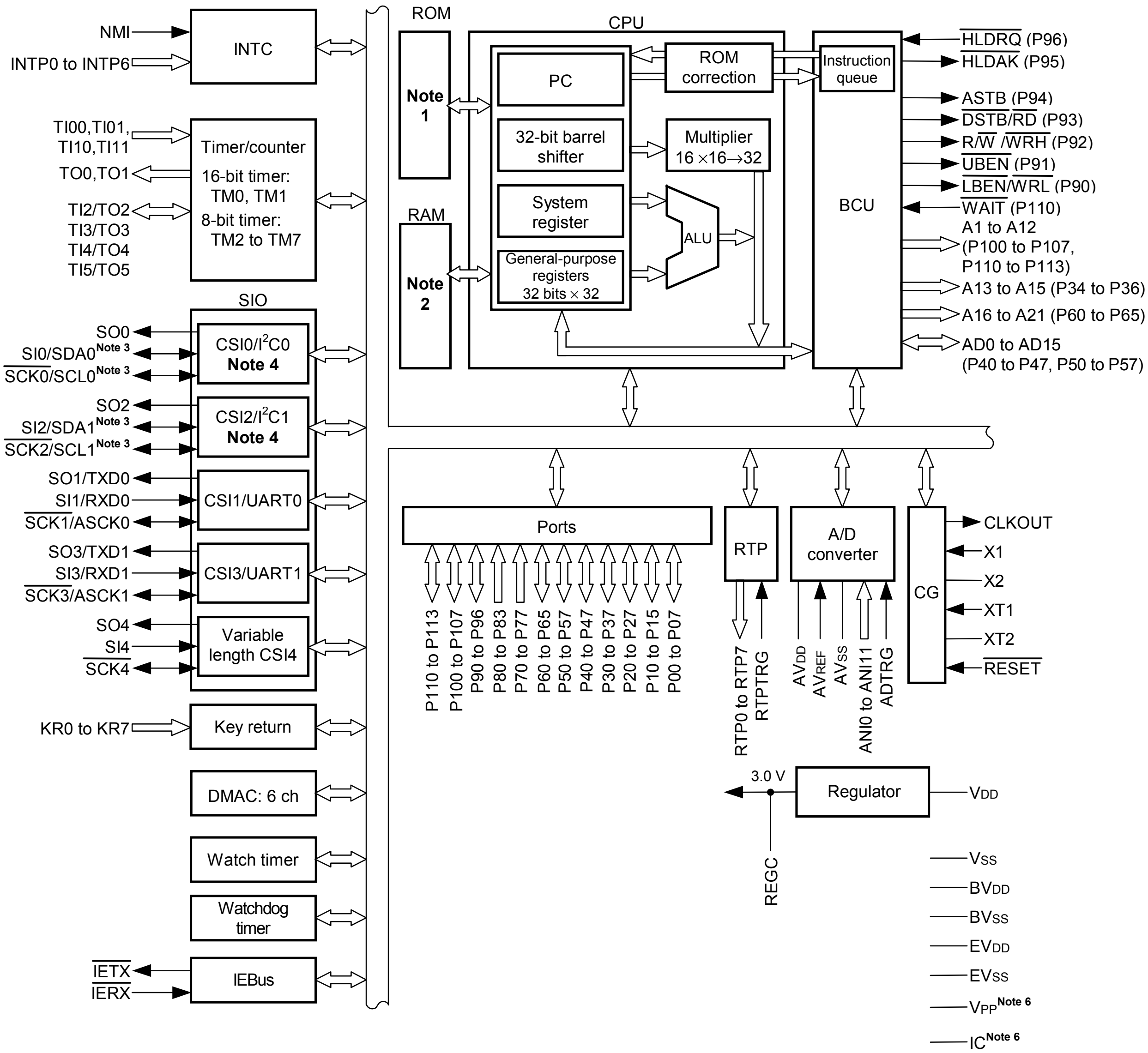
2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703034AY, 703035AY, 703037AY, 70F3035AY, and 70F3037AY.

Pin names (V850/SB2)

A1 to A21:	Address bus
AD0 to AD15:	Address/data bus
ADTRG:	A/D trigger input
ANI0 to ANI11:	Analog input
ASCK0, ASCK1:	Asynchronous serial clock
ASTB:	Address strobe
AV_{DD}:	Analog V_{DD}
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog V_{SS}
BV_{DD}:	Power supply for bus interface
BV_{SS}:	Ground for bus interface
CLKOUT:	Clock output
$\overline{\text{DSTB}}$:	Data strobe
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
$\overline{\text{HLDAK}}$:	Hold acknowledge
$\overline{\text{HLDRQ}}$:	Hold request
IC:	Internally connected
$\overline{\text{IERX}}$:	IEBus receive data
$\overline{\text{IETX}}$:	IEBus transmit data
INTP0 to INTP6:	Interrupt request from peripherals
KR0 to KR7:	Key return
$\overline{\text{LBEN}}$:	Lower byte enable
NMI:	Non-maskable interrupt request
P00 to P07:	Port 0
P10 to P15:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P65:	Port 6
P70 to P77:	Port 7
P80 to P83:	Port 8
P90 to P96:	Port 9
P100 to P107:	Port 10
P110 to P113:	Port 11
$\overline{\text{RD}}$:	Read
REGC:	Regulator control
$\overline{\text{RESET}}$:	Reset
RTP0 to RTP7:	Real-time output port
RTPTRG:	RTP trigger
R/$\overline{\text{W}}$:	Read/write status
RXD0, RXD1:	Receive data
$\overline{\text{SCK0}}$ to $\overline{\text{SCK4}}$:	Serial clock
SCL0, SCL1:	Serial clock
SDA0, SDA1:	Serial data
SI0 to SI4:	Serial input
SO0 to SO4:	Serial output
TI00, TI01, TI10, TI11, TI2 to TI5:	Timer input
TO0 to TO5:	Timer output
TXD0,TXD1:	Transmit data
$\overline{\text{UBEN}}$:	Upper byte enable
V_{DD}:	Power supply
V_{PP}:	Programming power supply
V_{SS}:	Ground
$\overline{\text{WAIT}}$:	Wait
$\overline{\text{WRH}}$:	Write strobe high level data
$\overline{\text{WRL}}$:	Write strobe low level data
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for subclock

1.4.5 Function blocks (V850/SB2 (A versions))

(1) Internal block diagram



Notes 1. μPD703034A, 703034AY: 128 KB (mask ROM)
μPD703035A, 703035AY: 256 KB (mask ROM)
μPD703037A, 703037AY: 512 KB (mask ROM)
μPD70F3035A, 70F3035AY: 256 KB (flash memory)
μPD70F3037A, 70F3037AY: 512 KB (flash memory)

2. μPD703034A, 703034AY: 12 KB
μPD703035A, 703035AY, 70F3035A, 70F3035AY: 16 KB
μPD703036A, 703036AY: 20 KB
μPD703037A, 703037AY, 70F3037A, 70F3037AY: 24 KB

3. SDA0, SDA1, SCL0, and SCL1 pins are available only in the μPD703034AY, 703035AY, 703037AY, 70F3035AY, and 70F3037AY

4. I^{2}C function is available only in the μPD703034AY, 703035AY, 703037AY, 70F3035AY, and 70F3037AY

5. μ PD70F3035A, 70F3035AY, 70F3037A, 70F3037AY

6. μ PD703034A, 703034AY, 703035A, 703035AY, 703037A, 703037AY

(2) Internal units

(a) 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 $\times$ 16 bits $\rightarrow$ 32 bits) and the barrel shifter (32 bits) help accelerate processing of complex instructions.

(b) 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.

(c) ROM

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

The ROM capacity varies depending on the product. The ROM capacity of each product is shown below.

μPD703034A, 703034AY:	128 KB (mask ROM)
μPD703035A, 703035AY:	256 KB (mask ROM)
μPD70F3035A, 70F3035AY:	256 KB (flash memory)
μPD703037A, 703037AY:	512 KB (mask ROM)
μPD70F3037A, 70F3037AY:	512 KB (flash memory)

ROM can be accessed by the CPU in one clock cycle during instruction fetch.

(d) RAM

The RAM capacity and mapping addresses vary depending on the product. The RAM capacity of each product is shown below.

μPD703034A, 703034AY:	12 KB (mapping starts at FFFFC000H)
μPD703035A, 703035AY, 70F3035A, 70F3035AY:	16 KB (mapping starts at FFFFB000H)
μPD703037A, 703037AY, 70F3037A, 70F3037AY:	24 KB (mapping starts at FFFF9000H)

RAM can be accessed by the CPU in one clock cycle during data access.

(e) 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 multiplexed servicing control can be performed for interrupt sources.

(f) Clock generator (CG)

The clock generator includes two types of oscillators; each for main clock (f_{XX}) and for 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}).

(g) Timer/counter

A two-channel 16-bit timer/event counter, a four-channel 8-bit timer/event counter, and a two-channel 8-bit interval timer 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 to enable use as a 16-bit timer.

The two-channel 8-bit interval timer can be connected via a cascade to enable to be used as a 16-bit timer.

(h) Watch timer

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

(i) Watchdog timer

A watchdog timer is equipped to detect inadvertent 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.

(j) Serial interface (SIO)

The V850/SB2 includes three kinds of serial interfaces: asynchronous serial interfaces (UART0, UART1), clocked serial interfaces (CSI0 to CSI3), and an 8-/16-bit variable-length serial interface (CSI4). These plus the I^2C bus interfaces (I^2C0, I^2C1) comprise five channels. Two of these channels are switchable between the UART and CSI and another two switchable between CSI and I^2C.

For UART0 and UART1, data is transferred via the TXD0, TXD1, RXD0, and RXD1 pins.

For CSI0 to CSI3, data is transferred via the SO0 to SO3, SI0 to SI3, and $\overline{SCK0}$ to $\overline{SCK3}$ pins.

For CSI4, data is transferred via the SO4, SI4, and $\overline{SCK4}$ pins.

For I^2C0 and I^2C1, data is transferred via the SDA0, SDA1, SCL0, and SCL1 pins.

I^2C0 and I^2C1 are equipped only in the μPD703034AY, 703035AY, 703037AY, 70F3035AY, and 70F3037AY.

For UART and CSI4, a dedicated baud rate generator is equipped.

(k) A/D converter

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

(l) DMA controller

A six-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.

(m) Real-time output port (RTP)

The RTP is a real-time output function that transfers preset 8-bit data to an output latch when an external trigger signal occurs or when there is a match signal in a timer compare register. It can also be used for 4-bit × 2 channels.

(n) 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, serial interface
Port 4	8-bit I/O		External address/data bus
Port 5	8-bit I/O		
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, key return input, IEBus data I/O
Port 11	4-bit I/O		Wait control, external address bus

(o) IEBus controller

The IEBus controller is a small-scale digital data transfer system aiming at data transfer among units.
The IEBus controller is incorporated only in the V850/SB2.

★ ## 1.5 V850/SB2 (B and H Versions)

1.5.1 Features (V850/SB2 (B and H versions))

○ Number of instructions: 74	
○ Minimum instruction execution time	B versions: 79 ns (@ 12.58 MHz operation, external power supply 5 V, regulator output 3.0 V operation) H versions: 53 ns (@ 18.87 MHz operation, external power supply 5 V, regulator output 3.3 V operation)
○ General-purpose registers	32 bits × 32 registers
○ Instruction set	Signed multiplication (16 × 16 → 32): (able to execute instructions in parallel continuously without creating any register hazards). B versions: 158 ns (@ 12.58 MHz operation) H versions: 106 ns (@ 18.87 MHz operation) 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 3 V to 5 V interface enabled Bus hold function External wait function
○ Internal memory	μPD703034B, 703034BY (mask ROM: 128 KB/RAM: 8 KB) μPD703035B, 703035BY (mask ROM: 256 KB/RAM: 16 KB) μPD703036H, 703036HY (mask ROM: 384 KB/RAM: 24 KB) μPD703037H, 703037HY (mask ROM: 512 KB/RAM: 24 KB) μPD70F3035B, 70F3035BY (flash memory: 256 KB/RAM: 16 KB) μPD70F3036H, 70F3036HY (flash memory: 384 KB/RAM: 24 KB) μPD70F3037H, 70F3037HY (flash memory: 512 KB/RAM: 24 KB)
○ Interrupts and exceptions	Non-maskable interrupts: 2 sources Maskable interrupts: 39 sources (μPD703034B, 703035B, 703036H, 703037H, 70F3035B, 70F3036H, 70F3037H) 40 sources (μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, 70F3037HY) Software exceptions: 32 sources Exception trap: 1 source
○ I/O lines	Total: 83 (12 input ports and 71 I/O ports) 3 V to 5 V interface enabled
○ Timer/counter	16-bit timer: 2 channels (PWM output) 8-bit timer: 6 channels (four PWM outputs, cascade connection enabled)

○ Watch timer	When operating under subclock or main clock: 1 channel Operation using the subclock or main clock is also possible in the IDLE mode.
○ Watchdog timer	1 channel
○ Serial interface (SIO)	Asynchronous serial interface (UART) Clocked serial interface (CSI) I^2C bus interface (I^2C) (only for μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, and 70F3037HY) 8-/16-bit variable-length serial interface CSI/UART: 2 channels CSI/I^2C: 2 channels CSI (8-/16-bit valuable): 1 channel Dedicated baud rate generator: 3 channels
○ A/D converter	10-bit resolution: 12 channels
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels
○ Real-time output port (RTP)	8 bits × 1 channel or 4 bits × 2 channels
○ ROM correction	Modifiable 4 points
○ Regulator	4.0 V to 5.5 V input → internal 3.0 V
○ Key return function	4 to 8 selecting enabled, falling edge fixed
○ Clock generator	During main clock or subclock operation 5-level CPU clock (including sub operations)
○ Power-saving functions	HALT/IDLE/STOP modes
○ IEBus controller	1 ch
○ Package	100-pin plastic LQFP (fine pitch, 14 × 14) 100-pin plastic QFP (14 × 20)
○ CMOS structure	All static circuits

1.5.2 Application fields (V850/SB2 (B and H versions))

AV equipment

Example: Audio, car audio equipment, VCR, and TV.

1.5.3 Ordering information (V850/SB2 (B and H versions))

Part Number	Package	Internal ROM
μPD703034BGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703034BGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703034BYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (128 KB)
μPD703034BYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (128 KB)
μPD703035BGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703035BGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703035BYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (256 KB)
μPD703035BYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (256 KB)
μPD703036HGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (384 KB)
μPD703036HGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (384 KB)
μPD703036HYGC-xxx-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (384 KB)
μPD703036HYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (384 KB)
μPD703037HGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD703037HYGF-xxx-3BA	100-pin plastic QFP (14 × 20)	Mask ROM (512 KB)
μPD70F3035BGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3035BGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3035BYGF-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3035BYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (256 KB)
μPD70F3036HGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (384 KB)
μPD70F3036HGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (384 KB)
μPD70F3036HYGC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (384 KB)
μPD70F3036HYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (384 KB)
μPD70F3037HGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)
μPD70F3037HYGF-3BA	100-pin plastic QFP (14 × 20)	Flash memory (512 KB)

Note In planning

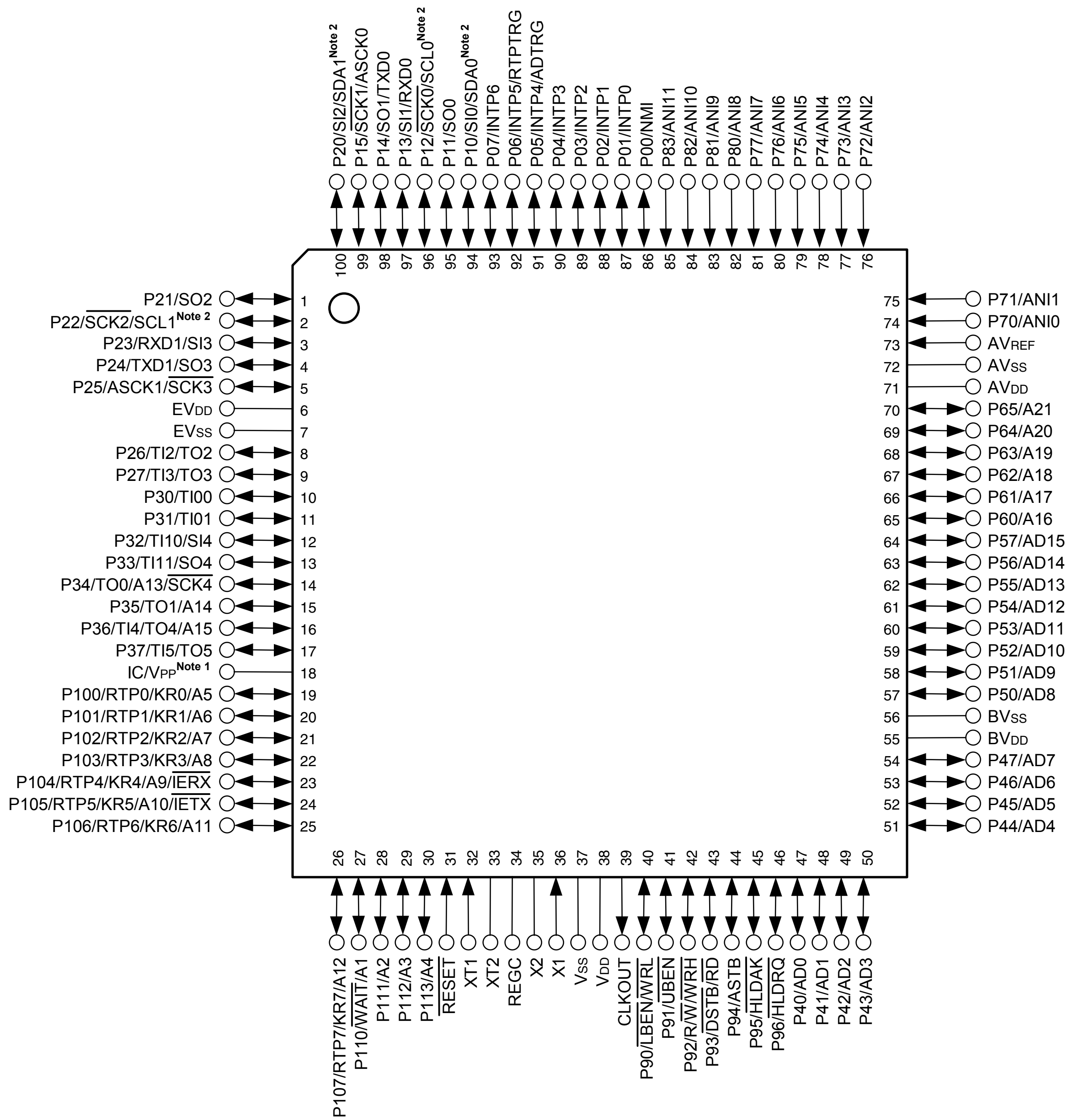
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.5.4 Pin configuration (top view) (V850/SB2 (B and H versions))

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

- μPD703034BGC-xxx-8EU
- μPD703034BYGC-xxx-8EU
- μPD703035BGC-xxx-8EU
- μPD703035BYGC-xxx-8EU
- μPD703036HGC-xxx-8EU
- μPD703036HYGC-xxx-8EU
- μPD70F3035BGC-8EU
- μPD70F3035BYGC-8EU
- μPD70F3036HGC-8EU
- μPD70F3036HYGC-8EU

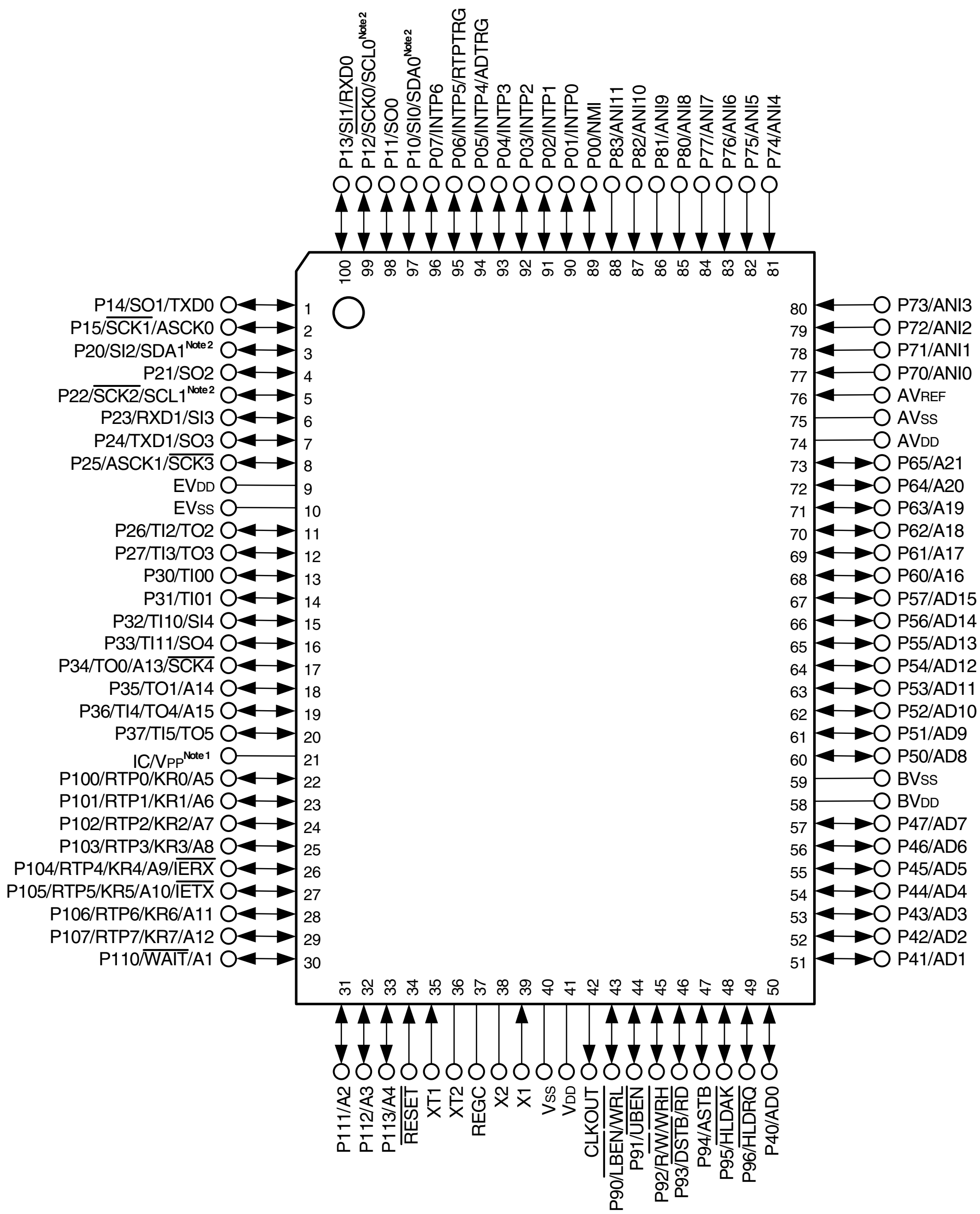


Notes 1. IC (μPD703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY): Connect directly to V_{SS}.
V_{PP} (μPD70F3035B, 70F3035BY, 70F3036H, 70F3036HY): Connect to V_{SS} in normal operation mode.

2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703034BY, 703035BY, 703036HY, 70F3035BY, and 70F3036HY.

100-pin plastic QFP (14 × 20)

- μPD703034BGF-xxx-3BA
- μPD703034BYGF-xxx-3BA
- μPD703035BGF-xxx-3BA
- μPD703035BYGF-xxx-3BA
- μPD703036HGF-xxx-3BA
- μPD703036HYGF-xxx-3BA
- μPD703037HGF-xxx-3BA
- μPD703037HYGF-xxx-3BA
- μPD70F3035BGF-3BA
- μPD70F3035BYGF-3BA
- μPD70F3036HGF-xxx-3BA
- μPD70F3036HYGF-xxx-3BA
- μPD70F3037HGF-3BA
- μPD70F3037HYGF-3BA



Notes 1. IC (μPD703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY, 703037H, 703037HY): Connect directly to V_{SS}.

V_{PP} (μPD70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY): Connect to V_{SS} in normal operation mode.

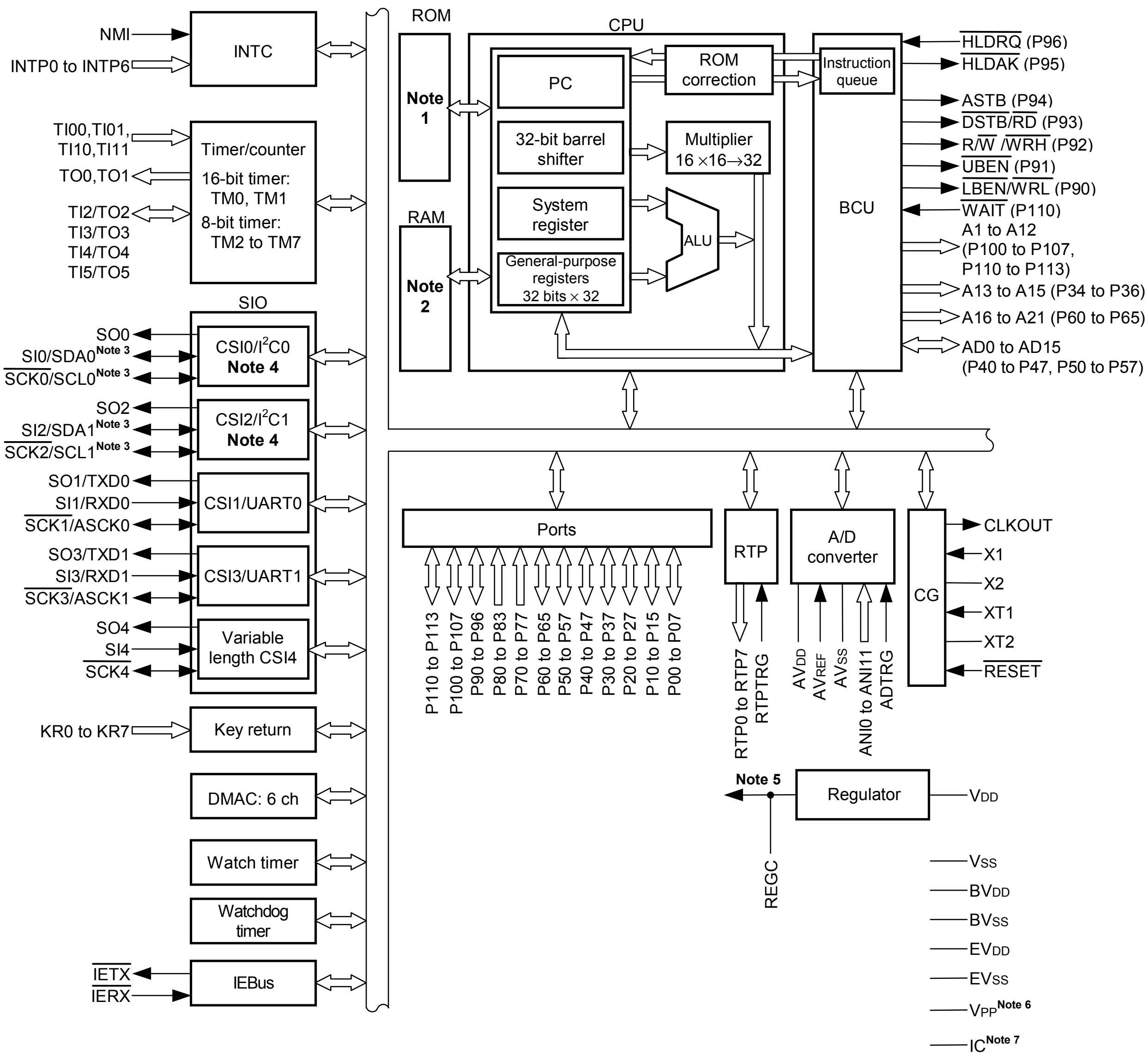
2. SCL0, SCL1, SDA0, and SDA1 are available only in the μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, and 70F3037HY.

Pin names (V850/SB2 (B and H versions))

A1 to A21:	Address bus
AD0 to AD15:	Address/data bus
ADTRG:	A/D trigger input
ANI0 to ANI11:	Analog input
ASCK0, ASCK1:	Asynchronous serial clock
ASTB:	Address strobe
AV_{DD}:	Analog V_{DD}
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog V_{SS}
BV_{DD}:	Power supply for bus interface
BV_{SS}:	Ground for bus interface
CLKOUT:	Clock output
$\overline{\text{DSTB}}$:	Data strobe
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
$\overline{\text{HLDAK}}$:	Hold acknowledge
$\overline{\text{HLDRQ}}$:	Hold request
IC:	Internally connected
$\overline{\text{IERX}}$:	IEBus receive data
$\overline{\text{IETX}}$:	IEBus transmit data
INTP0 to INTP6:	Interrupt request from peripherals
KR0 to KR7:	Key return
$\overline{\text{LBEN}}$:	Lower byte enable
NMI:	Non-maskable interrupt request
P00 to P07:	Port 0
P10 to P15:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P65:	Port 6
P70 to P77:	Port 7
P80 to P83:	Port 8
P90 to P96:	Port 9
P100 to P107:	Port 10
P110 to P113:	Port 11
$\overline{\text{RD}}$:	Read
REGC:	Regulator control
$\overline{\text{RESET}}$:	Reset
RTP0 to RTP7:	Real-time output port
RTPTRG:	RTP trigger
R/$\overline{\text{W}}$:	Read/write status
RXD0, RXD1:	Receive data
$\overline{\text{SCK0}}$ to $\overline{\text{SCK4}}$:	Serial clock
SCL0, SCL1:	Serial clock
SDA0, SDA1:	Serial data
SI0 to SI4:	Serial input
SO0 to SO4:	Serial output
TI00, TI01, TI10, TI11, TI2 to TI5:	Timer input
TO0 to TO5:	Timer output
TXD0,TXD1:	Transmit data
$\overline{\text{UBEN}}$:	Upper byte enable
V_{DD}:	Power supply
V_{PP}:	Programming power supply
V_{SS}:	Ground
$\overline{\text{WAIT}}$:	Wait
$\overline{\text{WRH}}$:	Write strobe high level data
$\overline{\text{WRL}}$:	Write strobe low level data
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for subclock

1.5.5 Function blocks (V850/SB2 (B and H versions))

(1) Internal block diagram



Notes 1. μPD703034B, 703034BY: 128 K (mask ROM)
μPD703035B, 703035BY: 256 K (mask ROM)
μPD703036H, 703036HY: 384 K (mask ROM)
μPD703037H, 703037HY: 512 K (mask ROM)
μPD70F3035B, 70F3035BY: 256 K (flash memory)
μPD70F3036H, 70F3036HY: 384 K (flash memory)
μPD70F3037H, 70F3037HY: 512 K (flash memory)

2. μPD703034B, 703034BY 8 KB
μPD703035B, 703035BY, 70F3035B, 70F3035BY: 16 KB
μPD703036H, 703036HY, 703037H, 703037HY,: 24 KB
70F3036H, 70F3036HY, 70F3037H, 70F3037HY

3. SDA0, SDA1, SCL0, SCL1 pins are available only in the μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, and 70F3037HY

4. I[2]C functions is available only in the μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, and 70F3037HY

5. B versions: 3.0 V, H versions: 3.3 V

6. μPD70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY

7. μPD703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY, 703037H, 703037HY

(2) Internal units

(a) 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 $\times$ 16 bits $\rightarrow$ 32 bits) and the barrel shifter (32 bits) help accelerate processing of complex instructions.

(b) 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.

(c) ROM

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

The ROM capacity varies depending on the product. The ROM capacity of each product is shown below.

μPD703034B, 703034BY:	128 KB (mask ROM)
μPD703035B, 703035BY:	256 KB (mask ROM)
μPD70F3035B, 70F3035BY:	256 KB (flash memory)
μPD703036H, 703036HY:	384 KB (mask ROM)
μPD70F3036H, 70F3036HY:	384 KB (flash memory)
μPD703037H, 703037HY:	512 KB (mask ROM)
μPD70F3037H, 70F3037HY:	512 KB (flash memory)

ROM can be accessed by the CPU in one clock cycle during instruction fetch.

(d) RAM

The RAM capacity and mapping addresses vary depending on the product. The RAM capacity of each product is shown below.

μPD703034B, 703034BY:	8 KB (mapping starts at FFFFD000H)
μPD703035B, 703035BY, 70F3035B, 70F3035BY:	16 KB (mapping starts at FFFFB000H)
μPD703036H, 703036HY, 70F3036H, 70F3036HY, 703037H, 703037HY, 70F3037H, 70F3037HY:	24 KB (mapping starts at FFFF9000H)

RAM can be accessed by the CPU in one clock cycle during data access.

(e) 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 multiplexed servicing control can be performed for interrupt sources.

(f) Clock generator (CG)

The clock generator includes two types of oscillators; each for main clock (f_{XX}) and for 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}).

(g) Timer/counter

A two-channel 16-bit timer/event counter, a four-channel 8-bit timer/event counter, and a two-channel 8-bit interval timer 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 to enable use as a 16-bit timer.

The two-channel 8-bit interval timer can be connected via a cascade to enable to be used as a 16-bit timer.

(h) Watch timer

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

(i) Watchdog timer

A watchdog timer is equipped to detect inadvertent 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.

(j) Serial interface (SIO)

The V850/SB2 includes three kinds of serial interfaces: asynchronous serial interfaces (UART0, UART1), clocked serial interfaces (CSI0 to CSI3), and an 8-/16-bit variable-length serial interface (CSI4). These plus the I^2C bus interfaces (I^2C0, I^2C1) comprise five channels. Two of these channels are switchable between the UART and CSI and another two switchable between CSI and I^2C.

For UART0 and UART1, data is transferred via the TXD0, TXD1, RXD0, and RXD1 pins.

For CSI0 to CSI3, data is transferred via the SO0 to SO3, SI0 to SI3, and $\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$ pins.

For CSI4, data is transferred via the SO4, SI4, and $\overline{\text{SCK4}}$ pins.

For I^2C0 and I^2C1, data is transferred via the SDA0, SDA1, SCL0, and SCL1 pins.

I^2C0 and I^2C1 are equipped only in the μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, and 70F3037HY.

For UART and CSI4, a dedicated baud rate generator is equipped.

(k) A/D converter

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

(l) DMA controller

A six-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.

(m) Real-time output port (RTP)

The RTP is a real-time output function that transfers preset 8-bit data to an output latch when an external trigger signal occurs or when there is a match signal in a timer compare register. It can also be used for 4-bit × 2 channels.

(n) 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, serial interface
Port 4	8-bit I/O		External address/data bus
Port 5	8-bit I/O		
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, key return input, IEBus data I/O
Port 11	4-bit I/O		Wait control, external address bus

(o) IEBus controller

The IEBus controller is a small-scale digital data transfer system aiming at data transfer among units. The IEBus controller is incorporated only in the V850/SB2.

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The names and functions of the pins of the V850/SB1 and V850/SB2 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 EV_{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	When using A/D converter: $4.5\ V \leq AV_{DD} \leq 5.5\ V$ When not using A/D converter: $3.5\ V \leq AV_{DD} \leq 5.5\ V$
BV_{DD}	Port 4, port 5, port 6, port 9, CLKOUT	$3.0\ V \leq BV_{DD} \leq 5.5\ V$
EV_{DD}	Port 0, port 1, port 2, port 3, port 10, port 11, $\overline{\text{RESET}}$	$3.0\ V \leq EV_{DD} \leq 5.5\ V$

Caution The electrical specifications in the case of 3.0 V to up to 4.0 V are different from those for 4.0 to 5.5 V.

Differences in pins between the V850/SB1 and V850/SB2 are shown below.

★ **Table 2-2. Differences in Pins Between V850/SB1 and V850/SB2**

Pin	V850/SB1				V850/SB2			
	μPD703031A, μPD703032A, μPD703033A, μPD703030B, μPD703031B, μPD703032B, μPD703033B	μPD70F3032A, μPD70F3033A, μPD70F3030B, μPD70F3032B, μPD70F3033B	μPD703031AY, μPD703032AY, μPD703033AY, μPD703030BY, μPD703031BY, μPD703032BY, μPD703033BY	μPD70F3032AY, μPD70F3033AY, μPD70F3030BY, μPD70F3032BY, μPD70F3033BY	μPD703034A, μPD703035A, μPD703037A, μPD703034B, μPD703035B, μPD703036H, μPD703037H	μPD70F3035A, μPD70F3037A, μPD70F3035B, μPD70F3036H, μPD70F3037H	μPD703034AY, μPD703035AY, μPD703037AY, μPD703034BY, μPD703035BY, μPD703036HY, μPD703037HY	μPD70F3035AY, μPD70F3037AY, μPD70F3035BY, μPD70F3036HY, μPD70F3037HY
IC	Available	None	Available	None	Available	None	Available	None
V_{PP}	None	Available	None	Available	None	Available	None	Available
SDA0, SDA1	None		Available		None		Available	
SCL0, SCL1	None		Available		None		Available	
$\overline{\text{IERX}}$	None				Available			
$\overline{\text{IETX}}$	None				Available			

(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/SDA0
P11				SO0
P12				$\overline{\text{SCK0}}$/SCL0
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/SDA1
P21				SO2
P22				$\overline{\text{SCK2}}$/SCL1
P23				SI3/RXD1
P24				SO3/TXD1
P25				$\overline{\text{SCK3}}$/ASCK1
P26				TI2/TO2
P27				TI3/TO3

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/SI4
P33				TI11/SO4
P34				TO0/A13/$\overline{\text{SCK4}}$
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/KR0
P101				RTP1/A6/KR1
P102				RTP2/A7/KR2
P103				RTP3/A8/KR3
P104				RTP4/A9/KR4/$\overline{\text{IERX}}$
P105				RTP5/A10/KR5/$\overline{\text{IETX}}$
P106				RTP6/A11/KR6
P107				RTP7/A12/KR7
P110	I/O	Yes	Port 11 4-bit I/O port Input/output mode can be specified in 1-bit units.	A1/$\overline{\text{WAIT}}$
P111				A2
P112				A3
P113				A4

Remark PULL: On-chip pull-up resistor

(2) Non-port pins

(1/3)

Pin Name	I/O	PULL	Function	Alternate Function
A1	Output	Yes	Lower address bus used for external memory expansion	P110/$\overline{\text{WAIT}}$
A2 to A4				P111 to P113
A5 to A8				P100/RTP0/KR0 to P103/RTP3/KR3
A9				P104/RTP4/KR4/$\overline{\text{IERX}}$
A10				P105/RTP5/KR5/$\overline{\text{IETX}}$
A11, A12				P106/RTP6/KR6 to P107/RTP7/KR7
A13				P34/TO0/$\overline{\text{SCK4}}$
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 clock input for UART0 and UART1	P15/$\overline{\text{SCK1}}$
ASCK1				P25/$\overline{\text{SCK3}}$
ASTB	Output	No	External address strobe signal output	P94
AV_{DD}	–	–	Positive power supply for A/D converter and alternate-function port	–
AV_{REF}	Input	–	Reference voltage input for A/D converter	–
AV_{SS}	–	–	Ground potential for A/D converter and alternate-function port	–
BV_{DD}	–	–	Positive power supply for bus interface and alternate-function port	–
BV_{SS}	–	–	Ground potential for bus interface and alternate-function port	–
CLKOUT	Output	–	Internal system clock output	–
$\overline{\text{DSTB}}$	Output	No	External data strobe signal output	P93/$\overline{\text{RD}}$
EV_{DD}	–	–	Power supply for I/O port and alternate-function pin (except for bus interface)	–
EV_{SS}	–	–	Ground potential for I/O port and alternate-function pin (except for bus interface)	–
$\overline{\text{HLDAK}}$	Output	No	Bus hold acknowledge output	P95
$\overline{\text{HLDRQ}}$	Input	No	Bus hold request input	P96
$\overline{\text{IERX}}$	Input	Yes	IEBus data input (V850/SB2 only)	P104/RTP4/KR4/A9
$\overline{\text{IETX}}$	Output		IEBus data output (V850/SB2 only)	P105/RTP5/KR5/A10
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

Remark PULL: On-chip pull-up resistor

(2/3)

Pin Name	I/O	PULL	Function	Alternate Function
INTP6	Input	Yes	External interrupt request input (digital noise elimination for remote control)	P07
KR0 to KR3			Key return input	P100/A5/RTP0 to P103/A8/RTP3
KR4				P104/A9/RTP4/$\overline{\text{IERX}}$
KR5				P105/A10/RTP5/$\overline{\text{IETX}}$
KR6, KR7				P106/A11/RTP6 to P107/A12/RTP7
$\overline{\text{LBEN}}$	Output	No	External data bus's lower byte enable signal output	P90/$\overline{\text{WRL}}$
IC	–	–	Internally connected (mask ROM versions only)	–
NMI	Input	Yes	Non-maskable interrupt request input (analog noise elimination)	P00
$\overline{\text{RD}}$	Output	No	Read strobe signal output	P93/$\overline{\text{DSTB}}$
REGC	–	–	Capacitor connection for regulator output stabilization	–
$\overline{\text{RESET}}$	Input	–	System reset input	–
RTP0 to RTP3	Output	Yes	Real-time output port	P100/A5/KR0 to P103/A8/KR3
RTP4				P104/A9/KR4/$\overline{\text{IERX}}$
RTP5				P105/A10/KR5/$\overline{\text{IETX}}$
RTP6, RTP7				P106/A11/KR6, P107/A12/KR7
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/SI3
$\overline{\text{SCK0}}$	I/O	Yes	Serial clock I/O (3-wire type) for CSI0 to CSI3	P12/SCL0
$\overline{\text{SCK1}}$				P15/ASCK0
$\overline{\text{SCK2}}$				P22/SCL1
$\overline{\text{SCK3}}$				P25/ASCK1
$\overline{\text{SCK4}}$			Serial clock I/O for variable-length CSI4 (3-wire type)	P34/TO0/A13
SCL0	I/O	Yes	Serial clock I/O for I^{2}C0 and I^{2}C1 (Y versions (products with on-chip I^{2}C) only)	P12/$\overline{\text{SCK0}}$
SCL1				P22/$\overline{\text{SCK2}}$
SDA0	I/O	Yes	Serial transmit/receive data I/O for I^{2}C0 and I^{2}C1 (Y versions (products with on-chip I^{2}C) only)	P10/SI0
SDA1				P20/SI2
SI0	Input	Yes	Serial receive data input (3-wire type) for CSI0 to CSI3	P10/SDA0
SI1				P13/RXD0
SI2				P20/SDA1
SI3				P23/RXD1
SI4			Serial receive data input (3-wire type) for variable-length CSI4	P32/TI10
SO0	Output	Yes	Serial transmit data output (3-wire type) for CSI0 to CSI3	P11

Remark PULL: On-chip pull-up resistor

(3/3)

Pin Name	I/O	PULL	Function	Alternate Function
SO1	Output	Yes	Serial transmit data output (3-wire type) for CSI0 to CSI3	P14/TXD0
SO2				P21
SO3				P24/TXD1
SO4			Serial transmit data output for variable-length CSI4 (3-wire type)	P33/TI11
TI00	Input	Yes	Shared as external capture trigger input and external count clock input for TM0	P30
TI01			External capture trigger input for TM0	P31
TI10			Shared as external capture trigger input and external count clock input for TM1	P32/SI4
TI11			External capture trigger input for TM1	P33/SO4
TI2			External count clock input for TM2	P26/TO2
TI3			External count clock input for TM3	P27/TO3
TI4	Input	Yes	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/$\overline{\text{SCK4}}$/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/SO3
$\overline{\text{UBEN}}$	Output	No	Higher byte enable signal output for external data bus	P91
V_{DD}	–	–	Positive power supply pin	–
V_{PP}	–	–	High-voltage apply pin for program write/verify (flash memory versions only)	–
V_{SS}	–	–	GND potential	–
$\overline{\text{WAIT}}$	Input	Yes	Control signal input for inserting wait in bus cycle	P110/A1
$\overline{\text{WRH}}$	Output	No	Higher byte write strobe signal output for external data bus	P92/R/$\overline{\text{W}}$
$\overline{\text{WRL}}$			Lower byte write strobe signal output for external data bus	P90/$\overline{\text{LBEN}}$
X1	Input	No	Resonator connection for main clock	–
X2	–			–
XT1	Input	No	Resonator connection for subclock	–
XT2	–			–

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 2-3. Operating States of Pins in Each Operating Mode**

Pin \ Operating State	Reset[Note 1]	HALT Mode/ Idle State	IDLE Mode/ 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 preset 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 (A/D trigger input) ··· input

This is the A/D converter's external trigger input pin. This pin is controlled by A/D converter mode register 1 (ADM1).

(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) SDA0 (Serial data 0) ··· I/O

This is the serial transmit/receive data I/O pin of I^2C0 (Y versions (products with on-chip I^2C) only).

(v) SCL0 (Serial clock 0) ··· I/O

This is the serial clock I/O pin for I^2C0 (Y versions (products with on-chip I^2C) 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.

P20 to P22, P24, and P25 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, SI3 (Serial input 2, 3) ··· input

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

(ii) SO2, SO3 (Serial output 2, 3) ··· output

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

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

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

(iv) SDA1 (Serial data 1) ... I/O

This is the serial transmit/receive data I/O pin of I^{2}C1 (Y versions (products with on-chip I^{2}C) only).

(v) SCL1 (Serial clock) ... I/O

This is the serial clock I/O pin of I^{2}C1 (Y versions (products with I^{2}C) only).

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

This is the serial receive data input pin of UART1.

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

This is the serial transmit data output pin of UART1.

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

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

(ix) TI2, TI3 (Timer input 2, 3) ... input

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

(x) TO2, TO3 (Timer output 2, 3) ... output

These are the pulse signal output pins of 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, an address bus (A13 to A15) when memory is expanded externally, and serial interface I/O.

P33 and P34 can be selected as normal output or N-ch open-drain output.

(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 bus 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.

(iv) SI4 (Serial input 4) ··· input

This is the serial receive data input pin of CSI4.

(v) SO4 (Serial output 4) ··· output

This is the serial transmit data output pin of CSI4.

(vi) $\overline{\text{SCK4}}$ (Serial clock 4) ··· 3-state I/O

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

(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 bus 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 bus 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 bus 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.

(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		$\overline{\text{UBEN}}$	$\overline{\text{LBEN}}$	AD0
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) $\overline{\textbf{HLDAK}}$ (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/SB1 and V850/SB2 receive a bus hold request.

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

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

This is an input pin by which an external device requests the V850/SB1 and V850/SB2 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/SB1 and V850/SB2 complete 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) $\overline{\textbf{WRL}}$ (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) $\overline{\textbf{WRH}}$ (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) $\overline{\textbf{RD}}$ (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, an address bus (A5 to A12) when memory is expanded externally, key return input, and IEBus data I/O (V850/SB2 only).

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 bus 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.

(iii) KR0 to KR7 (Key return 0 to 7) ... input

These are key return input pins. Their operations are specified by the key return mode register (KRM).

(iv) $\overline{\text{IERX}}$ (IEBus receive data) ... input

This is an IEBus data input signal. This pin is only available in the V850/SB2.

(v) $\overline{\text{IETX}}$ (IEBus transmit data) ... output

This is an IEBus data output signal. This pin is only available in the V850/SB2.

(11) P110 to P113 (Port 11) ··· 3-state I/O

P110 to P113 constitute a 4-bit I/O port that can be set to input or output in 1-bit units.

P110 to P113 can also function as an address bus (A1 to A4) when memory is expanded externally, signal ($\overline{\text{WAIT}}$) that inserts waits into the bus cycle and a control.

(a) Port function

P110 to P113 can be set to input or output in 1-bit units using the port 11 mode register (PM11).

(b) Alternate functions

(i) A1 to A4 (Address bus 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) $\overline{\text{WAIT}}$ (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.

ON/OFF switching of the wait function is performed by the port alternate function control register (PAC).

Caution Because the supply voltage to the I/O buffer of the $\overline{\text{WAIT}}$ pin is EV_{DD}, if the voltage of EV_{DD} and that of BV_{DD} differ, use EV_{DD} as the voltage of the external wait signal, instead of BV_{DD}.

(12) $\overline{\text{RESET}}$ (Reset) ··· input

The $\overline{\text{RESET}}$ 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 STOP mode).

(13) REGC (Regulator control) ... input

This pin is used to connect the capacitor for the regulator.

(14) CLKOUT (Clock out) … output

This pin outputs the bus clock generated internally.

(15) X1, X2 (Crystal)

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

(16) XT1, XT2 (Crystal for subclock)

These pins are used to connect the resonator that generates the subclock.

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

This is the analog positive power supply pin for the A/D converter or alternate-function port.

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

This is the ground pin for the A/D converter or alternate-function port.

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

This is the reference voltage supply pin for the A/D converter.

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

This is the positive power supply pin for the bus interface and its alternate-function ports.

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

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

(22) EV_{DD} (Power supply for port)

This is the positive power supply pin for I/O ports and alternate-function pins (except for the alternate-function ports of the bus interface).

(23) EV_{SS} (Ground for port)

This is the ground pin for I/O ports and alternate-function pins (except for the alternate-function ports of the bus interface).

(24) V_{DD} (Power supply)

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

(25) V_{SS} (Ground)

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

(26) V_{PP} (Programming power supply)

This is the positive power supply pin used for flash memory programming mode.
This pin is used in the flash memory versions. In normal operation mode, connect directly to V_{SS}.

(27) IC (Internally connected)

This is an internally connected pin used in the mask ROM versions. Be sure to connect directly to V_{SS}.

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

(1/2)

Pin	Alternate Function	I/O Buffer Power Supply	I/O Circuit Type	Recommended Connection Method
P00	NMI	EV_{DD}	8-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P01 to P04	INTP0 to INTP3			
P05	INTP4/ADTRG			
P06	INTP5/RTPTRG			
P07	INTP6			
P10	SI0/SDA0	EV_{DD}	10-A	
P11	SO0		26	
P12	$\overline{SCK0}$/SCL0		10-A	
P13	SI1/RXD0		8-A	
P14	SO1/TXD0		26	
P15	$\overline{SCK1}$/ASCK0		10-A	
P20	SI2/SDA1	EV_{DD}	10-A	
P21	SO2		26	
P22	$\overline{SCK2}$/SCL1		10-A	
★ P23	SI3/RXD1		8-A	
P24	SO3/TXD1		26	
P25	$\overline{SCK3}$/ASCK1		10-A	
P26, P27	TI2/TO2, TI3/TO3		8-A	
P30, P31	TI00, TI01	EV_{DD}	8-A	
P32	TI10/SI4			
★ P33	TI11/SO4		10-A	
P34	TO0/A13/$\overline{SCK4}$			
P35	TO1/A14		5-A	
P36	TI4/TO4/A15		8-A	
P37	TI5/TO5			
P40 to P47	AD0 to AD7	BV_{DD}	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	AV_{DD}	9	Independently connect to AV_{DD} or AV_{SS} via a resistor
P80 to P83	ANI8 to ANI11			

(2/2)

Pin	Alternate Function	I/O Buffer Power Supply	I/O Circuit Type	Recommended Connection Method
P90	$\overline{\text{LBEN}}$/$\overline{\text{WRL}}$	BV_{DD}	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
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 to P103	RTP0/A5/KR0 to RTP3/A8/KR3	EV_{DD}	10-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open
P104	RTP4/A9/KR4/$\overline{\text{IERX}}$			
P105	RTP5/A10/KR5/$\overline{\text{IETX}}$			
P106, P107	RTP6/A11/KR6, RTP7/A12/KR7			
P110	A1/$\overline{\text{WAIT}}$	EV_{DD}	5-A	
P111 to P113	A2 to A4			
AV_{REF}	–	–	–	Connect to AV_{SS} via a resistor
CLKOUT	–	BV_{DD}	4	Leave open
$\overline{\text{RESET}}$	–	EV_{DD}	2	–
X1	–	–	–	–
X2	–	–	–	–
XT1	–	–	16	Connect to V_{SS} via a resistor
XT2	–	–	16	Leave open
V_{PP} Note 1	–	–	–	Connect to V_{SS}
IC Note 2	–	–	–	Connect directly to V_{SS}
V_{SS}	–	–	–	–
AV_{DD}	–	–	–	–
AV_{SS}	–	–	–	–
BV_{DD}	–	–	–	–
BV_{SS}	–	–	–	–
EV_{DD}	–	–	–	–
EV_{SS}	–	–	–	–

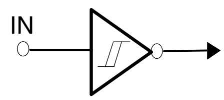
Notes 1. Flash memory versions only

2. Mask ROM versions only

2.5 Pin I/O Circuits

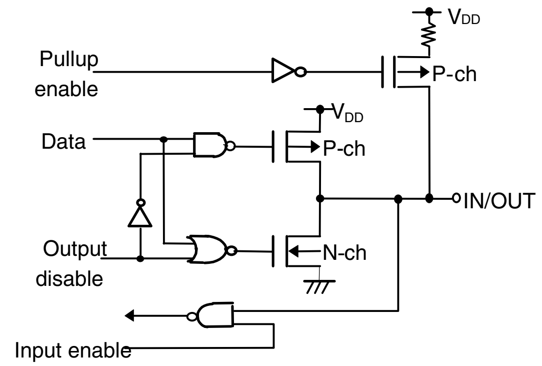
(1/2)

Type 2

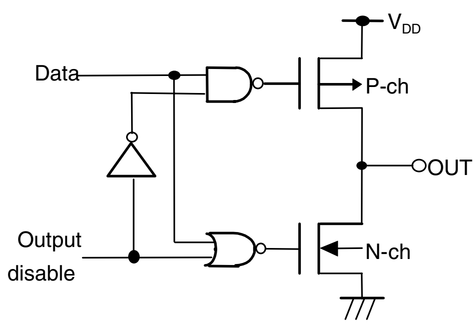


Schmitt-triggered input with hysteresis characteristics

Type 5-A

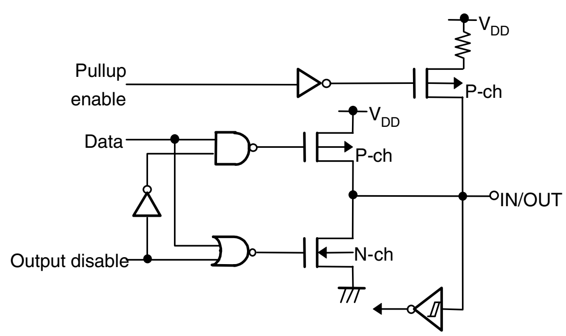


Type 4

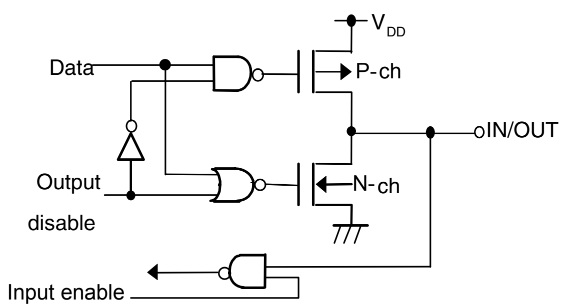


Push-pull output that can be set for high impedance output (both P-ch and N-ch off).

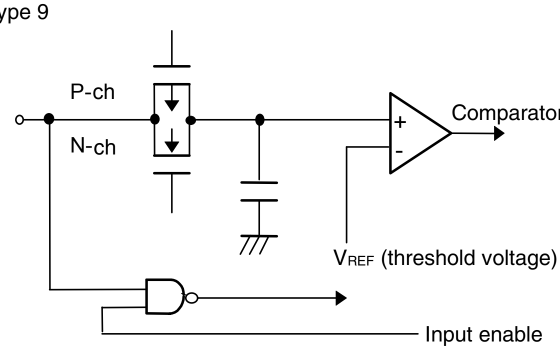
Type 8-A



Type 5

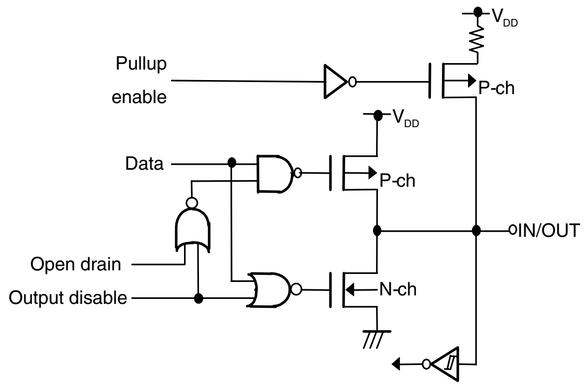


Type 9

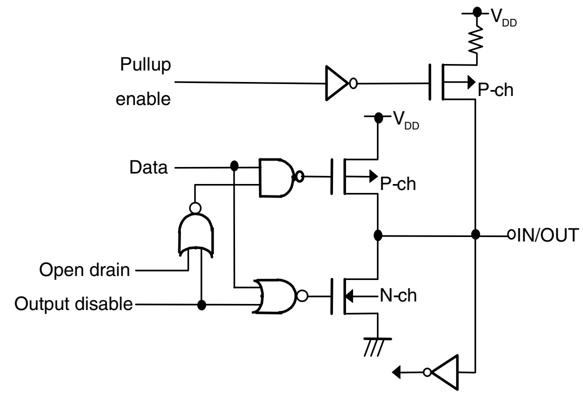


(2/2)

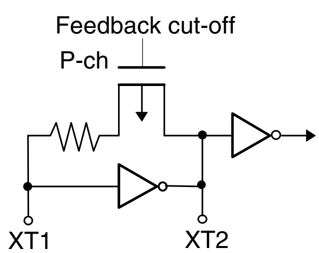
Type 10-A



Type 26



Type 16



CHAPTER 3 CPU FUNCTIONS

The CPU of the V850/SB1 and V850/SB2 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 V850/SB1 (A version, B version): 50 ns (@20 MHz internal operation)
V850/SB2 (A version, B version): 79 ns (@12.58 MHz internal operation)
V850/SB2 (H version): 53 ns (@18.87 MHz internal operation)

- Address space: 16 MB linear
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division 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/SB1 and V850/SB2 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers are 32 bits wide. For details, refer to **V850 Series Architecture User's Manual**

Figure 3-1. CPU Register Set

Program register set

31	0
r0	Zero Register
r1	Reserved for Address Register
r2	
r3	Stack Pointer (SP)
r4	Global Pointer (GP)
r5	Text Pointer (TP)
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	Element Pointer (EP)
r31	Link Pointer (LP)

31	0
PC	Program Counter

System register set

31	0
EIPC	Exception/Interrupt PC
EIPSW	Exception/Interrupt PSW

31	0
FEPC	Fatal Error PC
FEPSW	Fatal Error PSW

31	0
ECR	Exception Cause Register

31	0
PSW	Program Status Word

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, and care must be exercised when using these registers. Also, r1, r3, r4, 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.

There are cases when r2 is used by the real-time OS. If r2 is not used by the real-time OS, 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 r2 is not used by the real-time OS)	
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 24, it is ignored.

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

After reset: 00000000H

Symbol	31 to 24	23 to 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 PC and PSW when NMI occurs.
3	FEPSW		
4	ECR	Interrupt source register	If 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 exception code of NMI is set. The lower 16 bits are called EICC, to which exception code of exception/interrupt is set.
5	PSW	Program status word	A program status word is a collection of flags that indicate program status (instruction execution result) and CPU status.
6 to 31	Reserved		

To read/write these system registers, specify a 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 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 acknowledgment enabled (EI).
1	Maskable interrupt acknowledgment disabled (DI). This flag is set (1) when a maskable interrupt request is acknowledged. For details, refer to **5.3.6 ID flag.**

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	Carry or borrow has not occurred.
1	Carry or borrow occurred.

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

(2/2)

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.

<table>
<tr><th rowspan="2">Status of operation result</th><th colspan="3">Flag status</th><th rowspan="2">Saturation-processed operation result</th></tr>
<tr><th>SAT</th><th>OV</th><th>S</th></tr>
<tr><td>Maximum positive value exceeded</td><td>1</td><td>1</td><td>0</td><td>7FFFFFFFH</td></tr>
<tr><td>Maximum negative value exceeded</td><td>1</td><td>1</td><td>1</td><td>80000000H</td></tr>
<tr><td>Positive (not exceeding the maximum)</td><td rowspan="2">Retains the value before operation</td><td rowspan="2">0</td><td>0</td><td rowspan="2">Operation result itself</td></tr>
<tr><td>Negative (not exceeding the maximum)</td><td>1</td></tr>
</table>

3.3 Operation Modes

The V850/SB1 and V850/SB2 have 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 for port mode, execution branches to the reset entry address of the internal ROM, and instruction processing written in the internal ROM is started. However, external expansion mode that connects external device to external memory area is enabled by setting in the memory expansion mode register (MM) by instruction.

(2) Flash memory programming mode

This mode is provided only in the flash memory versions. 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

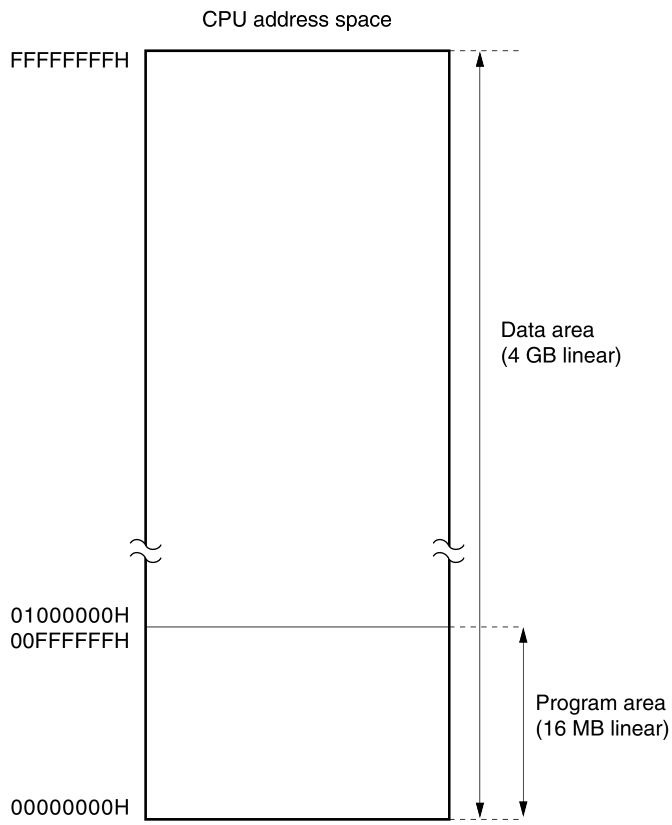
3.4 Address Space

3.4.1 CPU address space

The CPUs of the V850/SB1 and V850/SB2 are of 32-bit architecture and support 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.

The CPU address space is shown below.

Figure 3-2. CPU Address Space

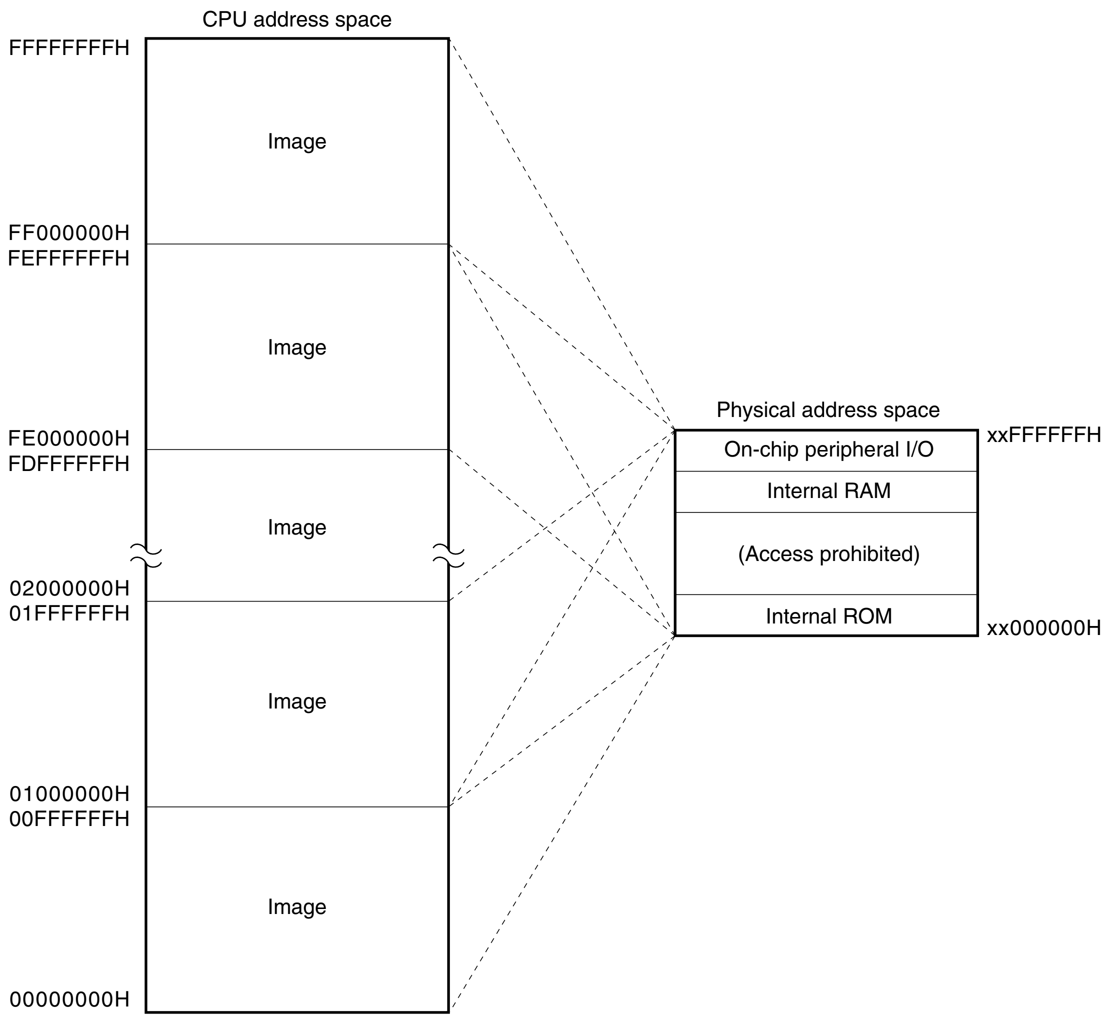


3.4.2 Image

The core 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. The image of the virtual addressing space is shown below.

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 Wrap-around of CPU address space

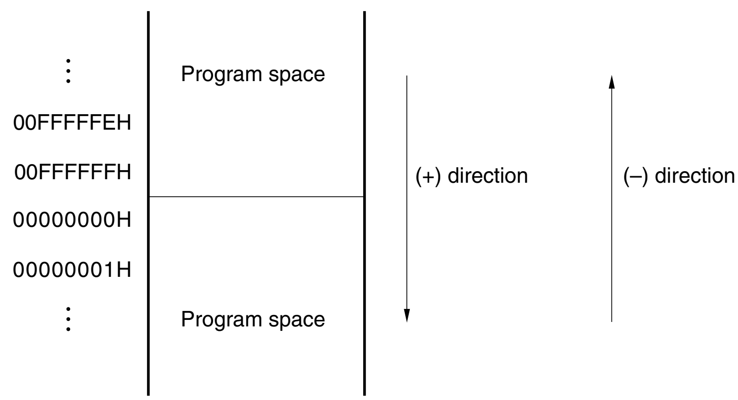
(1) Program space

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 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

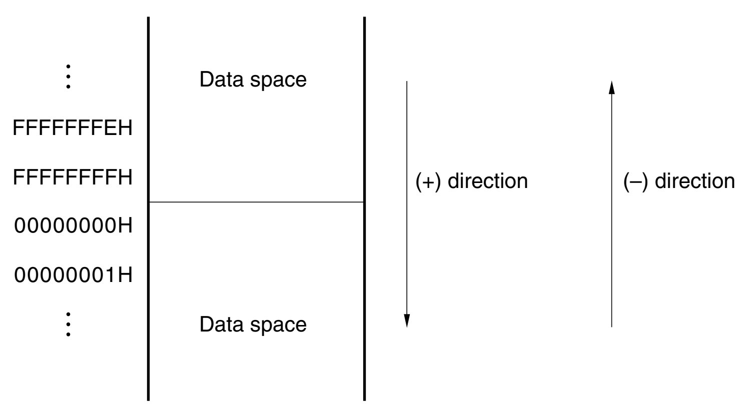


(2) Data space

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.

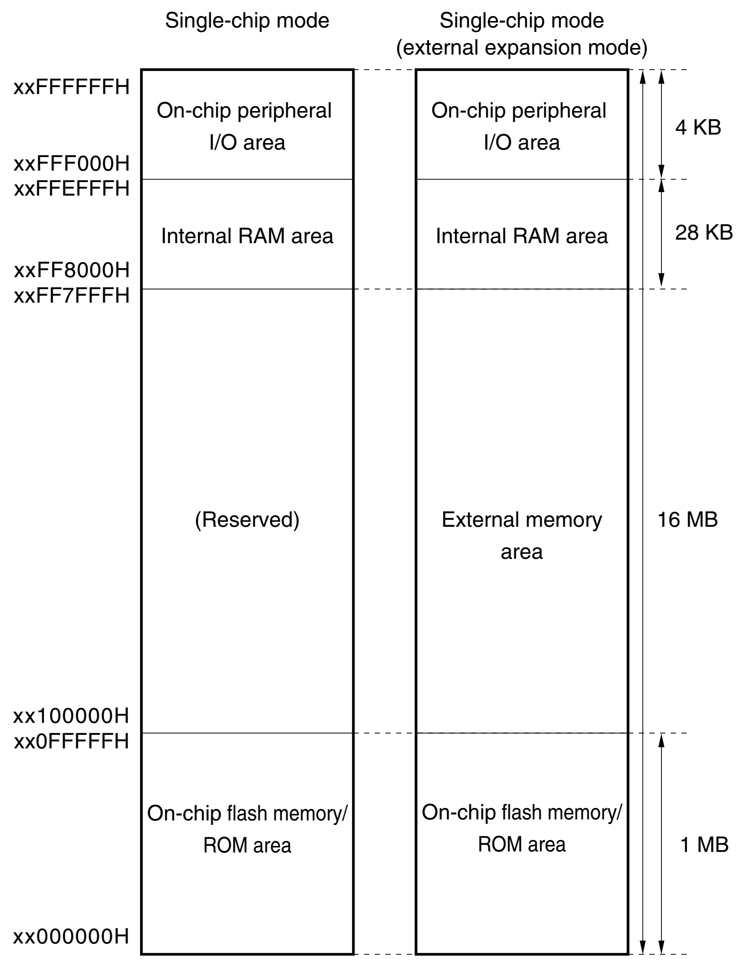
Figure 3-5. Data Space



3.4.4 Memory map

The V850/SB1 and V850/SB2 reserve areas as shown below.

Figure 3-6. Memory Map



3.4.5 Area

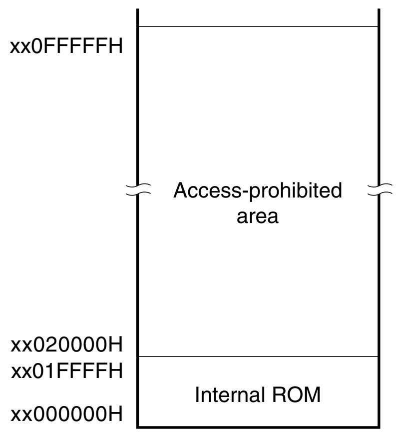
(1) Internal ROM/flash memory area

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

★ **(a) V850/SB1 (μPD703031A, 703031AY, 703031B, 703031BY)**
V850/SB2 (μPD703034A, 703034AY, 703034B, 703034BY)

128 KB are available for the addresses xx000000H to xx01FFFFH.
Addresses xx020000H to xx0FFFFFH are an access-prohibited area

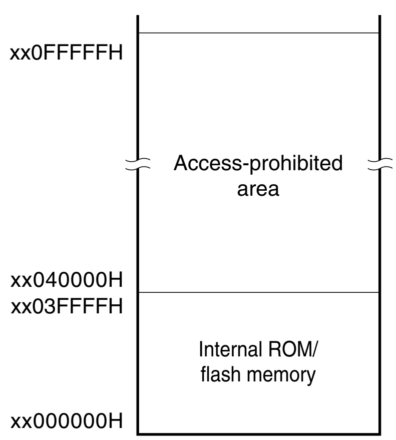
Figure 3-7. Internal ROM Area (128 KB)



★ **(b) V850/SB1**
(μPD703033A, 703033AY, 70F3033A, 70F3033AY, 703033B, 703033BY, 70F3033B, 70F3033BY)
V850/SB2
(μPD703035A, 703035AY, 70F3035A, 70F3035AY, 703035B, 703035BY, 70F3035B, 70F3035BY)

256 KB are available for the addresses xx000000H to xx03FFFFH.
Addresses xx040000H to xx0FFFFFH are an access-prohibited area

Figure 3-8. Internal ROM/Flash Memory Area (256 KB)

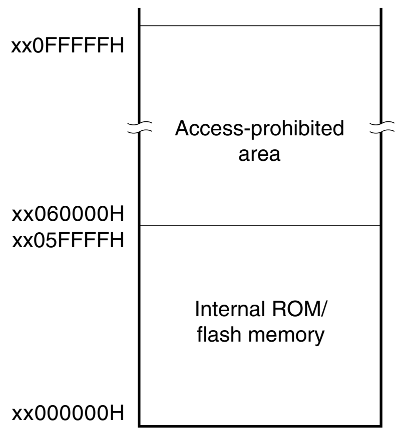


★ **(c) V850/SB1 (μPD703030B, 703030BY, 70F3030B, 70F3030BY)**
V850/SB2 (μPD703036H, 703036HY, 70F3036H, 70F3036HY)

384 KB are available for the addresses xx000000H to xx05FFFFH.

Addresses xx060000H to xx0FFFFFH are an access-prohibited area

Figure 3-9. Internal ROM/Flash Memory Area (384 KB)

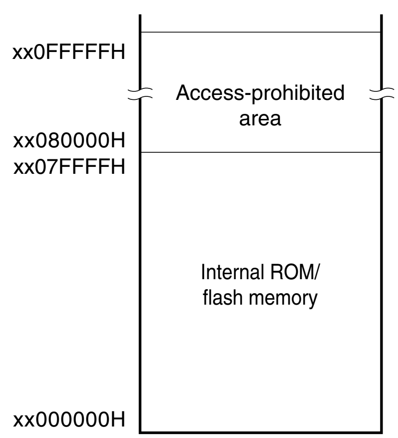


★ **(d) V850/SB1**
(μPD703032A, 703032AY, 70F3032A, 70F3032AY, 703032B, 703032BY, 70F3032B, 70F3032BY)
V850/SB2
(μPD703037A, 703037AY, 70F3037A, 70F3037AY, 703037H, 703037HY, 70F3037H, 70F3037HY)

512 KB are available for the addresses xx000000H to xx07FFFFH.

Addresses xx080000H to xx0FFFFFH are an access-prohibited area

Figure 3-10. Internal ROM/Flash Memory Area (512 KB)



Interrupt/exception table

The V850/SB1 and V850/SB2 increase 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 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	Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000000H	RESET	000001D0H	INTTM6
00000010H	NMI	000001E0H	INTTM7
00000020H	INTWDT	000001F0H	INTIIC0 Note/INTCSI0
00000040H	TRAP0n (n = 0 to F)	00000200H	INTSER0
00000050H	TRAP1n (n = 0 to F)	00000210H	INTSR0/INTCSI1
00000060H	ILGOP	00000220H	INTST0
00000080H	INTWDTM	00000230H	INTCSI2
00000090H	INTP0	00000240H	INTIIC1 Note
000000A0H	INTP1	00000250H	INTSER1
000000B0H	INTP2	00000260H	INTSR1/INTCSI3
000000C0H	INTP3	00000270H	INTST1
000000D0H	INTP4	00000280H	INTCSI4
000000E0H	INTP5	00000290H	INTIE1 (V850/SB2 only)
000000F0H	INTP6	000002A0H	INTIE2 (V850/SB2 only)
00000140H	INTWTNI	000002B0H	INTAD
00000150H	INTTM00	000002C0H	INTDMA0
00000160H	INTTM01	000002D0H	INTDMA1
00000170H	INTTM10	000002E0H	INTDMA2
00000180H	INTTM11	000002F0H	INTDMA3
00000190H	INTTM2	00000300H	INTDMA4
000001A0H	INTTM3	00000310H	INTDMA5
000001B0H	INTTM4	00000320H	INTWTN
000001C0H	INTTM5	00000330H	INTKR

Note Available only in the Y versions (products with on-chip I^2C).

(2) Internal RAM area

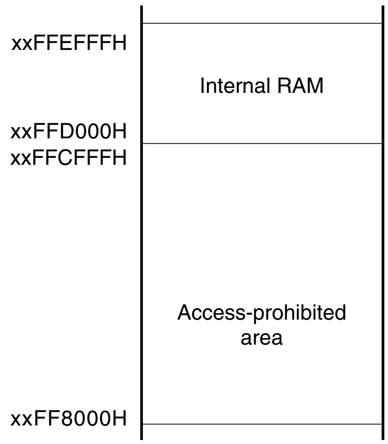
An area of 28 KB maximum is reserved for the internal RAM area.

★ **(a) V850/SB1 (μPD703031B, 703031BY), V850/SB2 (μPD703034B, 703034BY)**

8 KB are available for the addresses xxFFD000H to xxFFEFFFH.

Addresses xxFF8000H to xxFFCFFFH are an access-prohibited area

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

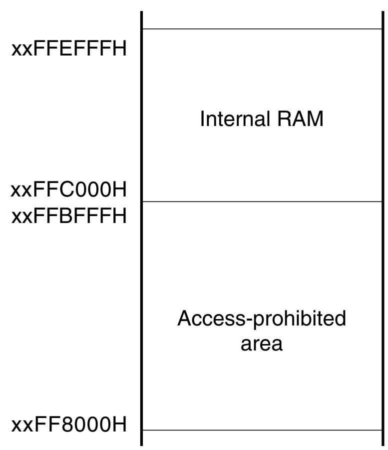


★ **(b) V850/SB1 (μPD703031A, 703031AY), V850/SB2 (μPD703034A, 703034AY)**

12 KB are available for the addresses xxFFC000H to xxFFEFFFH.

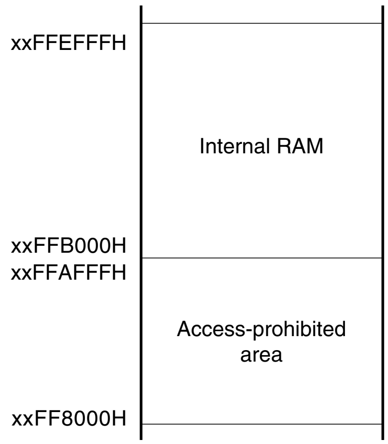
Addresses xxFF8000H to xxFFBFFFH are an access-prohibited area

Figure 3-12. Internal RAM Area (12 KB)



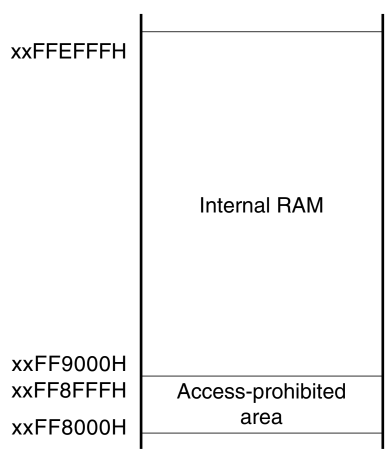
★ **(c) V850/SB1**
(μPD703033A, 703033AY, 70F3033A, 70F3033AY, 703033B, 703033BY, 70F3033B, 70F3033BY)
V850/SB2
(μPD703035A, 703035AY, 70F3035A, 70F3035AY, 703035B, 703035BY, 70F3035B, 70F3035BY)
16 KB are available for the addresses xxFFB000H to xxFFEFFFH.
Addresses xxFF8000H to xxFFAFFFH are an access-prohibited area

Figure 3-13. Internal RAM Area (16 KB)



★ **(d) V850/SB1**
(μPD703030B, 703030BY, 70F3030B, 70F3030BY, 703032A, 703032AY, 70F3032A, 70F3032AY, 703032BY, 70F3032B, 70F3032BY)
V850/SB2
(μPD703036H, 703036HY, 70F3036H, 70F3036HY, 703037A, 703037AY, 70F3037A, 70F3037AY, 703037H, 703037HY, 70F3037H, 70F3037HY)
24 KB are available for the addresses xxFF9000H to xxFFEFFFH.
Addresses xxFF8000H to xxFF8FFFH are an access-prohibited area

Figure 3-14. Internal RAM Area (24 KB)

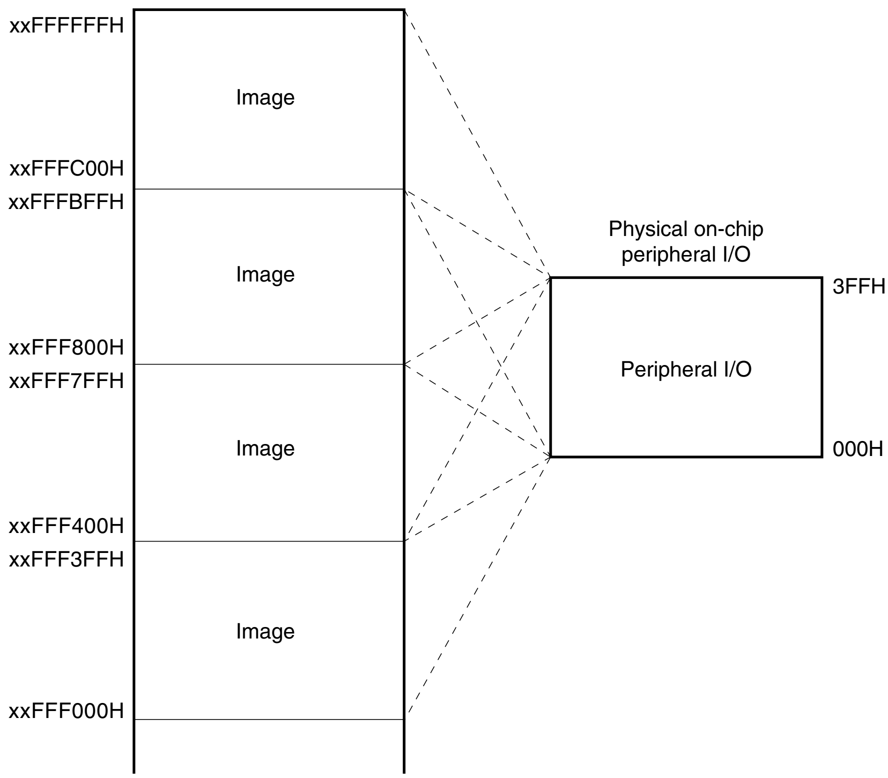


(3) On-chip peripheral I/O area

A 4 KB area of addresses FFF000H to FFFFFFH is reserved as an on-chip peripheral I/O area. The V850/SB1 and V850/SB2 are 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 the operation mode specification and the 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 3-15. On-Chip Peripheral I/O Area



Cautions 1. The least significant bit of an address is not decoded since all registers reside on 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 with n address 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 to the register with n address and the second operation (higher 16 bits) accesses to the register with n + 2 address.

4. If a register with n address 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 to the register with n address and the second operation (higher 16 bits) accesses to the register with n + 2 address.

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

The V850/SB1 and V850/SB2 can use an area of up to 16 MB (xx100000H to xxFF7FFFH) for external memory accesses (in single-chip mode: external expansion).

64 K, 256 K, 1 M, 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 3-16. External Memory Area (When Expanded to 64 K, 256 K, or 1 MB)

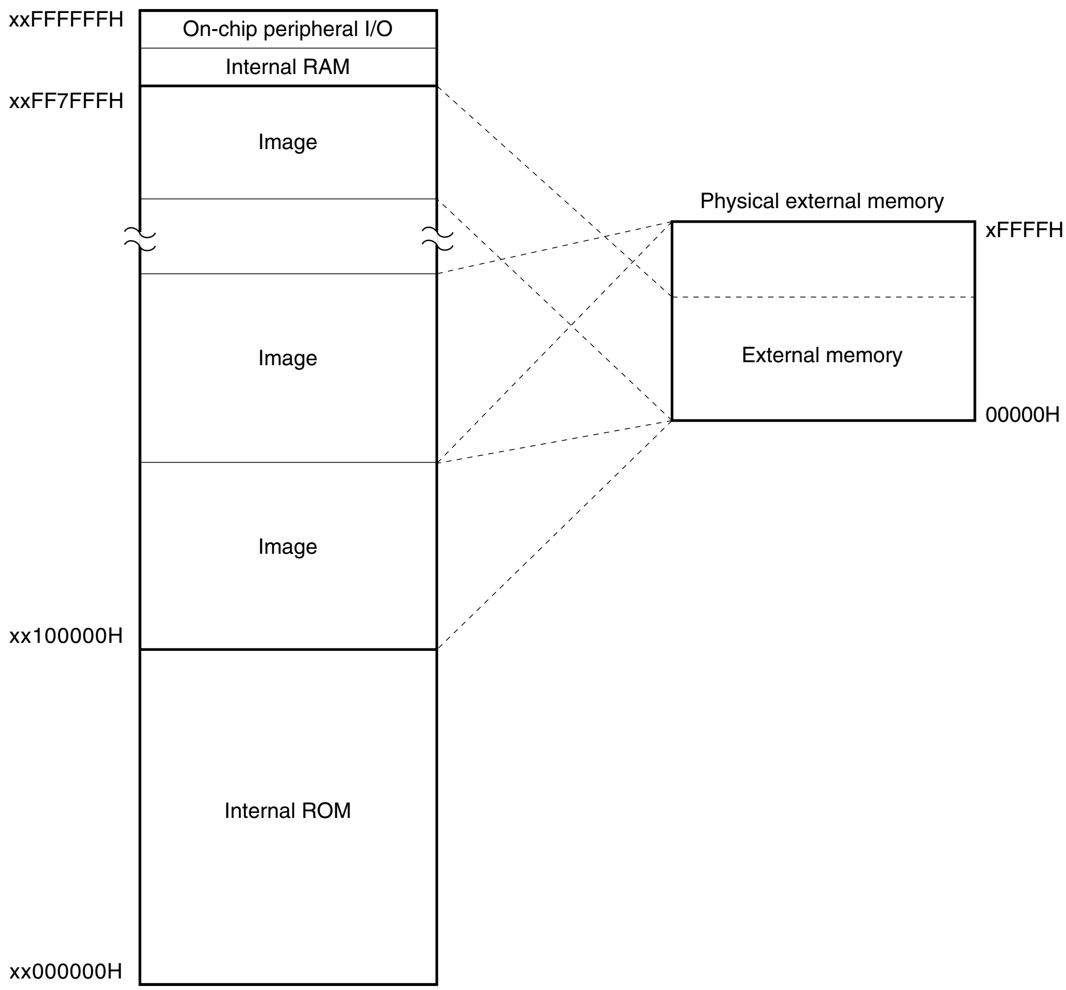
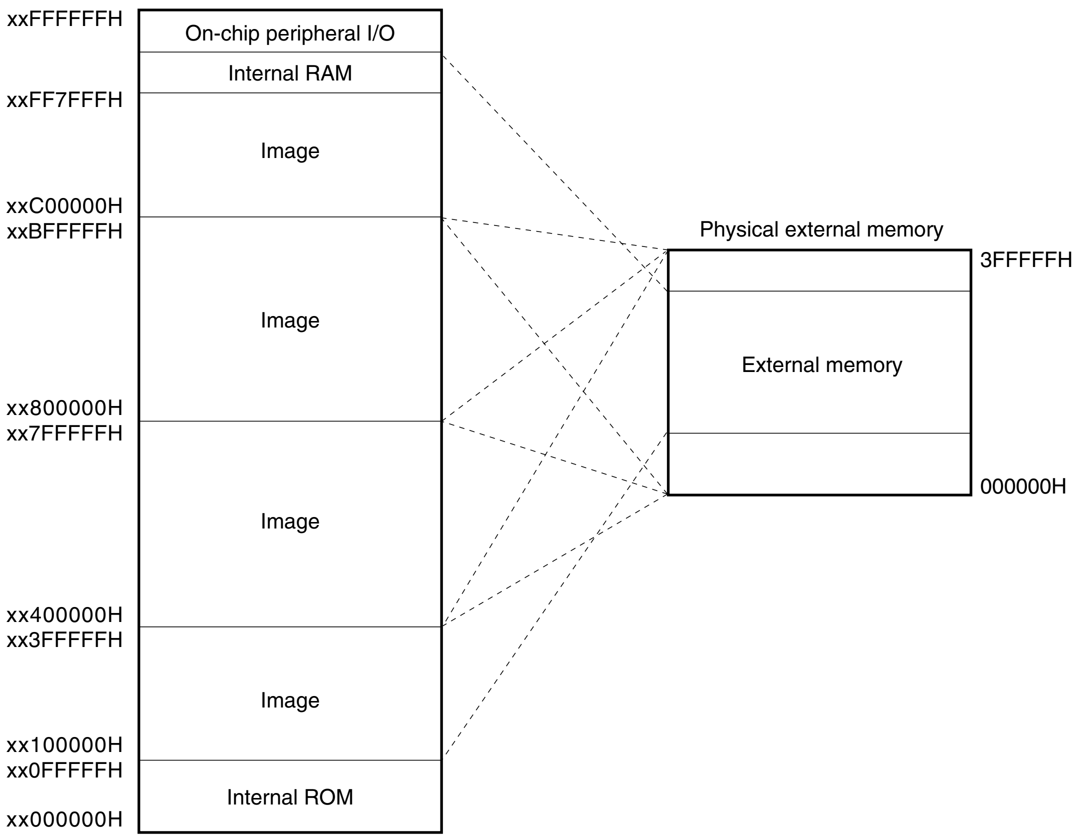


Figure 3-17. External Memory Area (When Expanded to 4 MB)



3.4.6 External expansion mode

The V850/SB1 and V850/SB2 allow 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 in the external expansion mode by using the memory expansion mode register (MM).

The address bus (A1 to A15) is set to multiplexed output with data bus (D1 to D15), though separate output is also available by setting the memory address output mode register (MAM) (see the User's Manual of relevant in-circuit emulator about debugging when using the separate bus).

Caution Because the A1 pin and $\overline{\text{WAIT}}$ pin are alternate-function pins, the wait function by the $\overline{\text{WAIT}}$ pin cannot be used when using a separate bus (programmable wait can be used however). Similarly, a separate bus cannot be used when the wait function by the $\overline{\text{WAIT}}$ pin is being used.

Because the V850/SB1 and V850/SB2 are fixed to single-chip mode in the normal operation mode, the port alternate pins become the port mode, thereby the external memory cannot be used. When the external memory is used (external expansion mode), specify the MM register by the program.

(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 the external memory area of up to 4 MB. However, the 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)

<table>
<tr><th>MM2</th><th>MM1</th><th>MM0</th><th>Address space</th><th>Port 4</th><th>Port 5</th><th colspan="3">Port 6</th><th>Port 9</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>–</td><td colspan="6">Port mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>64 KB expansion mode</td><td rowspan="4">AD0 to AD7</td><td rowspan="4">AD8 to AD15</td><td colspan="3"></td><td rowspan="4">$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$, R/$\overline{\text{W}}$, $\overline{\text{DSTB}}$, ASTB, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>256 KB expansion mode</td><td rowspan="3">A16, A17</td><td colspan="2"></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1 MB expansion mode</td><td rowspan="2">A18, A19</td><td></td></tr>
<tr><td>1</td><td>1</td><td>×</td><td>4 MB expansion mode</td><td>A20, A21</td></tr>
<tr><td colspan="4">Other than above</td><td colspan="6">RFU (reserved)</td></tr>
</table>

Caution Before switching to the external expansion mode, be sure to set P93 and P94 of Port 9 (P9) to 1.

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

(2) Memory address output mode register (MAM)

Sets the mode of ports 3, 10, and 11. Separate output can be set for the address bus (A1 to A15) in the external expansion mode.

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

After reset: 00H W Address: FFFFF068H

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

<table>
<tr><th>MAM2</th><th>MAM1</th><th>MAM0</th><th>Address space</th><th>Port 11</th><th colspan="2">Port 10</th><th colspan="3">Port 3</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>–</td><td colspan="6">Port mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>32 bytes</td><td rowspan="6">A1 to A4</td><td colspan="5"></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>512 bytes</td><td rowspan="5">A5 to A8</td><td colspan="4"></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>8 KB</td><td rowspan="4">A9 to A12</td><td colspan="3"></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16 KB</td><td rowspan="3">A13</td><td colspan="2"></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>32 KB</td><td rowspan="2">A14</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>64 KB</td><td>A15</td></tr>
</table>

Caution Debugging the memory address output mode register (MAM) an in-circuit emulator is not available. Also, setting the MAM register by software cannot switch to the separate bus. For details, refer to the relevant User's Manual of in-circuit emulator.

Remark For details of the operation of each port, see **2.3 Description of Pin Functions**.

The separate path outputs are output from P34 to P36, P100 to P107, and P110 to P113. The procedure for performing separate path output is shown below.

<1> Set the Pn bit of Port m (Pm) used for separate output to 0 (m = 3, 10, 11).
<2> Set the PMn bit of the port m mode register (PMm) to 0 (output mode) (m = 3, 10, 11).
<3> When the port to be used for the separate path is used as an alternate-function pin for other than the separate path, turn off the function used by the alternate-function pin.
<4> Set the memory address output mode register (MAM).
<5> Set the memory expansion mode register (MM).

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

3.4.7 Recommended use of address space

The architectures of the V850/SB1 and V850/SB2 require that a register that serves as a pointer be secured for address generation in operand data accessing for data space. The address in this pointer register ±32 KB can be accessed directly from instruction. However, general-purpose register used as a pointer register is limited. Therefore, by minimizing the deterioration of 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 saved because instructions for calculating pointer addresses are not required.

To enhance the efficiency of using the pointer in connection with the memory maps of the V850/SB1 and V850/SB2, 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 through the wrap-around 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/SB1 or V850/SB2, 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 address sign-extended to 32 bits.

(a) Application of wrap-around

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. 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 3-18. Application of Wrap-Around

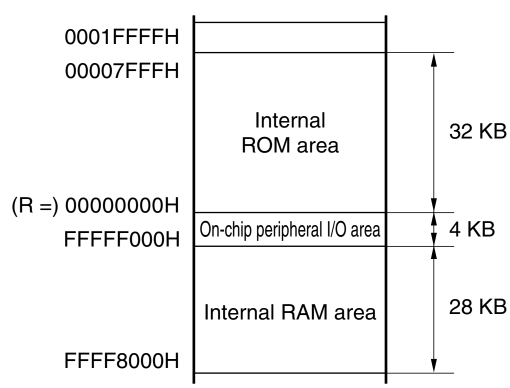
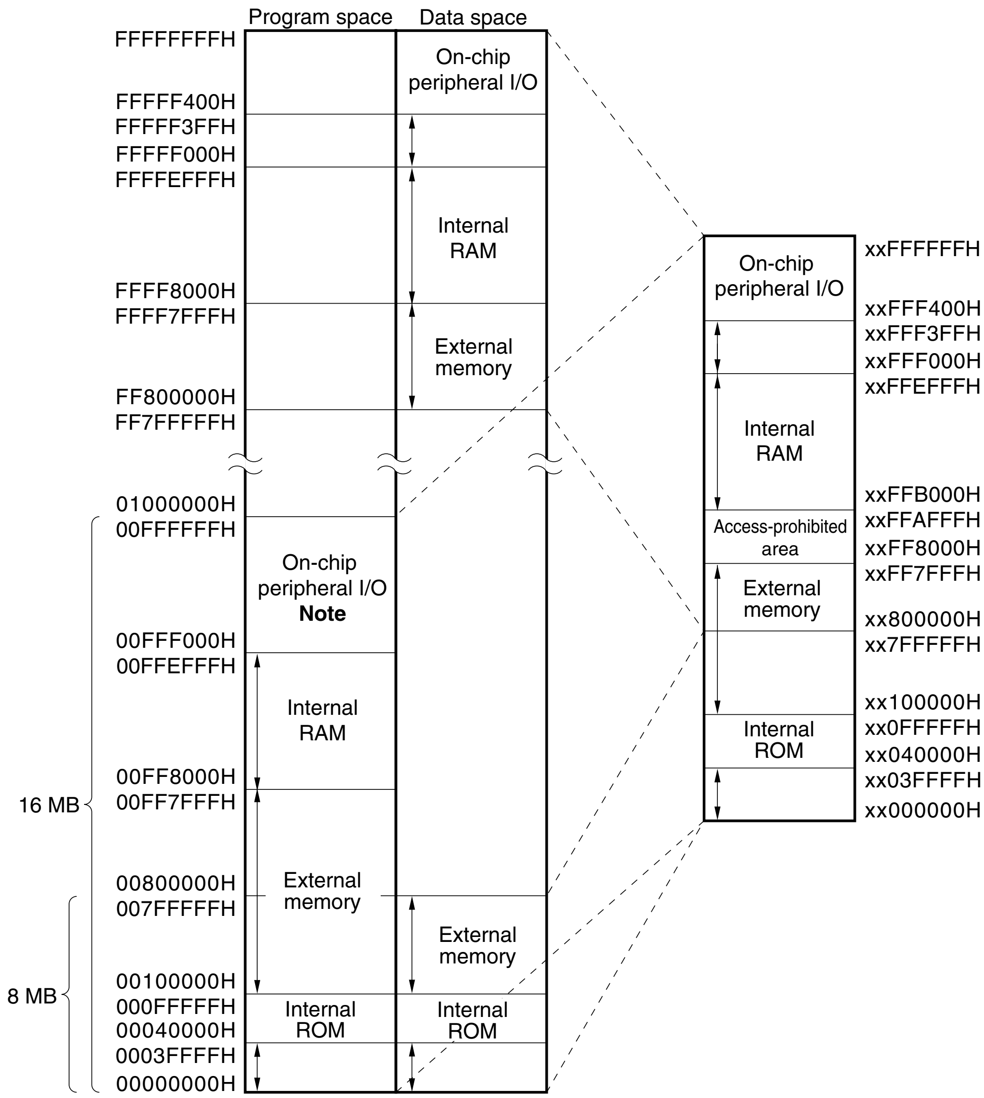


Figure 3-19. Recommended Memory Map (Flash Memory Version)



Note This area cannot be used as a program area.

Remarks 1. The arrows indicate the recommended area.

★ **2.** This is a recommended memory map for V850/SB1 (μPD70F3033A, 70F3033AY, 70F3033B, 70F3033BY), V850/SB2 (μPD70F3035A, 70F3035AY, 70F3035B, 70F3035BY).

3.4.8 Peripheral I/O registers

(1/7)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF000H	Port 0	P0	R/W	√	√			00H[Note]
FFFFF002H	Port 1	P1		√	√			
FFFFF004H	Port 2	P2		√	√			
FFFFF006H	Port 3	P3		√	√			
FFFFF008H	Port 4	P4		√	√			
FFFFF00AH	Port 5	P5		√	√			
FFFFF00CH	Port 6	P6		√	√			
FFFFF00EH	Port 7	P7	R	√	√			Undefined
FFFFF010H	Port 8	P8		√	√			
FFFFF012H	Port 9	P9	R/W	√	√			00H[Note]
FFFFF014H	Port 10	P10		√	√			
FFFFF016H	Port 11	P11		√	√			
FFFFF020H	Port 0 mode register	PM0		√	√			FFH
FFFFF022H	Port 1 mode register	PM1		√	√			3FH
FFFFF024H	Port 2 mode register	PM2		√	√			FFH
FFFFF026H	Port 3 mode register	PM3		√	√			
FFFFF028H	Port 4 mode register	PM4		√	√			
FFFFF02AH	Port 5 mode register	PM5		√	√			
FFFFF02CH	Port 6 mode register	PM6		√	√			3FH
FFFFF032H	Port 9 mode register	PM9		√	√			7FH
FFFFF034H	Port 10 mode register	PM10		√	√			FFH
FFFFF036H	Port 11 mode register	PM11		√	√			1FH
FFFFF040H	Port alternate function control register	PAC		√	√			00H
FFFFF04CH	Memory expansion mode register	MM		√	√			
FFFFF060H	Data wait control register	DWC				√		FFFFH
FFFFF062H	Bus cycle control register	BCC				√		AAAAH
FFFFF064H	System control register	SYC		√	√			00H
FFFFF068H	Memory address output mode register	MAM	W		√			
FFFFF070H	Power save control register	PSC	R/W	√	√			C0H
FFFFF074H	Processor clock control register	PCC		√	√			03H
FFFFF078H	System status register	SYS		√	√			00H
FFFFF080H	Pull-up resistor option register 0	PU0		√	√			
FFFFF082H	Pull-up resistor option register 1	PU1		√	√			
FFFFF084H	Pull-up resistor option register 2	PU2		√	√			
FFFFF086H	Pull-up resistor option register 3	PU3		√	√			
FFFFF094H	Pull-up resistor option register 10	PU10		√	√			

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

(2/7)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF096H	Pull-up resistor option register 11	PU11	R/W	√	√			00H
FFFFF0A2H	Port 1 function register	PF1		√	√			
FFFFF0A4H	Port 2 function register	PF2		√	√			
FFFFF0A6H	Port 3 function register	PF3		√	√			
FFFFF0B4H	Port 10 function register	PF10		√	√			
FFFFF0C0H	Rising edge specification register 0	EGP0		√	√			
FFFFF0C2H	Falling edge specification register 0	EGN0		√	√			
FFFFF100H	Interrupt control register	WDTIC		√	√			47H
FFFFF102H	Interrupt control register	PIC0		√	√			
FFFFF104H	Interrupt control register	PIC1		√	√			
FFFFF106H	Interrupt control register	PIC2		√	√			
FFFFF108H	Interrupt control register	PIC3		√	√			
FFFFF10AH	Interrupt control register	PIC4		√	√			
FFFFF10CH	Interrupt control register	PIC5		√	√			
FFFFF10EH	Interrupt control register	PIC6		√	√			
FFFFF118H	Interrupt control register	WTNIIC		√	√			
FFFFF11AH	Interrupt control register	TMIC00		√	√			
FFFFF11CH	Interrupt control register	TMIC01		√	√			
FFFFF11EH	Interrupt control register	TMIC10		√	√			
FFFFF120H	Interrupt control register	TMIC11		√	√			
FFFFF122H	Interrupt control register	TMIC2		√	√			
FFFFF124H	Interrupt control register	TMIC3		√	√			
FFFFF126H	Interrupt control register	TMIC4		√	√			
FFFFF128H	Interrupt control register	TMIC5		√	√			
FFFFF12AH	Interrupt control register	TMIC6		√	√			
FFFFF12CH	Interrupt control register	TMIC7		√	√			
FFFFF12EH	Interrupt control register	CSIC0		√	√			
FFFFF130H	Interrupt control register	SERIC0		√	√			
FFFFF132H	Interrupt control register	CSIC1		√	√			
FFFFF134H	Interrupt control register	STIC0		√	√			
FFFFF136H	Interrupt control register	CSIC2		√	√			
FFFFF138H	Interrupt control register[Note]	IICIC1		√	√			
FFFFF13AH	Interrupt control register	SERIC1		√	√			
FFFFF13CH	Interrupt control register	CSIC3		√	√			
FFFFF13EH	Interrupt control register	STIC1		√	√			

Note Available only in the Y versions (products with on-chip I^2C).

(3/7)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF140H	Interrupt control register	CSIC4		√	√			47H
FFFFF142H	Interrupt control register[Note]	IEBIC1		√	√			
FFFFF144H	Interrupt control register[Note]	IEBIC2		√	√			
FFFFF146H	Interrupt control register	ADIC		√	√			
FFFFF148H	Interrupt control register	DMAIC0		√	√			
FFFFF14AH	Interrupt control register	DMAIC1		√	√			
FFFFF14CH	Interrupt control register	DMAIC2		√	√			
FFFFF14EH	Interrupt control register	DMAIC3		√	√			
FFFFF150H	Interrupt control register	DMAIC4		√	√			
FFFFF152H	Interrupt control register	DMAIC5		√	√			
FFFFF154H	Interrupt control register	WTNIC		√	√			
FFFFF156H	Interrupt control register	KRIC		√	√			
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			√		
FFFFF182H	DMA internal RAM address register 0	DRA0				√		
FFFFF184H	DMA byte count register 0	DBC0			√			
FFFFF186H	DMA channel control register 0	DCHC0		√	√			00H
FFFFF190H	DMA peripheral I/O address register 1	DIOA1				√		Undefined
FFFFF192H	DMA internal RAM address register 1	DRA1				√		
FFFFF194H	DMA byte count register 1	DBC1			√			
FFFFF196H	DMA channel control register 1	DCHC1		√	√			00H
FFFFF1A0H	DMA peripheral I/O address register 2	DIOA2				√		Undefined
FFFFF1A2H	DMA internal RAM address register 2	DRA2				√		
FFFFF1A4H	DMA byte count register 2	DBC2			√			
FFFFF1A6H	DMA channel control register 2	DCHC2		√	√			00H
FFFFF1B0H	DMA peripheral I/O address register 3	DIOA3				√		Undefined
FFFFF1B2H	DMA internal RAM address register 3	DRA3				√		
FFFFF1B4H	DMA byte count register 3	DBC3			√			
FFFFF1B6H	DMA channel control register 3	DCHC3		√	√			00H
FFFFF1C0H	DMA peripheral I/O address register 4	DIOA4				√		Undefined
FFFFF1C2H	DMA internal RAM address register 4	DRA4				√		
FFFFF1C4H	DMA byte count register 4	DBC4			√			
FFFFF1C6H	DMA channel control register 4	DCHC4		√	√			00H
FFFFF1D0H	DMA peripheral I/O address register 5	DIOA5				√		Undefined
FFFFF1D2H	DMA internal RAM address register 5	DRA5				√		

Note Available only in the V850/SB2.

(4/7)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF1D4H	DMA byte count register 5	DBC5	R/W		√			Undefined
FFFFF1D6H	DMA channel control register 5	DCHC5		√	√			00H
FFFFF200H	16-bit timer register 0	TM0	R			√		0000H
FFFFF202H	16-bit capture/compare register 00	CR00	**Note**			√		
FFFFF204H	16-bit capture/compare register 01	CR01	**Note**			√		
FFFFF206H	Prescaler mode register 00	PRM00	R/W		√			00H
FFFFF208H	16-bit timer mode control register 0	TMC0		√	√			
FFFFF20AH	Capture/compare control register 0	CRC0		√	√			
FFFFF20CH	Timer output control register 0	TOC0		√	√			
FFFFF20EH	Prescaler mode register 01	PRM01			√			
FFFFF210H	16-bit timer register 1	TM1	R			√		0000H
FFFFF212H	16-bit capture/compare register 10	CR10	**Note**			√		
FFFFF214H	16-bit capture/compare register 11	CR11	**Note**			√		
FFFFF216H	Prescaler mode register 10	PRM10	R/W		√			00H
FFFFF218H	16-bit timer mode control register 1	TMC1		√	√			
FFFFF21AH	Capture/compare control register 1	CRC1		√	√			
FFFFF21CH	Timer output control register 1	TOC1		√	√			
FFFFF21EH	Prescaler mode register 11	PRM11			√			
FFFFF240H	8-bit counter 2	TM2	R		√			00H
FFFFF242H	8-bit compare register 2	CR20	R/W		√			
FFFFF244H	Timer clock selection register 20	TCL20			√			
FFFFF246H	8-bit timer mode control register 2	TMC2		√	√			04H
FFFFF24AH	16-bit counter 23 (during cascade connection only)	TM23	R			√		0000H
FFFFF24CH	16-bit compare register 23 (during cascade connection only)	CR23	R/W			√		
FFFFF24EH	Timer clock selection register 21	TCL21			√			00H
FFFFF250H	8-bit counter 3	TM3	R		√			
FFFFF252H	8-bit compare register 3	CR30	R/W		√			
FFFFF254H	Timer clock selection register 30	TCL30			√			
FFFFF256H	8-bit timer mode control register 3	TMC3		√	√			04H
FFFFF25EH	Timer clock selection register 31	TCL31			√			00H
FFFFF260H	8-bit counter 4	TM4	R		√			
FFFFF262H	8-bit compare register 4	CR40	R/W		√			
FFFFF264H	Timer clock selection register 40	TCL40			√			
FFFFF266H	8-bit timer mode control register 4	TMC4		√	√			04H

Note In compare mode: R/W
In capture mode: R

(5/7)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF26AH	16-bit counter 45 (during cascade connection only)	TM45	R			√		0000H
FFFFF26CH	16-bit compare register 45 (during cascade connection only)	CR45	R/W			√		
FFFFF26EH	Timer clock selection register 41	TCL41			√			00H
FFFFF270H	8-bit counter 5	TM5	R		√			
FFFFF272H	8-bit compare register 5	CR50	R/W		√			
FFFFF274H	Timer clock selection register 50	TCL50			√			
FFFFF276H	8-bit timer mode control register 5	TMC5		√	√			04H
FFFFF27EH	Timer clock selection register 51	TCL51			√			00H
FFFFF280H	8-bit counter 6	TM6	R		√			
FFFFF282H	8-bit compare register 6	CR60	R/W		√			
FFFFF284H	Timer clock selection register 60	TCL60			√			
FFFFF286H	8-bit timer mode control register 6	TMC6		√	√			04H
FFFFF28AH	16-bit counter 67 (during cascade connection only)	TM67	R			√		0000H
FFFFF28CH	16-bit compare register 67 (during cascade connection only)	CR67	R/W			√		
FFFFF28EH	Timer clock selection register 61	TCL61			√			00H
FFFFF290H	8-bit counter 7	TM7	R		√			
FFFFF292H	8-bit compare register 7	CR70	R/W		√			
FFFFF294H	Timer clock selection register 70	TCL70			√			
FFFFF296H	8-bit timer mode control register 7	TMC7		√	√			04H
FFFFF29EH	Timer clock selection register 71	TCL71			√			00H
FFFFF2A0H	Serial I/O shift register 0	SIO0			√			
FFFFF2A2H	Serial operation mode register 0	CSIM0		√	√			
FFFFF2A4H	Serial clock selection register 0	CSIS0			√			
FFFFF2B0H	Serial I/O shift register 1	SIO1			√			
FFFFF2B2H	Serial operation mode register 1	CSIM1		√	√			
FFFFF2B4H	Serial clock selection register 1	CSIS1			√			
FFFFF2C0H	Serial I/O shift register 2	SIO2			√			
FFFFF2C2H	Serial operation mode register 2	CSIM2		√	√			
FFFFF2C4H	Serial clock selection register 2	CSIS2			√			
FFFFF2D0H	Serial I/O shift register 3	SIO3			√			
FFFFF2D2H	Serial operation mode register 3	CSIM3		√	√			
FFFFF2D4H	Serial clock selection register 3	CSIS3		√	√			
FFFFF2E0H	Variable-length serial I/O shift register 4	SIO4				√		0000H

(6/7)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF2E2H	Variable-length serial control register 4	CSIM4	R/W	√	√			00H
FFFFF2E4H	Variable-length serial setting register 4	CSIB4		√	√			
FFFFF2E6H	Baud rate generator source clock selection register 4	BRGCN4			√			
FFFFF2E8H	Baud rate generator output clock selection register 4	BRGCK4			√			7FH
FFFFF300H	Asynchronous serial interface mode register 0	ASIM0		√	√			00H
FFFFF302H	Asynchronous serial interface status register 0	ASIS0	R	√	√			
FFFFF304H	Baud rate generator control register 0	BRGC0	R/W		√			
FFFFF306H	Transmission shift register 0	TXS0	W		√			FFH
FFFFF308H	Reception buffer register 0	RXB0	R		√			
FFFFF30EH	Baud rate generator mode control register 00	BRGMC00	R/W		√			00H
FFFFF310H	Asynchronous serial interface mode register 1	ASIM1		√	√			
FFFFF312H	Asynchronous serial interface status register 1	ASIS1	R	√	√			
FFFFF314H	Baud rate generator control register 1	BRGC1	R/W		√			
FFFFF316H	Transmission shift register 1	TXS1	W		√			FFH
FFFFF318H	Reception buffer register 1	RXB1	R		√			
FFFFF31EH	Baud rate generator mode control register 10	BRGMC10	R/W		√			00H
FFFFF320H	Baud rate generator mode control register 01	BRGMC01			√			
FFFFF322H	Baud rate generator mode control register 11	BRGMC11			√			
★ FFFFF33CH	IIC flag register 0[Note 1]	IICF0		√	√			
★ FFFFF33EH	IIC flag register 1[Note 1]	IICF1		√	√			
FFFFF340H	IIC control register 0[Note 2]	IICC0		√	√			
FFFFF342H	IIC state register 0[Note 2]	IICS0	R	√	√			
FFFFF344H	IIC clock selection register 0[Note 2]	IICCL0	R/W	√	√			
FFFFF346H	Slave address register 0[Note 2]	SVA0			√			
FFFFF348H	IIC shift register 0[Note 2]	IIC0			√			
FFFFF34AH	IIC function expansion register 0[Note 2]	IICX0		√	√			
FFFFF34CH	IIC clock expansion register 0[Note 2]	IICCE0			√			
FFFFF350H	IIC control register 1[Note 2]	IICC1		√	√			
FFFFF352H	IIC state register 1[Note 2]	IICS1	R	√	√			
FFFFF354H	IIC clock selection register 1[Note 2]	IICCL1	R/W	√	√			
FFFFF356H	Slave address register 1[Note 2]	SVA1			√			
FFFFF358H	IIC shift register 1[Note 2]	IIC1			√			
FFFFF35AH	IIC function expansion register 1[Note 2]	IICX1		√	√			
FFFFF35CH	IIC clock expansion register 1[Note 2]	IICCE1			√			
FFFFF360H	Watch timer mode register	WTNM		√	√			
FFFFF364H	Watch timer clock selection register	WTNCS			√			

Notes 1. Available only in the BY and HY versions (products with on-chip I^2C).

2. Available only in the Y versions (products with on-chip I^2C).

(7/7)

Address	Function Register Name		Symbol	R/W	Bit Units for Manipulation				After Reset
					1 Bit	8 Bits	16 Bits	32 Bits	
★ FFFFF366H	Watch timer high-speed clock selection register[Note 1]		WTNHC	R/W		√			00H
FFFFF36CH	Correction control register		CORCN		√	√			
FFFFF36EH	Correction request register		CORRQ		√	√			
FFFFF370H	Correction address register 0		CORAD0					√	00000000H
FFFFF374H	Correction address register 1		CORAD1					√	
FFFFF378H	Correction address register 2		CORAD2					√	
FFFFF37CH	Correction address register 3		CORAD3					√	
FFFFF380H	Oscillation stable time selection register		OSTS			√			04H
FFFFF382H	Watchdog timer clock selection register		WDCS			√			00H
FFFFF384H	Watchdog timer mode register		WDTM		√	√			
FFFFF38EH	DMA start factor expansion register		DMAS		√	√			
FFFFF3A0H	Real-time output buffer register L		RTBL			√			
FFFFF3A2H	Real-time output buffer register H		RTBH			√			
FFFFF3A4H	Real-time output port mode register		RTPM		√	√			
FFFFF3A6H	Real-time output port control register		RTPC		√	√			
FFFFF3C0H	A/D converter mode register 1		ADM1		√	√			
FFFFF3C2H	Analog input channel specification register		ADS		√	√			
FFFFF3C4H	A/D conversion result register		ADCR	R			√		0000H
FFFFF3C6H	A/D conversion result register H (higher 8 bits)		ADCRH			√			00H
FFFFF3C8H	A/D converter mode register 2		ADM2	R/W	√	√			
FFFFF3D0H	Key return mode register		KRM		√	√			
FFFFF3D4H	Noise elimination control register		NCC			√			
★ FFFFF3DEH	IEBus high-speed clock selection register	V850/SB2	IEHCLK[Note 2]			√			
FFFFF3E0H	IEBus control register	V850/SB2	BCR		√	√			
FFFFF3E2H	IEBus unit address register	V850/SB2	UAR				√		0000H
FFFFF3E4H	IEBus slave address register	V850/SB2	SAR				√		
FFFFF3E6H	IEBus partner address register	V850/SB2	PAR	R			√		
FFFFF3E8H	IEBus control data register	V850/SB2	CDR	R/W		√			01H
FFFFF3EAH	IEBus telegraph length register	V850/SB2	DLR			√			
FFFFF3ECH	IEBus data register	V850/SB2	DR			√			00H
FFFFF3EEH	IEBus unit status register	V850/SB2	USR	R	√	√			
FFFFF3F0H	IEBus interrupt status register	V850/SB2	ISR	R/W	√	√			
FFFFF3F2H	IEBus slave status register	V850/SB2	SSR	R	√	√			41H
FFFFF3F4H	IEBus communication success counter	V850/SB2	SCR			√			01H
FFFFF3F6H	IEBus transfer counter	V850/SB2	CCR			√			20H
FFFFF3F8H	IEBus clock selection register	V850/SB2	IECLK	R/W		√			00H

Notes 1. Available only in the B versions of the V850/SB1 and H versions of the V850/SB2.

2. Available only in the H versions of the V850/SB2.

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 store operations occur, it is notified by the system status register (SYS). The V850/SB1 and V850/SB2 have two specific registers, the power save control register (PSC) and processor clock control register (PCC). For details of the PSC register, refer to **6.3.1 (2) Power save control register (PSC),** and for details of the PCC register, refer to **6.3.1 (1) Processor clock control register (PCC).**

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 (by 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 the NOP instructions (5 instructions).
<7> 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 accepted 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 acceptance 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
         :
         :
```

Remark The above example assumes that rD (PCC set value), rX (value to be written to PSW), and rY (value rewritten to PSW) are already set.

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.

★ **2. Always stop DMA prior to accessing specific registers.**
3. If data is set to the PSC register to set IDLE mode or STOP mode, a dummy instruction needs to be inserted for correct execution of the routine after IDLE or STOP mode is released. For details, refer to 6.6 Cautions on Power Save Function.

(1) Command register (PRCMD)

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

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

Occurrence of illegal store 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 does not occur
1	Protection error occurs

Operation conditions of PRERR flag are shown as follows.

(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 apart from specific registers.

(b) Reset conditions: (PRERR = 0)

(1) When 0 is written to the PRERR flag of the SYS register. However, except for the case of **Remark 1**.

(2) At system reset.

★ **Remarks 1.** If 0 is written to the PRERR bit immediately after a write operation to the PRCMD register, the PRERR bit is set to 1 (because the SYS register is not a specific register).

2. If the PRCMD register is written again immediately after a write operation to the PRCMD register, the PRERR bit of the SYS register is set to 1 (because the SYS register is not a specific register).

CHAPTER 4 BUS CONTROL FUNCTION

The V850/SB1 and V850/SB2 are 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 $\overline{\text{WAIT}}$ pin input
- Idle state insertion function
- Bus mastership arbitration 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 11 (P110)

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 **(2) Memory address output mode register (MAM).**

Caution For debugging using the separate bus, refer to the user's manual of the corresponding in-circuit emulator.

4.2.2 Control register

(1) System control register (SYC)

This register switches control signals for the 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	$\overline{\text{DSTB}}$, R/$\overline{\text{W}}$, $\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$ Note signal output
1	$\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{UBEN}}$ Note signal output

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

★ **Caution In the V850/SB1 and V850/SB2, when using port 9 as an I/O port, set the BIC bit of the system control register (SYC) to 0. Note that the BIC bit is 0 after system reset.**

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

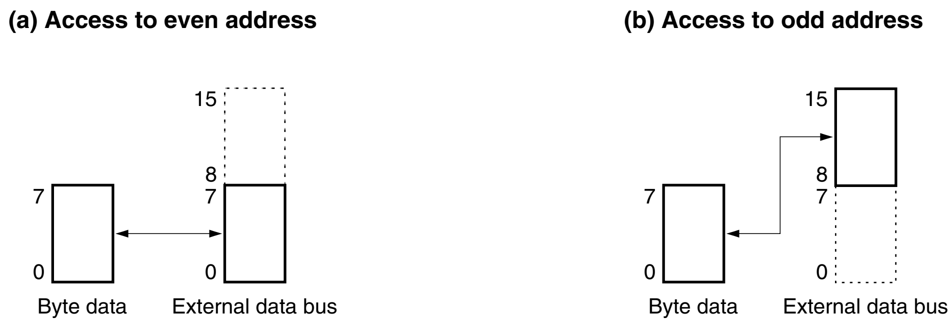
4.3.2 Bus width

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.

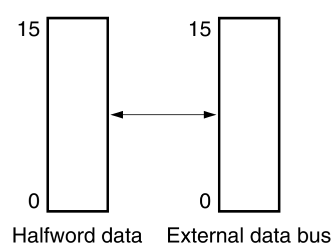
Figure 4-1. Byte Access (8 Bits)



(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.

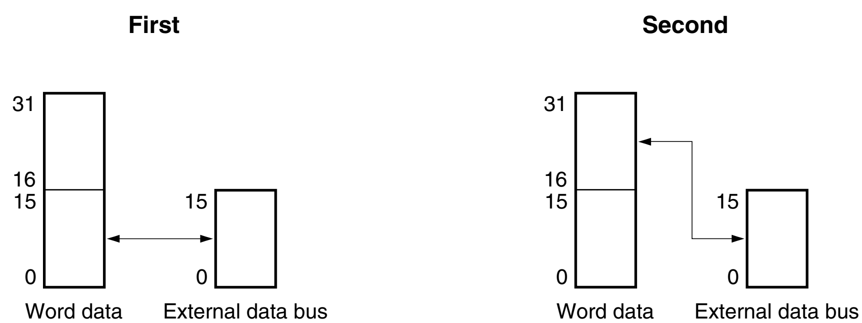
Figure 4-2. Halfword Access (16 Bits)



(3) Word access (32 bits)

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

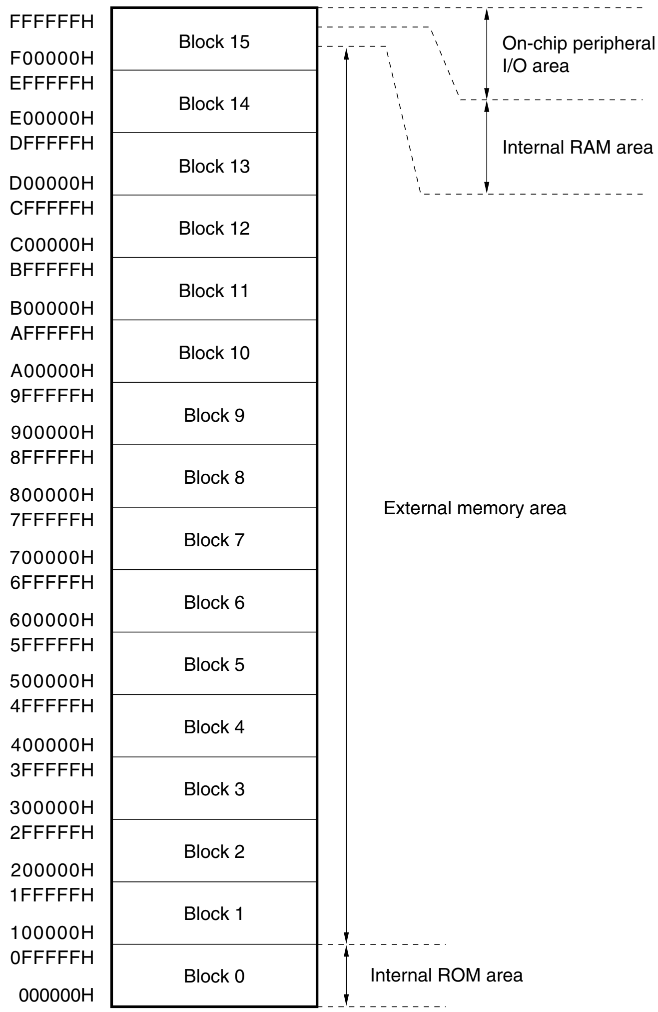
Figure 4-3. Word Access (32 Bits)



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 Block



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 the 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 a 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.

Caution Because the A1 pin and $\overline{\text{WAIT}}$ pin are alternate-function pins, the wait function by the $\overline{\text{WAIT}}$ pin cannot be used when using a separate bus (programmable wait can be used, however). Similarly, a separate bus cannot be used when the wait function by the $\overline{\text{WAIT}}$ pin is being used.

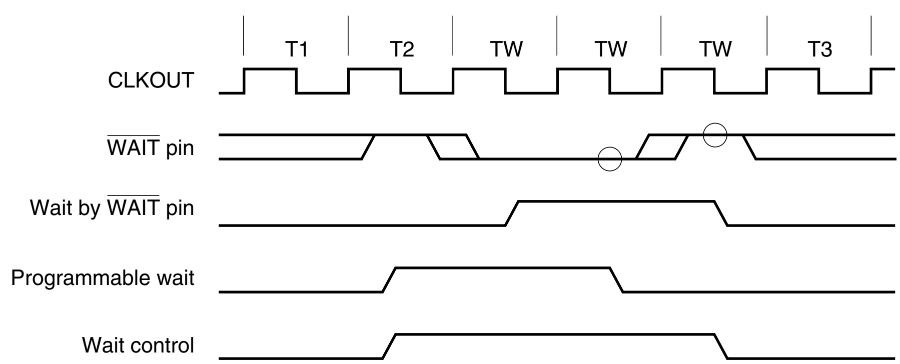
4.5.3 Relationship between programmable wait and external wait

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 has the greater 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; therefore, 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.

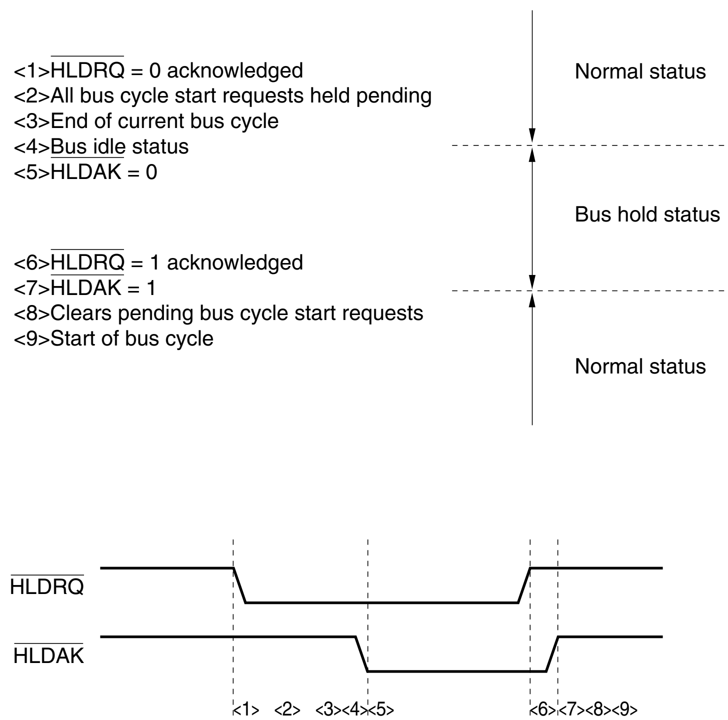
A bus hold request is not acknowledged between the first and the second word access, and between the read access and write access of the read-modify-write access executed using a bit manipulation instruction.

Note A1 to A15 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

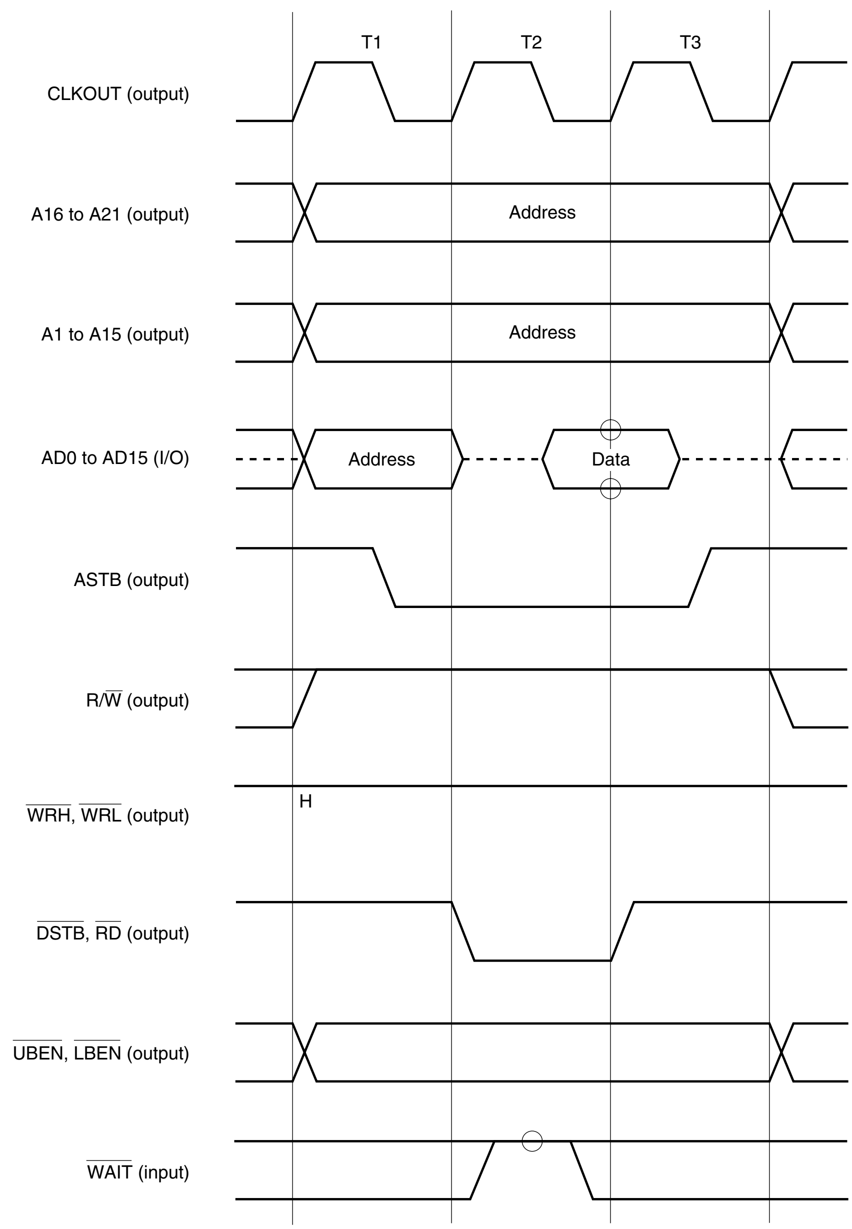
The V850/SB1 and V850/SB2 can execute read/write control for an external device using the following two modes.

- 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) (see **4.2.2 (1) System control register (SYC)**).

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

(a) 0 waits

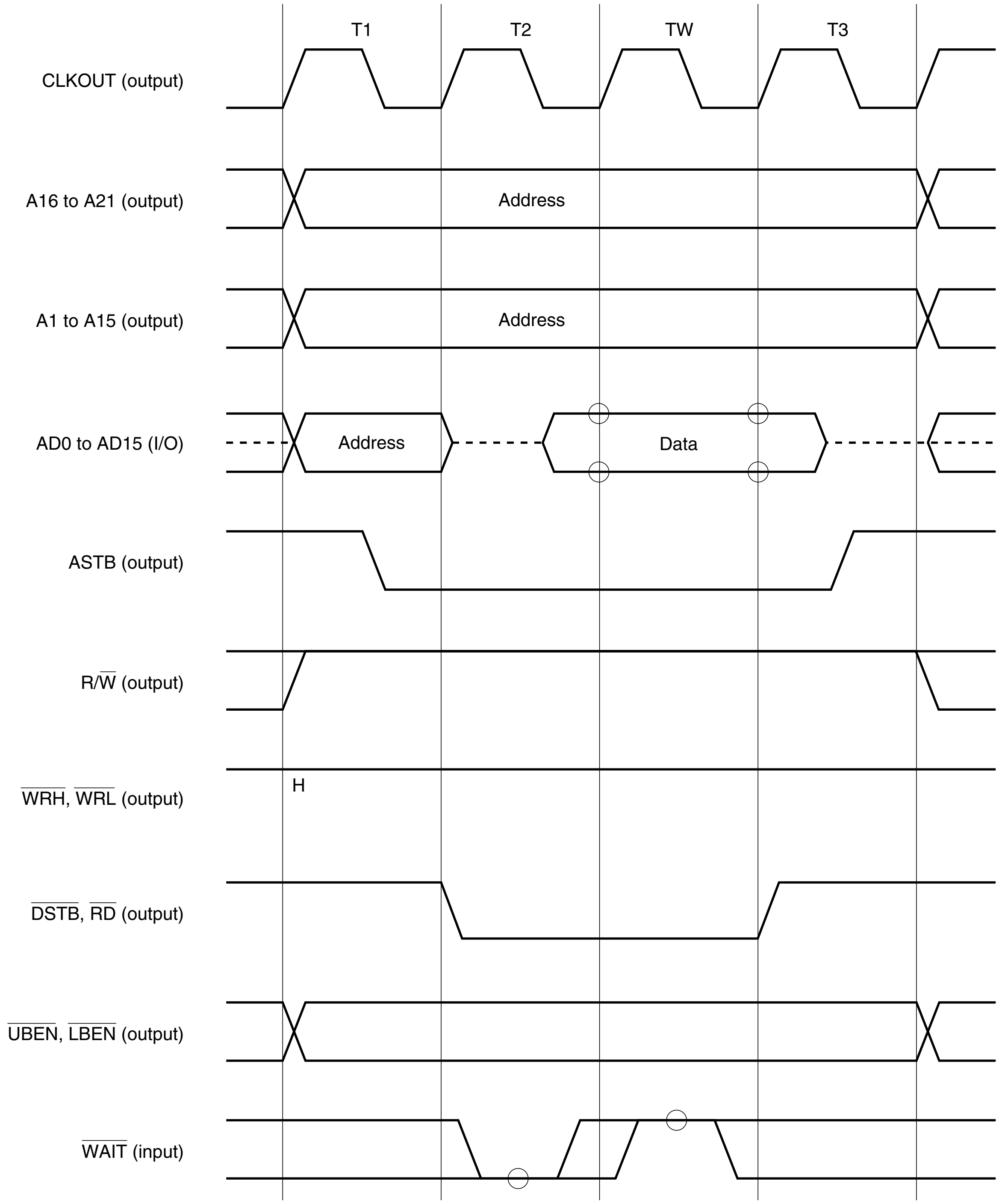


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 0.

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

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

(b) 1 wait

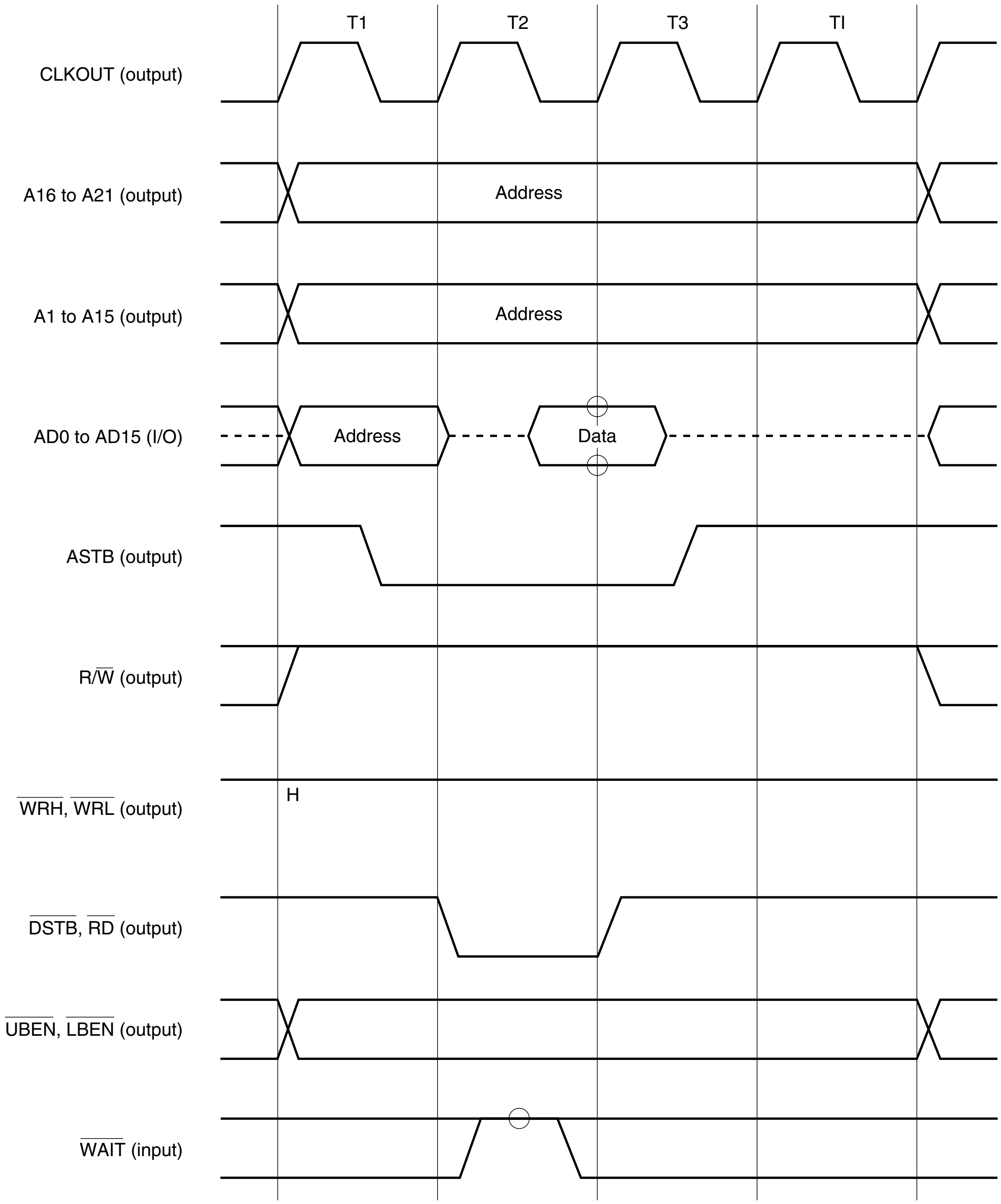


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 1.

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

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

(c) 0 waits, idle state

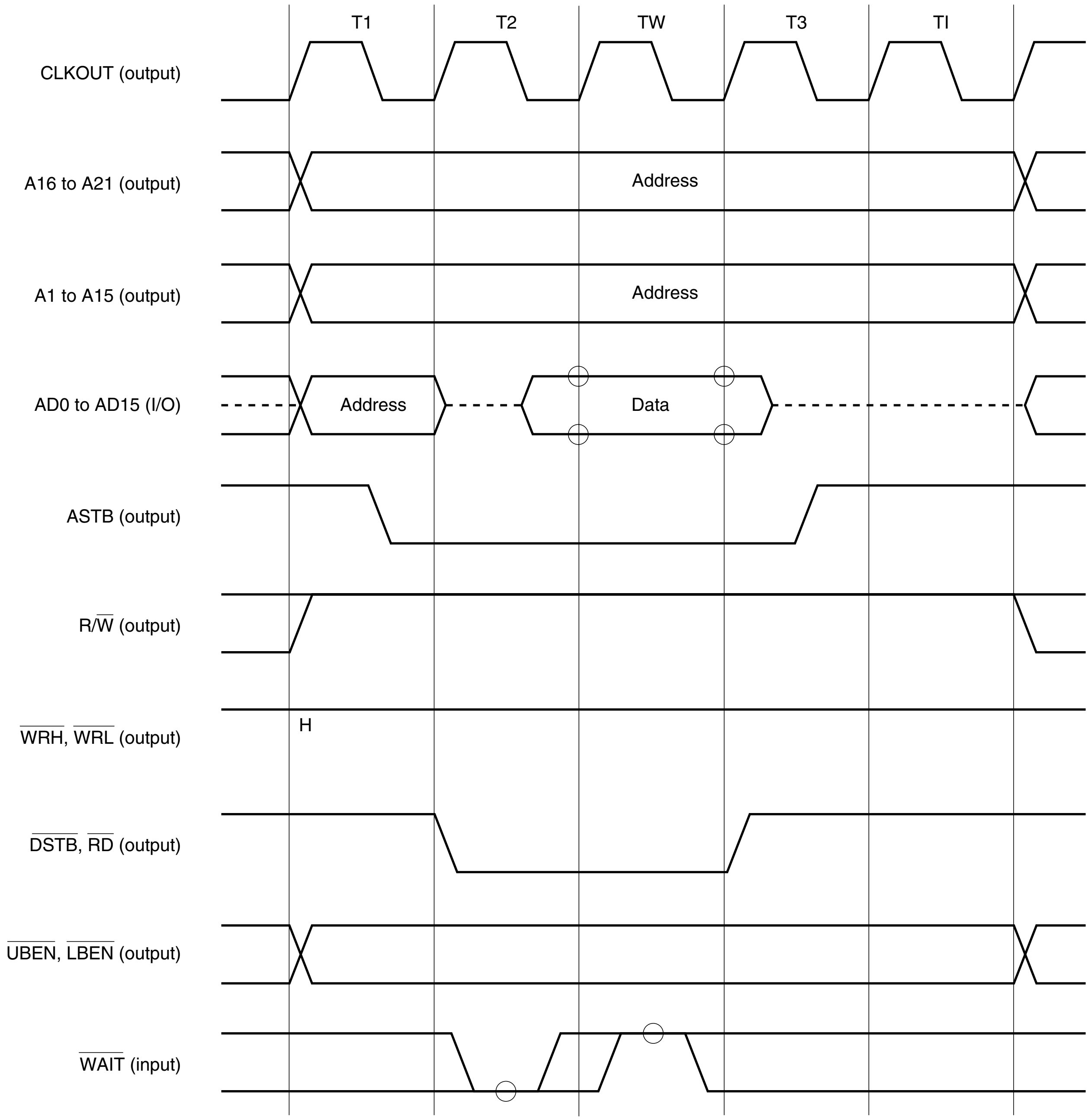


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 0.

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

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

(d) 1 wait, idle state

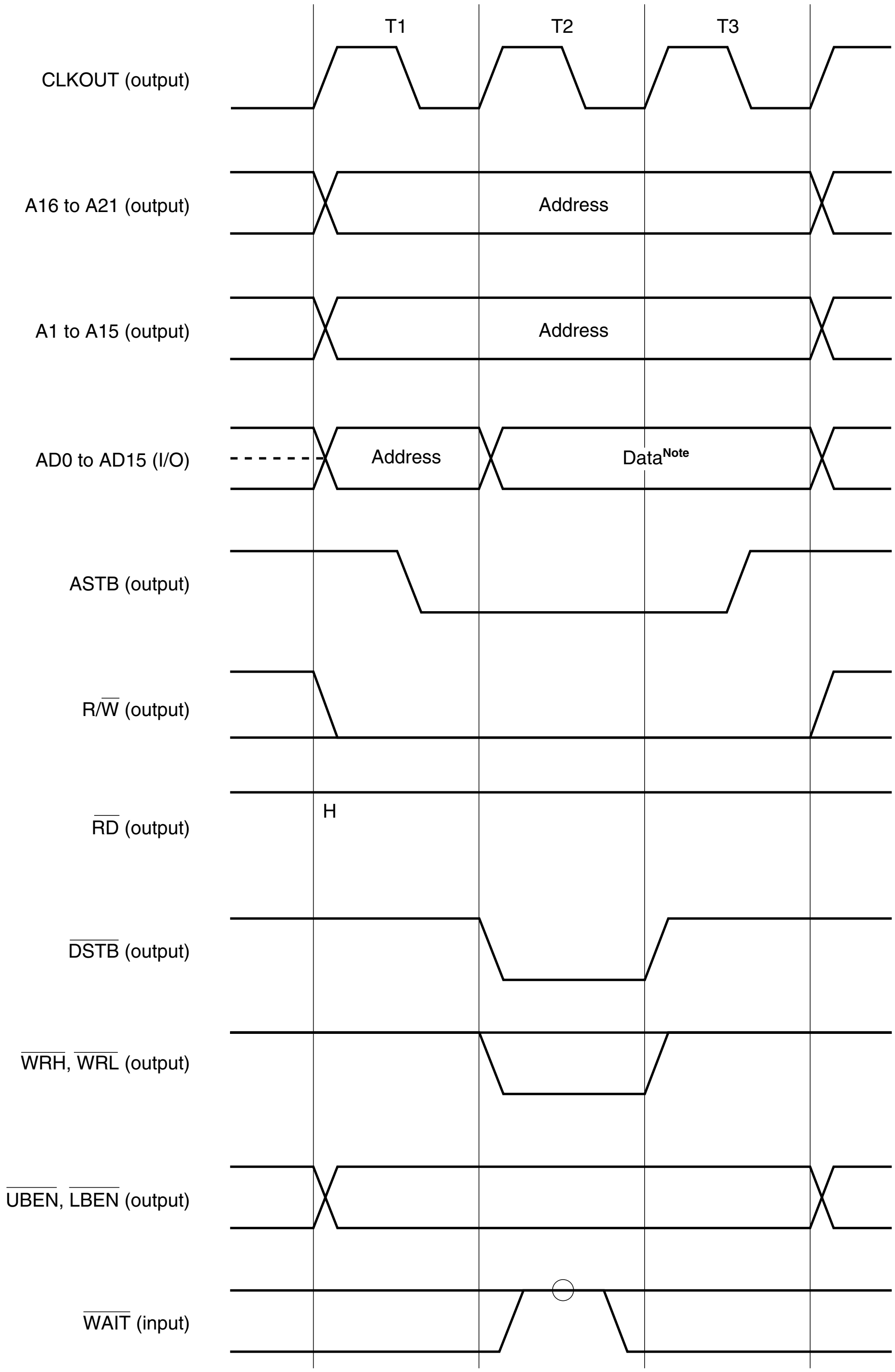


Remarks 1. ○ indicates the sampling timing when the number of programmable waits is set to 1.

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

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

(a) 0 waits



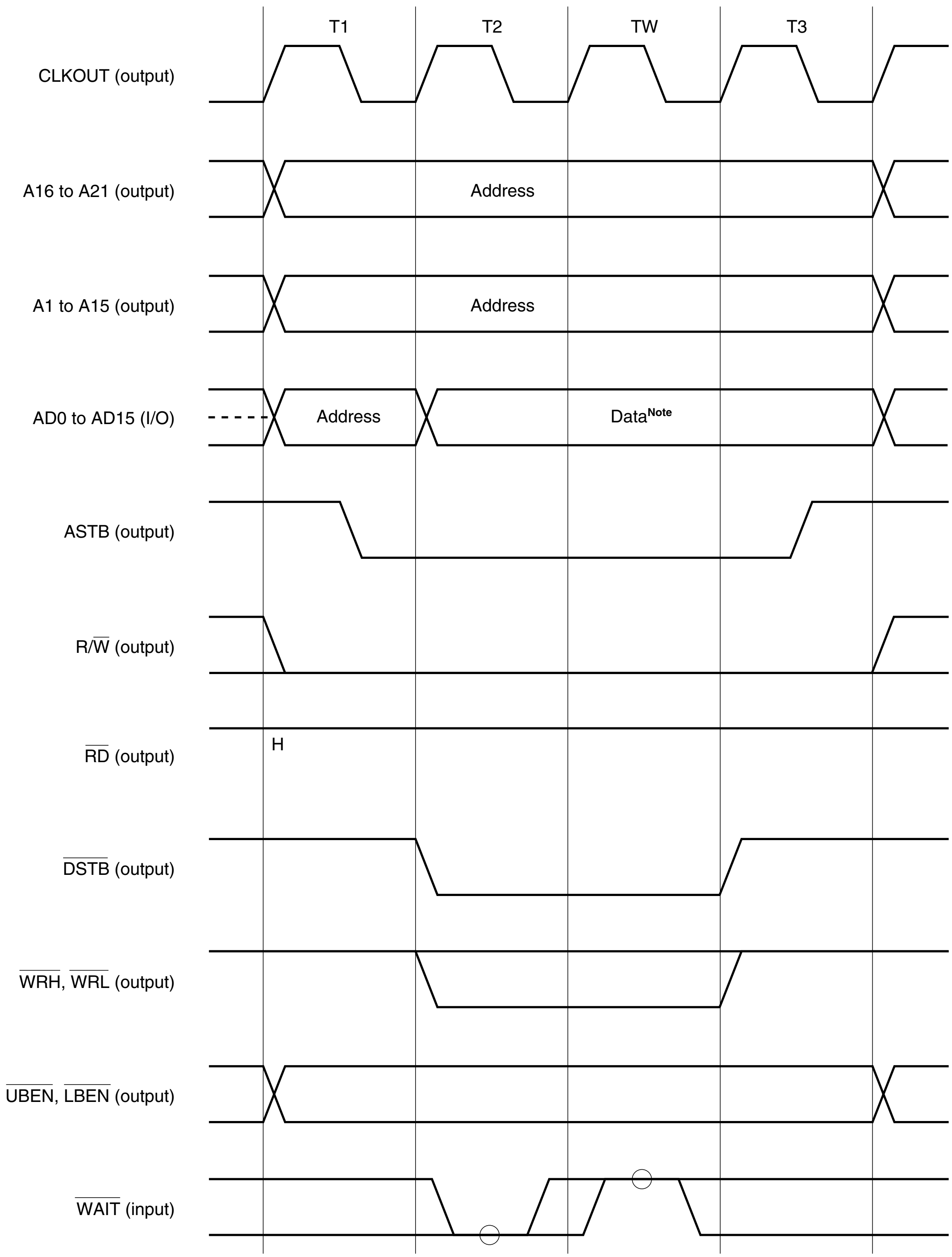
Note AD0 to AD7 output invalid data when odd-address byte data is accessed.
AD8 to AD15 output invalid data when even-address byte data is accessed.

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

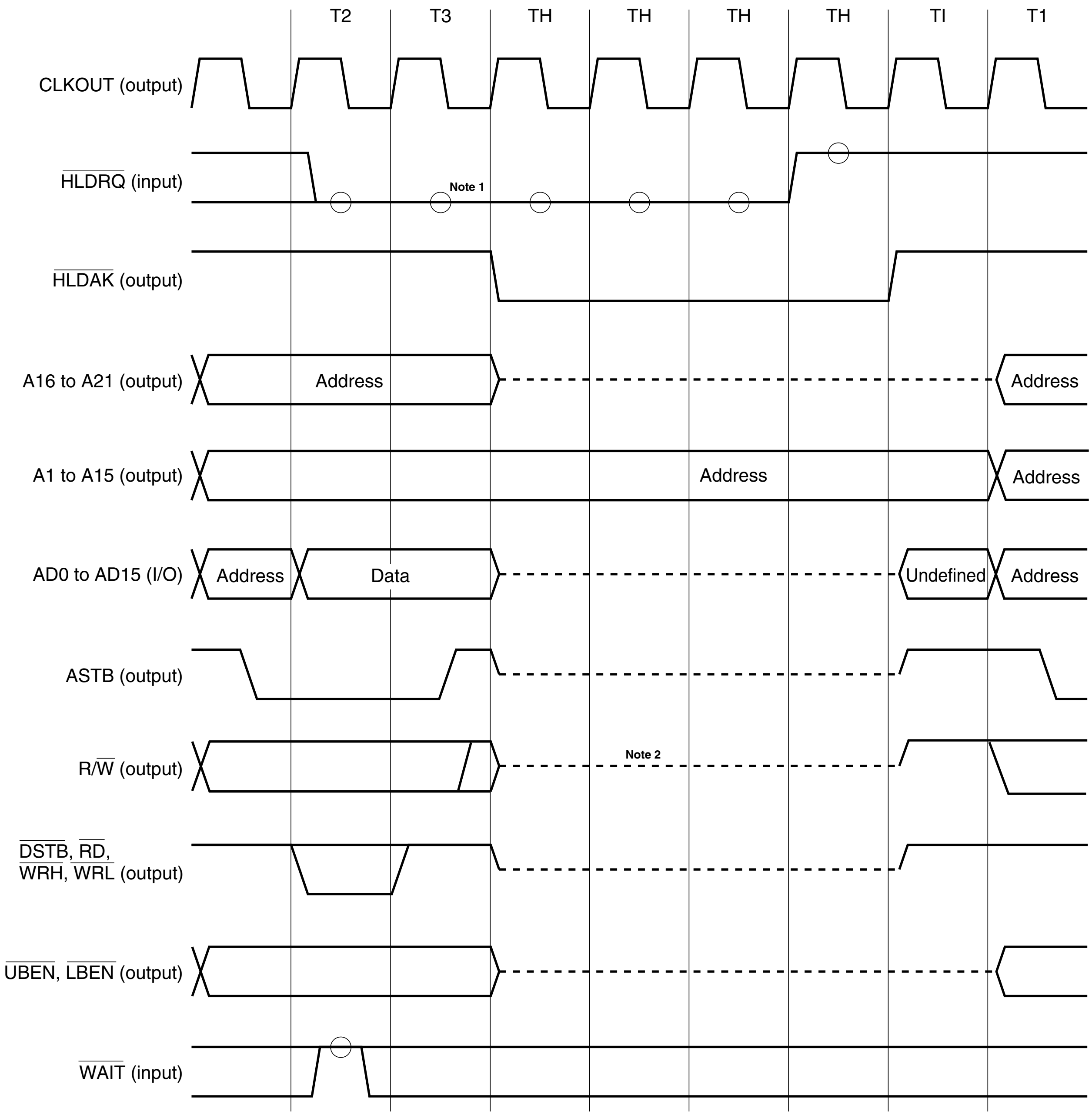


Note AD0 to AD7 output invalid data when odd-address byte data is accessed.
AD8 to AD15 output invalid data when even-address byte data is accessed.

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 the 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”) boundary 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/SB1 and V850/SB2 are provided with a dedicated interrupt controller (INTC) for interrupt servicing and realize a high-powered interrupt function that can service interrupt requests from a total of 37 to 40 sources.

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

The V850/SB1 and V850/SB2 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
  - Non-maskable interrupts: 2 sources
  - ★ Maskable interrupts: (the number of maskable interrupt sources differs depending on the product)

    (V850/SB1)

    μPD703030B, 703031A, 703031B, 703032A, 703032B, 703033A, 703033B, 70F3030B, 70F3032A, 70F3032B, 70F3033A, 70F3033B: 37 sources

    μPD703030BY, 703031AY, 703031BY, 703032AY, 703032BY, 703033AY, 703033BY, 70F3030BY, 70F3032AY, 70F3032BY, 70F3033AY, 70F3033BY: 38 sources

    (V850/SB2)

    μPD703034A, 703034B, 703035A, 703035B, 703036H, 703037A, 703037H, 70F3035A, 70F3035B, 70F3036H, 70F3037A, 70F3037H: 39 sources

    μPD703034AY, 703034BY, 703035AY, 703035BY, 703036HY, 703037AY, 703037HY, 70F3035AY, 70F3035BY, 70F3036HY, 70F3037AY, 70F3037HY: 40 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.
- 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	–	0010H	00000010H	nextPC	–
	Interrupt	–	INTWDT	WDTOVF non-maskable	WDT	0020H	00000020H	nextPC	–
Software exception	Exception	–	TRAP0n[Note 1]	TRAP instruction	–	004nH[Note 1]	00000040H	nextPC	–
	Exception	–	TRAP1n[Note 1]	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	INTWTNI	Watch timer prescaler	WT	0140H	00000140H	nextPC	WTNIIC
		9	INTTM00	INTTM00	TM0	0150H	00000150H	nextPC	TMIC00
		10	INTTM01	INTTM01	TM0	0160H	00000160H	nextPC	TMIC01
		11	INTTM10	INTTM10	TM1	0170H	00000170H	nextPC	TMIC10
		12	INTTM11	INTTM11	TM1	0180H	00000180H	nextPC	TMIC11
		13	INTTM2	TM2 compare match/OVF	TM2	0190H	00000190H	nextPC	TMIC2
		14	INTTM3	TM3 compare match/OVF	TM3	01A0H	000001A0H	nextPC	TMIC3
		15	INTTM4	TM4 compare match/OVF	TM4	01B0H	000001B0H	nextPC	TMIC4
		16	INTTM5	TM5 compare match/OVF	TM5	01C0H	000001C0H	nextPC	TMIC5
		17	INTTM6	TM6 compare match/OVF	TM6	01D0H	000001D0H	nextPC	TMIC6
		18	INTTM7	TM7 compare match/OVF	TM7	01E0H	000001E0H	nextPC	TMIC7
		19	INTIIC0[Note 2]/ INTCSI0	I^2C interrupt/ CSI0 transmit end	I^2C/ CSI0	01F0H	000001F0H	nextPC	CSIC0
		20	INTSER0	UART0 serial error	UART0	0200H	00000200H	nextPC	SERIC0
		21	INTSR0/ INTCSI1	UART0 receive end/ CSI1 transmit end	UART0/ CSI1	0210H	00000210H	nextPC	CSIC1
		22	INTST0	UART0 transmit end	UART0	0220H	00000220H	nextPC	STIC0

Notes 1. n: 0 to FH

2. Available only in the Y versions (products with on-chip I^2C).

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	23	INTCSI2	CSI2 transmit end	CSI2	0230H	00000230H	nextPC	CSIC2
		24	INTIIC1 Note 1	I^2C1 interrupt	I^2C1	0240H	00000240H	nextPC	IICIC1
		25	INTSER1	UART1 serial error	UART1	0250H	00000250H	nextPC	SERIC1
		26	INTSR1/ INTCSI3	UART1 receive end/ CSI3 transmit end	UART1/ CSI3	0260H	00000260H	nextPC	CSIC3
		27	INTST1	UART1 transmit end	UART1	0270H	00000270H	nextPC	STIC1
		28	INTCSI4	CSI4 transmit end	CSI4	0280H	00000280H	nextPC	CSIC4
		29	INTIE1 Note 2	IEBus transfer end	IEBus	0290H	00000290H	nextPC	IEBIC1
		30	INTIE2 Note 2	IEBus communication end	IEBus	02A0H	000002A0H	nextPC	IEBIC2
		31	INTAD	A/D conversion end	A/D	02B0H	000002B0H	nextPC	ADIC
		32	INTDMA0	DMA0 transfer end	DMA0	02C0H	000002C0H	nextPC	DMAIC0
		33	INTDMA1	DMA1 transfer end	DMA1	02D0H	000002D0H	nextPC	DMAIC1
		34	INTDMA2	DMA2 transfer end	DMA2	02E0H	000002E0H	nextPC	DMAIC2
		35	INTDMA3	DMA3 transfer end	DMA3	02F0H	000002F0H	nextPC	DMAIC3
		36	INTDMA4	DMA4 transfer end	DMA4	0300H	00000300H	nextPC	DMAIC4
		37	INTDMA5	DMA5 transfer end	DMA5	0310H	00000310H	nextPC	DMAIC5
		38	INTWTN	Watch timer OVF	WT	0320H	00000320H	nextPC	WTNIC
		39	INTKR	Key return interrupt	KR	0330H	00000330H	nextPC	KRIC

Notes 1. Available only in the Y versions (products with on-chip I^2C).

2. Available only in the V850/SB2.

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. Non-maskable interrupts (INTWDT) and maskable interrupts (INTWDTM) are set by the WDTM4 bit of the watchdog timer mode register (WDTM).

5.2 Non-Maskable Interrupt

A non-maskable interrupt is acknowledged unconditionally, even when interrupts are disabled (DI state). An NMI is not subject to priority control and takes precedence over all other interrupts.

The following two non-maskable interrupt requests are available in the V850/SB2.

- 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 acknowledgment of another non-maskable interrupt request is held pending. The pending NMI is acknowledged after the original service routine of the non-maskable interrupt under execution has been terminated (by the RETI instruction), or when PSW.NP is cleared to 0 by the LDSR 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 If PSW.NP is cleared to 0 by the LDSR instruction during non-maskable interrupt servicing, the interrupt afterwards cannot be acknowledged correctly.

5.2.1 Operation

If a non-maskable interrupt 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 and 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

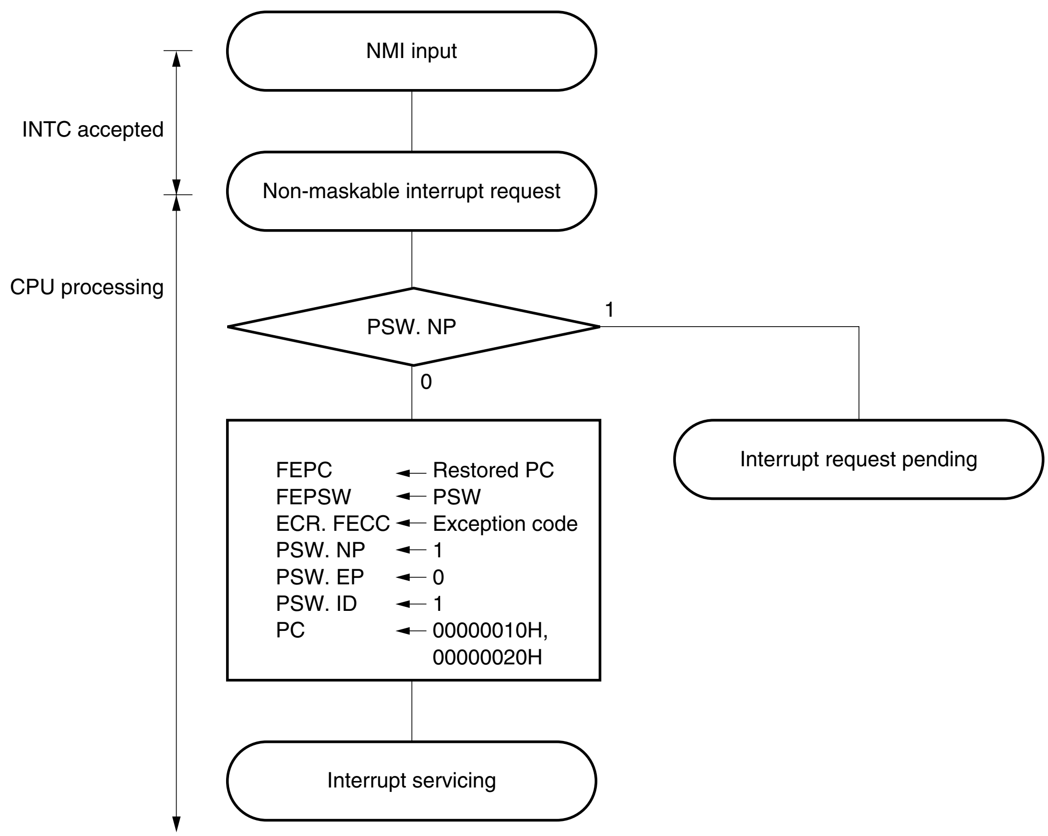
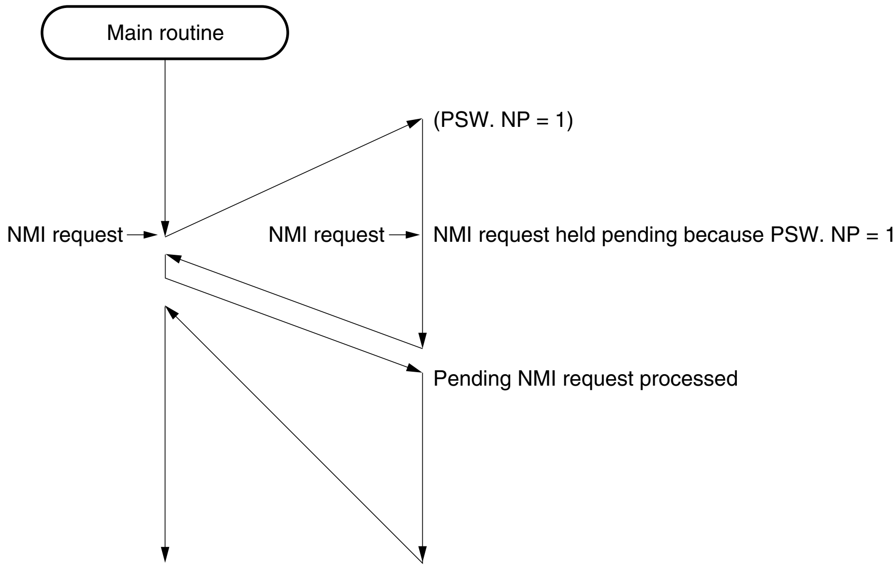
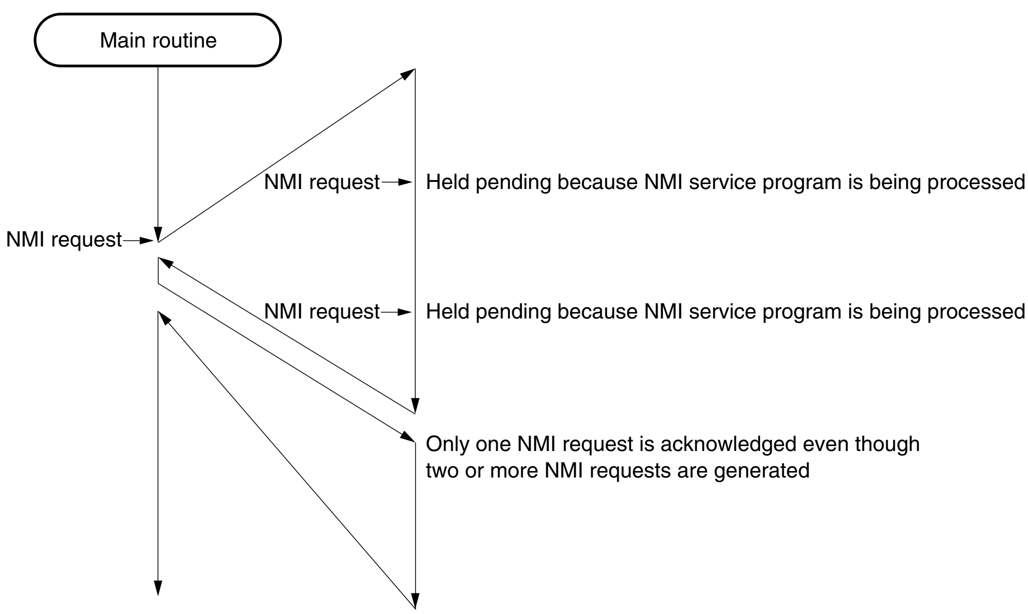


Figure 5-2. Acknowledging Non-Maskable Interrupt Request

(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.

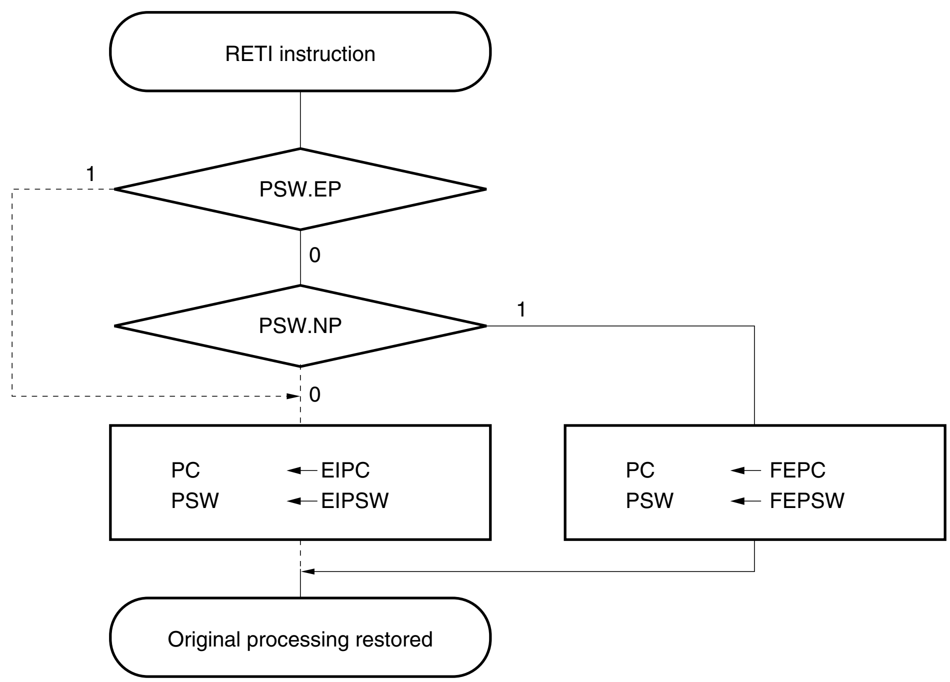
Operation of RETI instruction

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 the PSW is 0 and the NP bit of the PSW is 1.
(2) Transfers control back to the address of the restored PC and PSW.

How the RETI instruction is processed is shown below.

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.

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 eliminator of NMI pin

NMI pin noise is eliminated by the noise eliminator using analog delay. Therefore, a signal input to the NMI pin is not detected as an edge, unless it maintains its input level for a certain period. The edge is detected after a certain period has elapsed.

The NMI pin is 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.

5.2.5 Edge detection function of NMI pin

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

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

After reset, the valid edge of the NMI 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 P00 as an output port, set the NMI valid edge to “detect neither rising nor falling edge”.

(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/SB1 and V850/SB2 have 37 to 40 maskable interrupt sources (see **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 acknowledgment 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

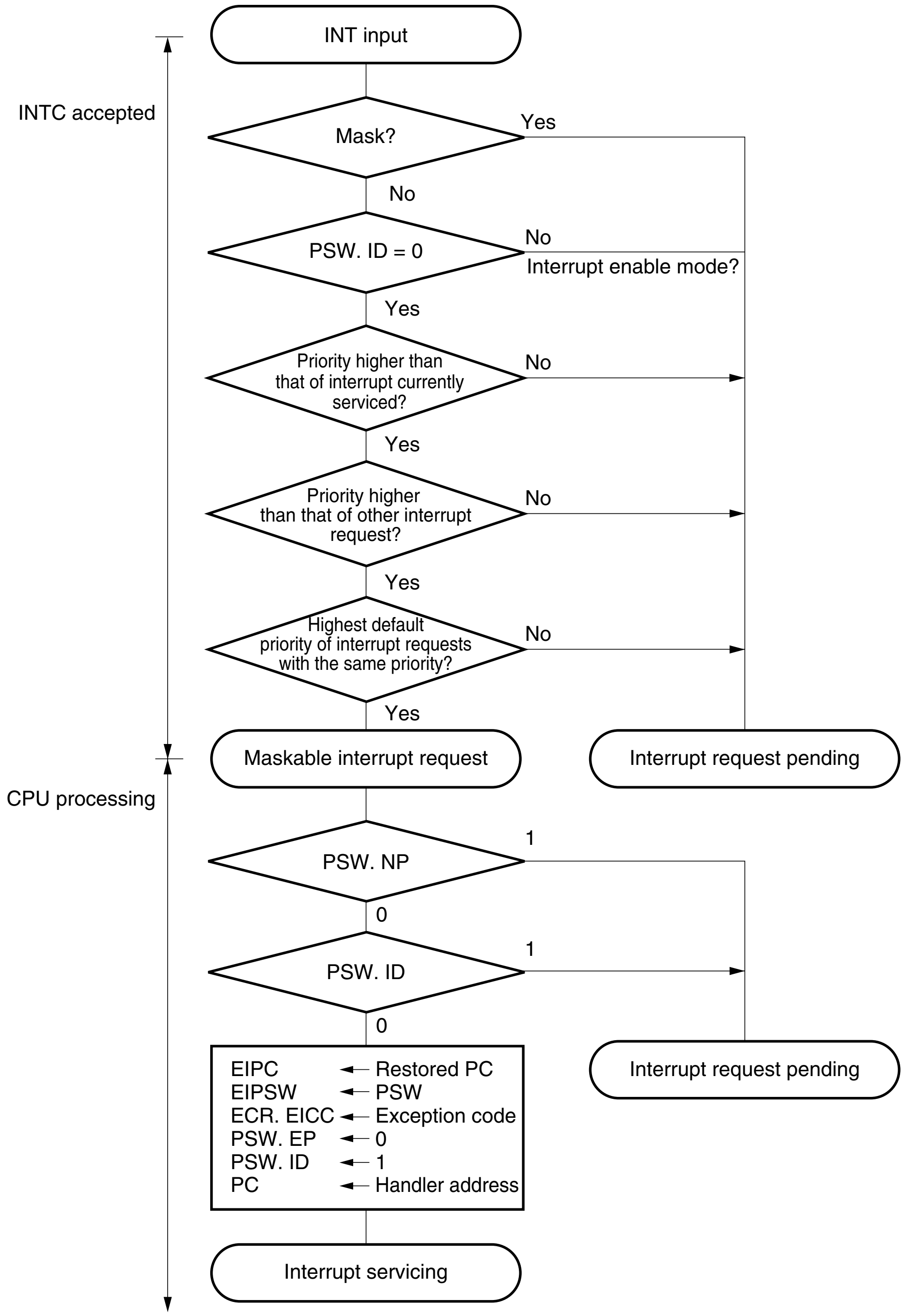
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine.

(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 5-4. Maskable Interrupt Servicing



5.3.2 Restore

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

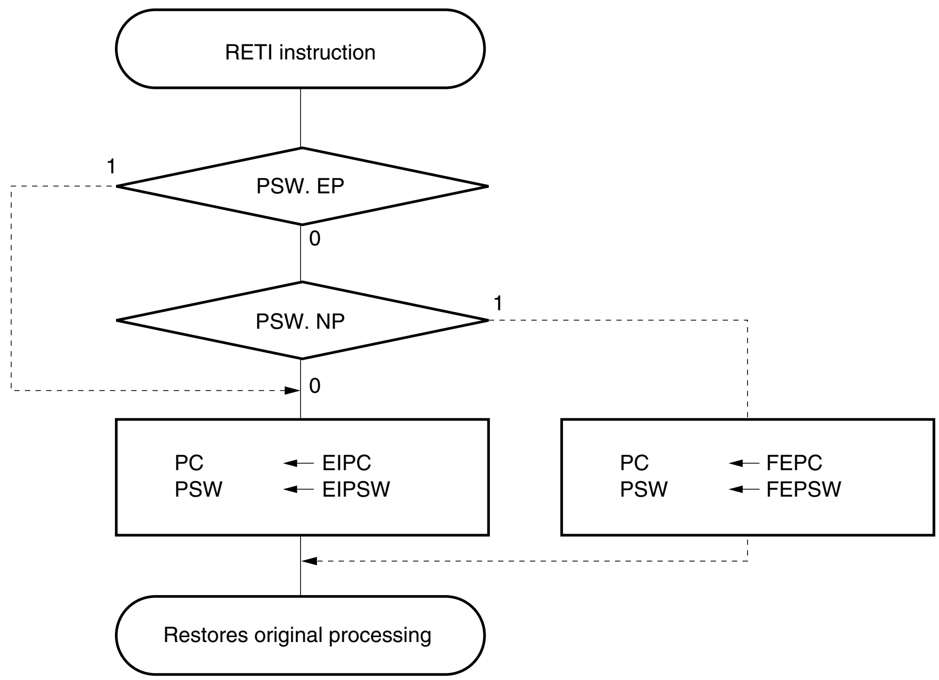
Operation of RETI instruction

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 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 5-5. 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

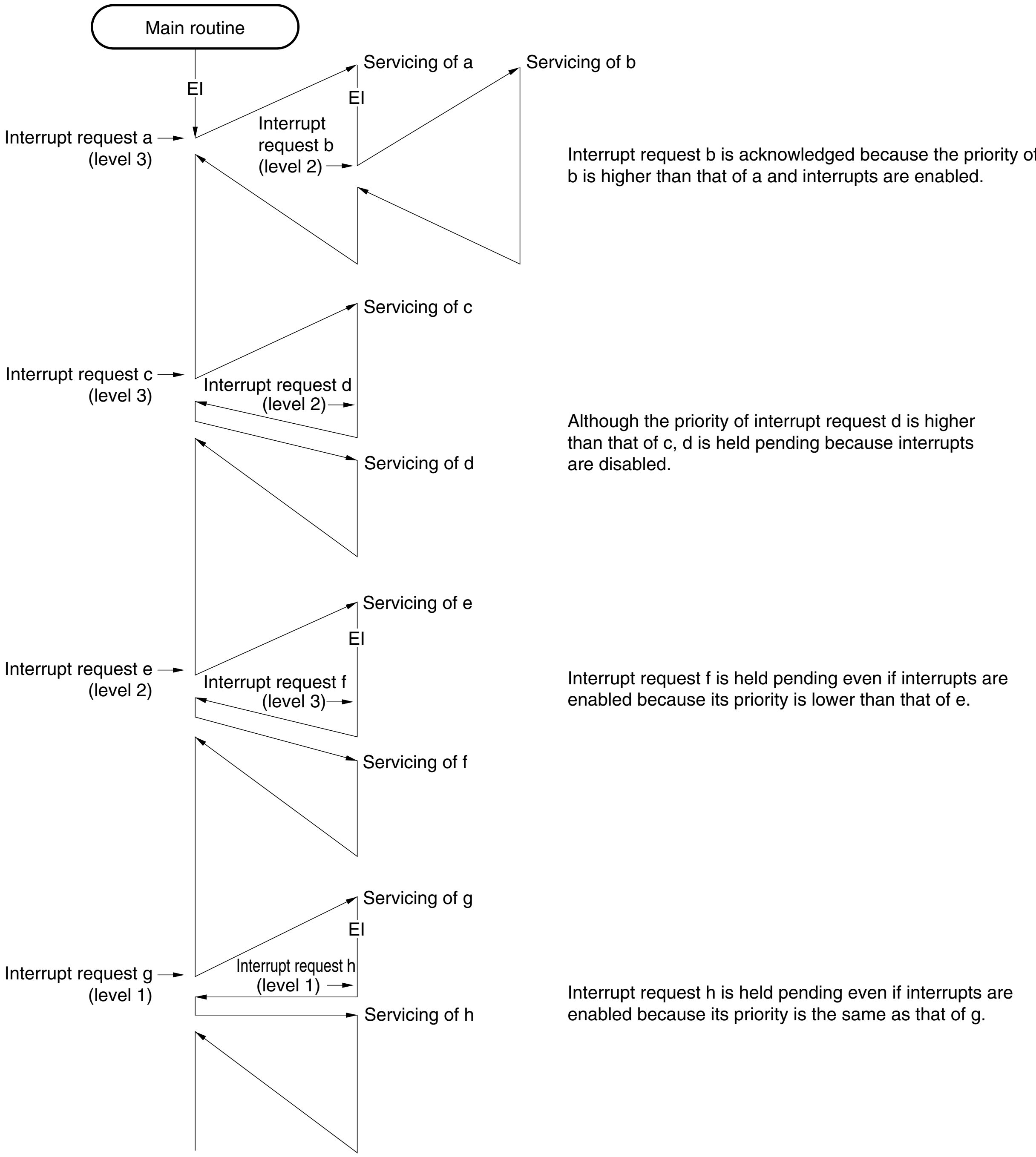
The V850/SB1 and V850/SB2 provide multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels 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**. 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 service 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 5-6. Example of Multiple Interrupt Servicing (1/2)

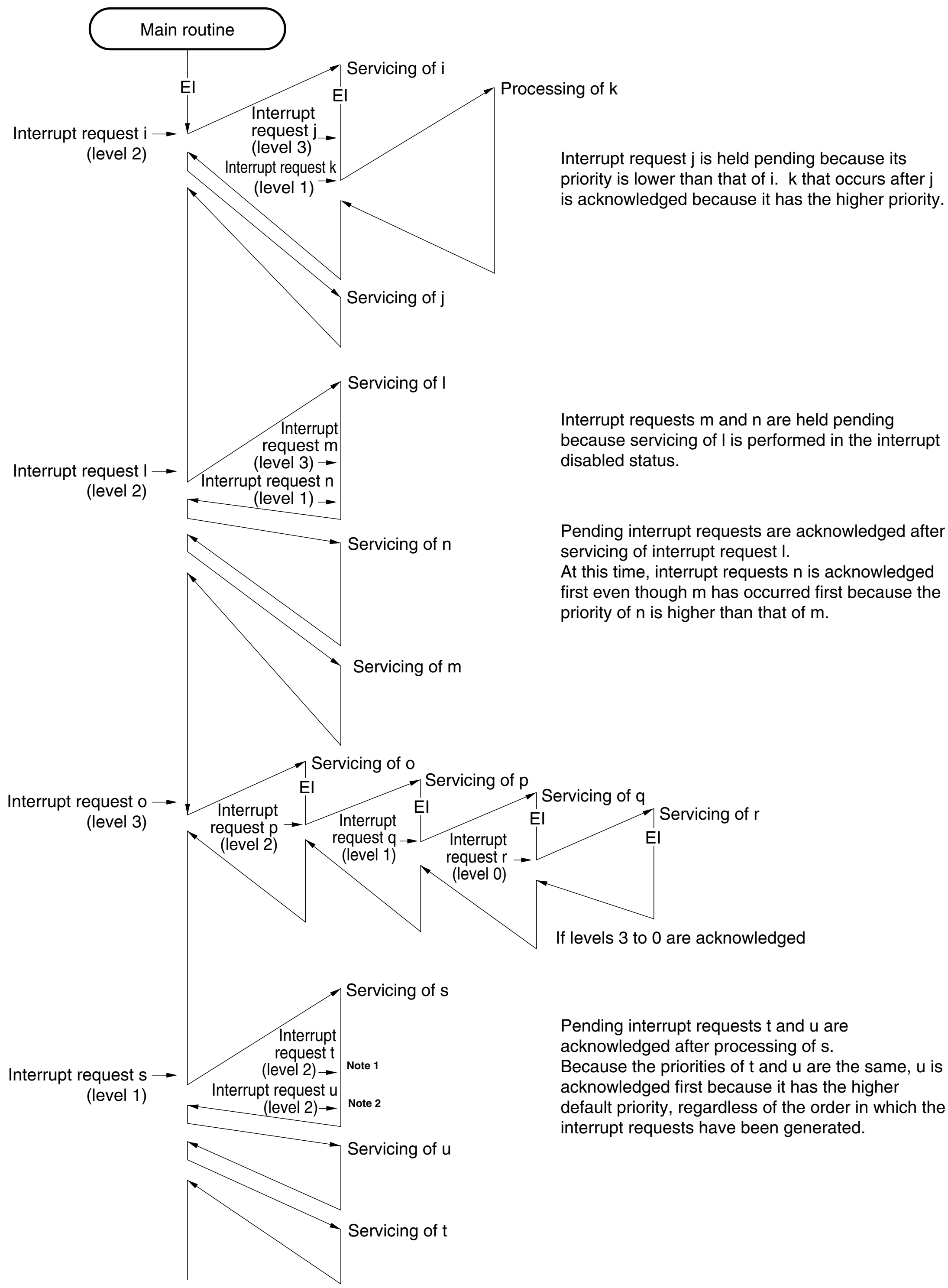


Caution The values of EIPC and EIPSW must be saved before executing multiple interrupts.

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 figure indicates the relative priority between two interrupt requests.

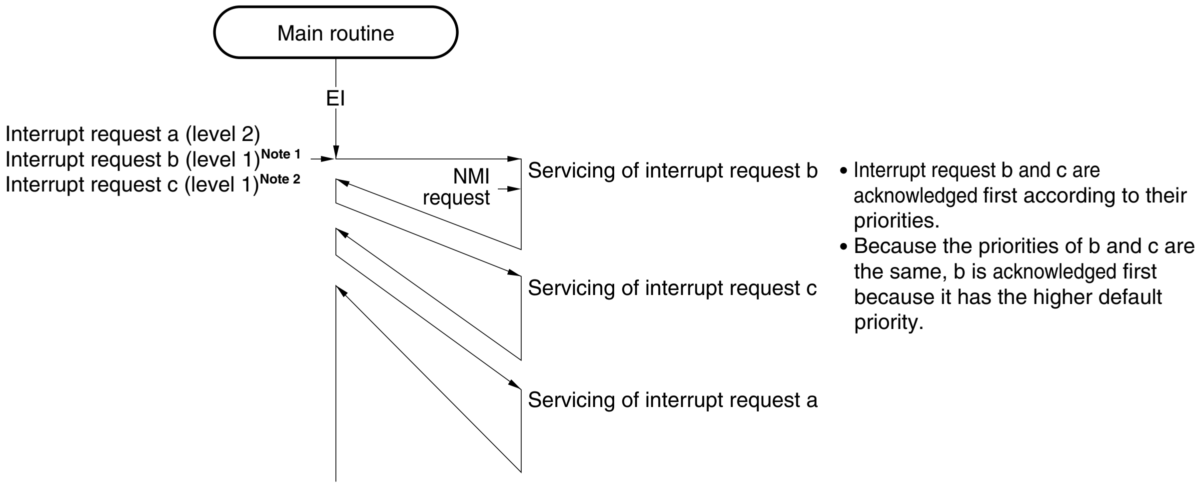
Figure 5-6. Example of Multiple Interrupt Servicing (2/2)



Notes 1. Lower default priority

2. Higher default priority

Figure 5-7. 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 is executed twice. However, when DMA is not used, interrupt servicing is not executed twice.

- **Execution of a bit manipulation instruction corresponding to the interrupt request flag (xxIFn)**
- **An interrupt via hardware of the same interrupt control register (xxICn) as the interrupt request flag (xxIFn) is generated**
- **DMA is started during execution of a bit manipulation instruction corresponding to the interrupt request flag (xxIFn)**

Two workarounds using software are shown below.

- **Insert a DI instruction before the software-based bit manipulation instruction and an EI instruction after it, so that jumping to an interrupt immediately after the bit manipulation instruction execution does not occur.**
- **When an interrupt request is acknowledged, since the hardware becomes interrupt disabled (DI state), 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 acknowledgment timing and the bit read timing conflict.**

After reset: 47H R/W Address: FFFFF100H to FFFFF156H

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 (see **Table 5-2**)
n: Peripheral unit number (see **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
FFFFF118H	WTNIIC	WTNIIF	WTNIMK	0	0	0	WTNIPR2	WTNIPR1	WTNIPR0
FFFFF11AH	TMIC00	TMIF00	TMMK00	0	0	0	TMPR002	TMPR001	TMPR000
FFFFF11CH	TMIC01	TMIF01	TMMK01	0	0	0	TMPR012	TMPR011	TMPR010
FFFFF11EH	TMIC10	TMIF10	TMMK10	0	0	0	TMPR102	TMPR101	TMPR100
FFFFF120H	TMIC11	TMIF11	TMMK11	0	0	0	TMPR112	TMPR111	TMPR110
FFFFF122H	TMIC2	TMIF2	TMMK2	0	0	0	TMPR22	TMPR21	TMPR20
FFFFF124H	TMIC3	TMIF3	TMMK3	0	0	0	TMPR32	TMPR31	TMPR30
FFFFF126H	TMIC4	TMIF4	TMMK4	0	0	0	TMPR42	TMPR41	TMPR40
FFFFF128H	TMIC5	TMIF5	TMMK5	0	0	0	TMPR52	TMPR51	TMPR50
FFFFF12AH	TMIC6	TMIF6	TMMK6	0	0	0	TMPR62	TMPR61	TMPR60
FFFFF12CH	TMIC7	TMIF7	TMMK7	0	0	0	TMPR72	TMPR71	TMPR70
FFFFF12EH	CSIC0	CSIF0	CSMK0	0	0	0	CSPR02	CSPR01	CSPR00
FFFFF130H	SERIC0	SERIF0	SERMK0	0	0	0	SERPR02	SERPR01	SERPR00
FFFFF132H	CSIC1	CSIF1	CSMK1	0	0	0	CSPR12	CSPR11	CSPR10
FFFFF134H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF136H	CSIC2	CSIF2	CSMK2	0	0	0	CSPR22	CSPR21	CSPR20
FFFFF138H	IICIC1 Note 1	IICIF1	IICMK1	0	0	0	IICPR12	IICPR11	IICPR10
FFFFF13AH	SERIC1	SERIF1	SERMK1	0	0	0	SERPR12	SERPR11	SERPR10
FFFFF13CH	CSIC3	CSIF3	CSMK3	0	0	0	CSPR32	CSPR31	CSPR30
FFFFF13EH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF140H	CSIC4	CSIF4	CSMK4	0	0	0	CSPR42	CSPR41	CSPR40
FFFFF142H	IEBIC1 Note 2	IEBIF1	IEBMK1	0	0	0	IEBPR12	IEBPR11	IEBPR10
FFFFF144H	IEBIC2 Note 2	IEBIF2	IEBMK2	0	0	0	IEBPR22	IEBPR21	IEBPR20
FFFFF146H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF148H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF14AH	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF14CH	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF14EH	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF150H	DMAIC4	DMAIF4	DMAMK4	0	0	0	DMAPR42	DMAPR41	DMAPR40
FFFFF152H	DMAIC5	DMAIF5	DMAMK5	0	0	0	DMAPR52	DMAPR51	DMAPR50
FFFFF154H	WTNIC	WTNIF	WTNMK	0	0	0	WTNPR2	WTNPR1	WTNPR0
FFFFF156H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0

Notes 1. Available only in the Y versions (products with on-chip I^2C).

2. Available only in the V850/SB2.

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 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 acknowledgment timing and the bit read 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.

After reset: 00000020H

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

ID	Specifies maskable interrupt servicing[Note]
0	Maskable interrupt acknowledgment enabled
1	Maskable interrupt acknowledgment disabled (pending)

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

The interrupt request generated during the acknowledgment 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.

5.3.8 Noise elimination

(1) Noise elimination of INTP0 to INTP3 pins

The INTP0 to INTP3 pins incorporate a noise eliminator that functions using analog delay. Therefore, a signal input to each pin is not detected as an edge, unless it maintains its input level for a certain period.
An edge is detected after a certain period has elapsed.

(2) Noise elimination of INTP4 and INTP5 pins

The INTP4 and INTP5 pins incorporate a digital noise eliminator. If the input level of the INTP pin is detected by the sampling clock (f_{XX}) and the same level is not detected three successive times, the input pulse is eliminated as a noise. 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 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.

(3) Noise elimination of INTP6 pin

The INTP6 pin incorporates a digital noise eliminator. The sampling clock for digital sampling can be selected from among f_{XX}, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$, $f_{XX}/1024$, and f_{XT}. Sampling is performed 3 times.

(a) Noise elimination control register (NCC)

The noise elimination control register (NCC) selects the clock to be used. Remote control signals can be received effectively with this function.

f_{XT} can be used for the noise elimination clock. In this case, the INTP6 external interrupt function is enabled in the IDLE/STOP mode.

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

Caution After the sampling clock has been changed, it takes 3 sampling clocks to initialize the noise eliminator. For that reason, if an INTP6 valid edge was input within these 3 clocks, an interrupt request may occur. Therefore, be careful of the following when using the interrupt and DMA functions.

- **When using the interrupt function, after 3 sampling clocks have elapsed, enable interrupts after the interrupt request flag (bit 7 of PIC6) has been cleared.**
- **When using the DMA function, after 3 sampling clocks have elapsed, enable DMA by setting bit 0 of DCHCn.**

After reset: 00H R/W Address: FFFFF3D4H

	7	6	5	4	3	2	1	0
NCC	0	0	0	0	0	NCS2	NCS1	NCS0

NCS2	NCS1	NCS0	Noise elimination clock	Reliably eliminated noise width[Note 1]	
				f_{XX} = 20 MHz[Note 2]	f_{XX} = 12.58 MHz
0	0	0	f_{XX}	100 ns	158 ns
0	0	1	$f_{XX}/64$	6.4 μs	10.1 μs
0	1	0	$f_{XX}/128$	12.8 μs	20.3 μs
0	1	1	$f_{XX}/256$	25.6 μs	40.6 μs
1	0	0	$f_{XX}/512$	51.2 μs	81.3 μs
1	0	1	$f_{XX}/1024$	102.4 μs	162.7 μs
1	1	0	Setting prohibited		
1	1	1	f_{XT}	61 μs	

Notes 1. Since sampling is preformed three times, the reliably eliminated noise width is 2 × noise elimination clock.

2. Only in the V850/SB1.

5.3.9 Edge detection function

The valid edges of the INTP0 to INTP6 pins can be selected for each pin from the following four types.

- Rising edge
- Falling edge
- Both rising and falling edges
- Neither edge

The validity of the rising edge is controlled by rising edge specification register 0 (EGP0), and the validity of the falling edge is controlled by falling edge specification register 0 (EGN0). Refer to **5.2.5 Edge detection function of NMI pin** for details of EGP0 and EGN0.

After reset, the valid edge of the NMI 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 P01 to P07 as output ports, set the valid edges of INTP0 to INTP6 to “detect neither rising nor falling edge” or mask the interrupt request.

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.**

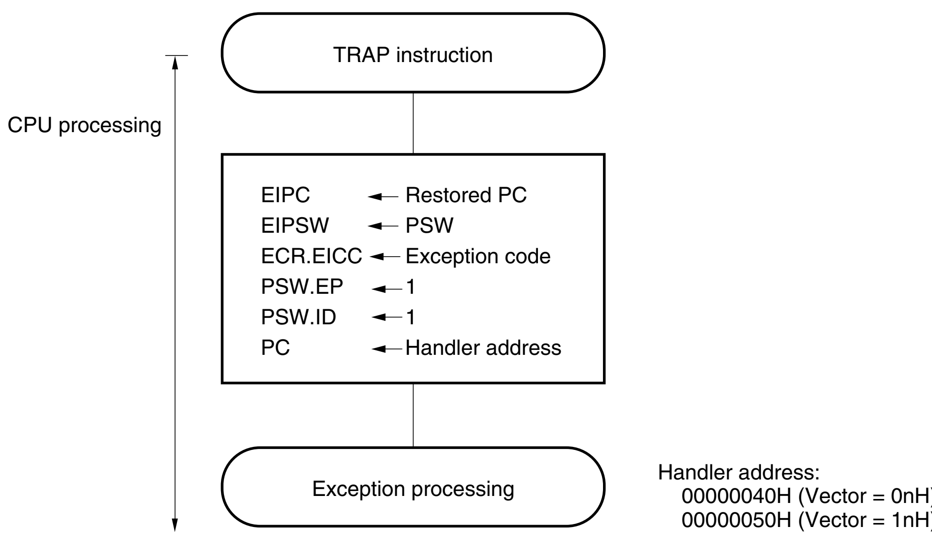
5.4.1 Operation

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 the 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 5-8. Software Exception Processing



5.4.2 Restore

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

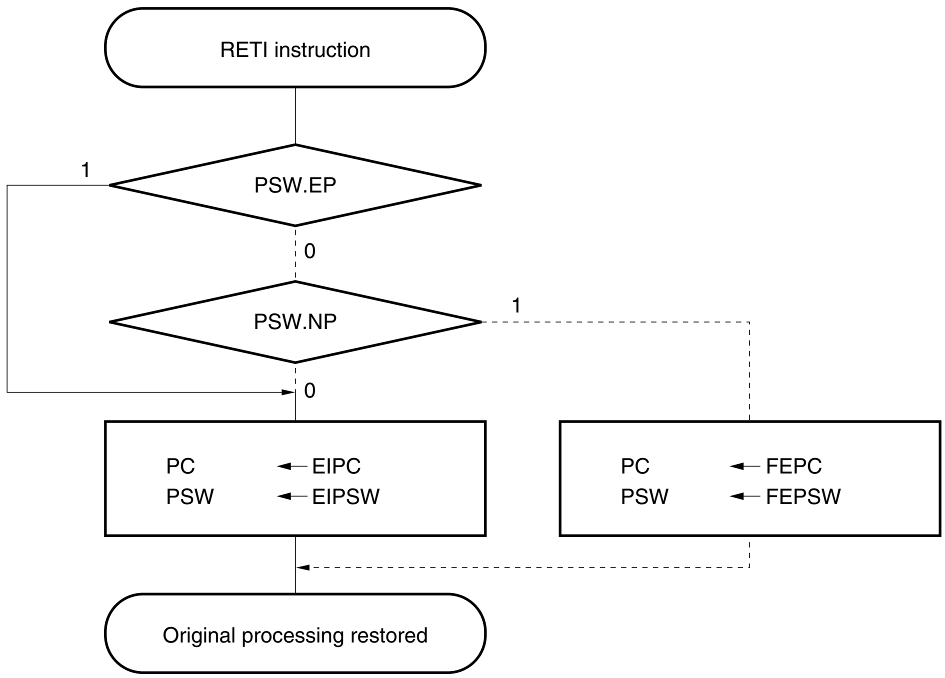
Operation of RETI instruction

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 5-9. RETI Instruction Processing



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

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/SB1 or V850/SB2, an illegal opcode exception (ILGOP: ILeGal 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 5-10. Illegal Opcode

15 13	12 11	10 5	4 0	31 27	26 23	22 21	20 17	16
x x x	x x	1 1 1 1 1 1	x x x x x	x x x x x	0 0 1 1 to 1 1 1 1	x x	x x x x	x

x: don't care

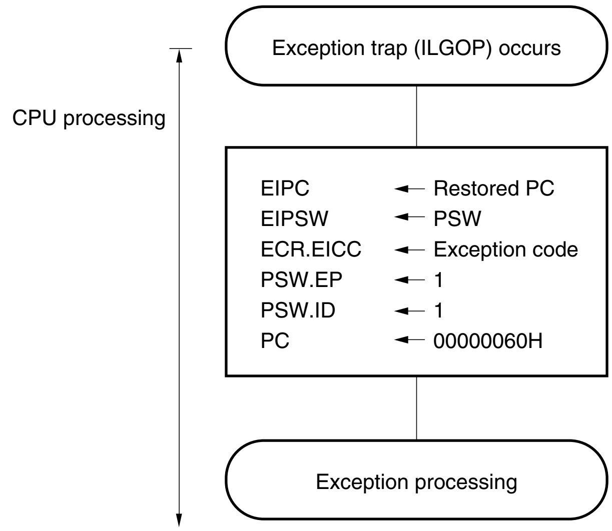
5.5.2 Operation

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 5-11. Exception Trap Processing



5.5.3 Restore

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

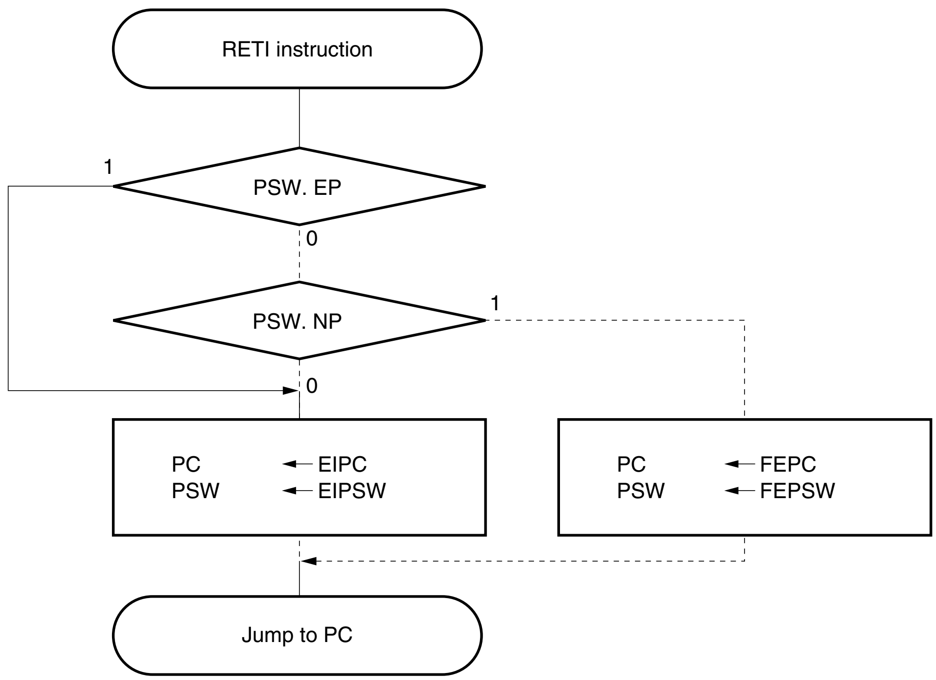
Operation of RETI instruction

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 5-12. RETI Instruction Processing



Caution **When the PSW.EP bit 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 interrupt servicing

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 service 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 in the interrupt enabled state (ID = 0). Even in an interrupt servicing routine, this control must be set in the interrupt enabled state (ID = 0). If a maskable interrupt is acknowledged or 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 acknowledgment) ... ... • DI instruction (disables interrupt acknowledgment) • Restore saved value to EIPSW • Restore saved value to EIPC • RETI instruction	← Acknowledgment 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 acknowledgment) ...	
• TRAP instruction	← Acknowledgment exception such as TRAP instruction.
• Illegal opcode	← Acknowledgment exception such as illegal opcode.
... • Restore saved value to EIPSW • Restore saved value to EIPC • RETI instruction	

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.

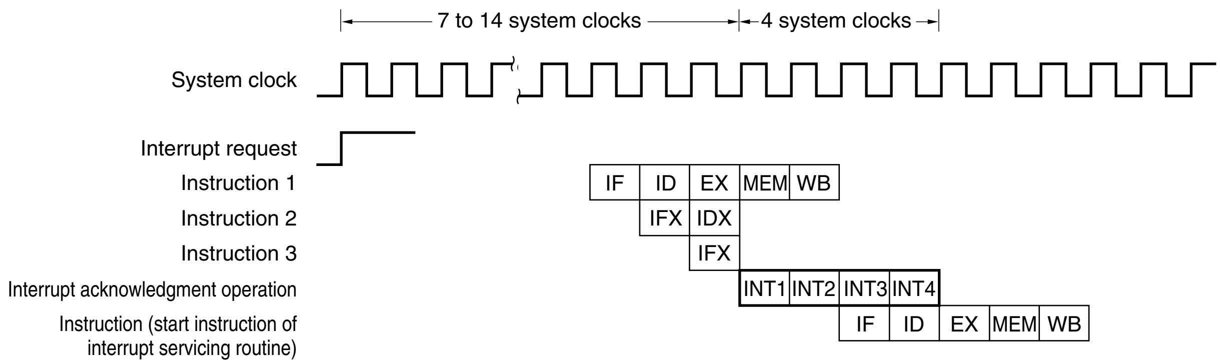
A pending interrupt request is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

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 interrupt latency time (from interrupt request generation to start of interrupt servicing).

Figure 5-13. Pipeline Operation at Interrupt Request Acknowledgment



INT1 to INT4: Interrupt acknowledgment 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/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 Interrupt Is 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 instruction

- 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 acknowledgment 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/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.

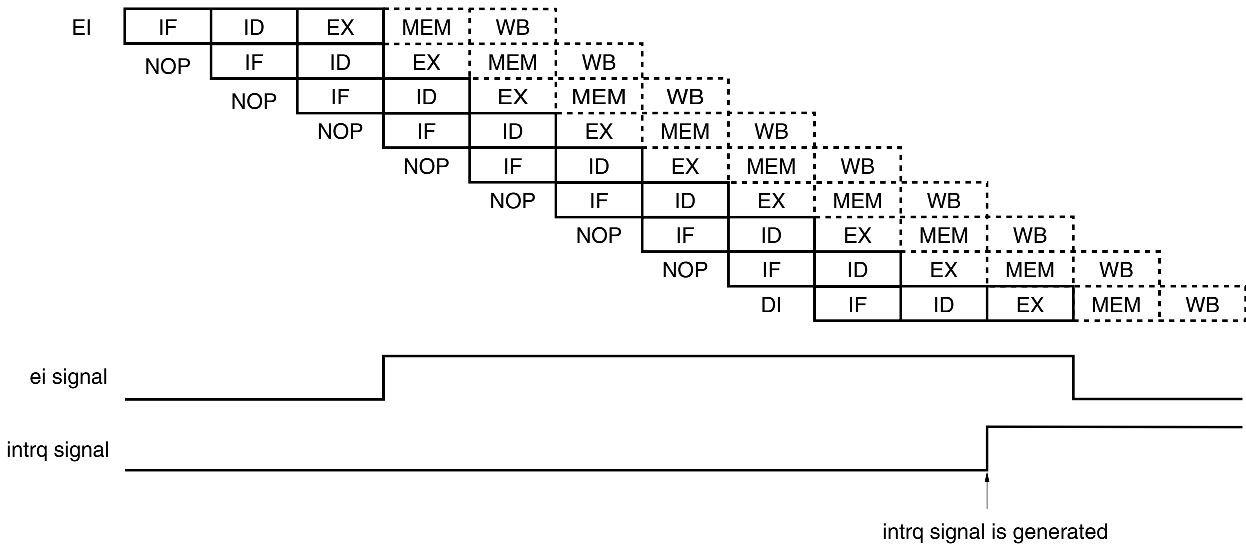
Remarks 1. In this example, the DI instruction is executed at the eighth clock after execution of the EI instruction, so the CPU acknowledges an interrupt request signal and performs interrupt servicing.

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 four system clocks after the CPU acknowledges the interrupt request signal.

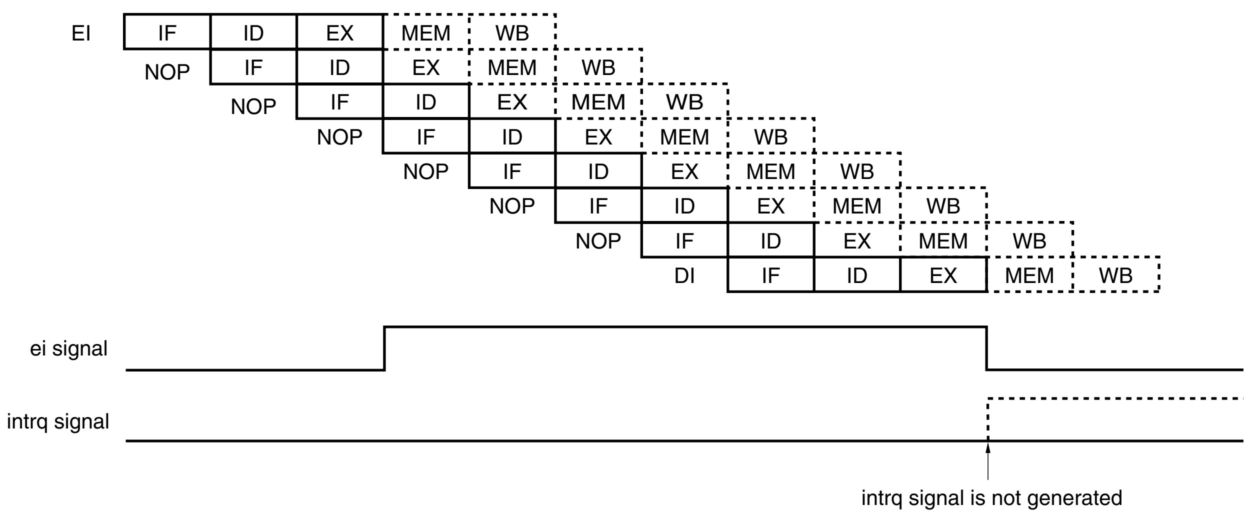
3. This example shows the case in which an interrupt request signal is generated (IF flag = 1) before the EI instruction is executed. If an interrupt request signal is generated after the EI instruction is executed, the CPU does not acknowledge the interrupt request signal if interrupts are disabled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

Figure 5-14. Pipeline Flow and Interrupt Request Signal Generation Timing

(a) When DI instruction is executed at eighth clock after EI instruction execution (interrupt request is acknowledged)



(b) When DI instruction is executed at seventh clock after EI instruction execution (interrupt request is not acknowledged)



★ 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**)

5.10 Key Interrupt Function

Key interrupts can be generated by inputting a falling edge to the key input pins (KR0 to KR7) by setting the key return mode register (KRM). The key return mode register (KRM) includes 5 bits. The KRM0 bit controls the KR0 to KR3 signals in 4-bit units and the KRM4 to KRM7 bits control corresponding signals from KR4 to KR7 (arbitrary setting from 4 to 8 bits is possible).

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

(1) Key return mode register (KRM)

After reset: 00H R/W Address: FFFFF3D0H

	<7>	<6>	<5>	<4>	3	2	1	<0>
KRM	KRM7	KRM6	KRM5	KRM4	0	0	0	KRM0

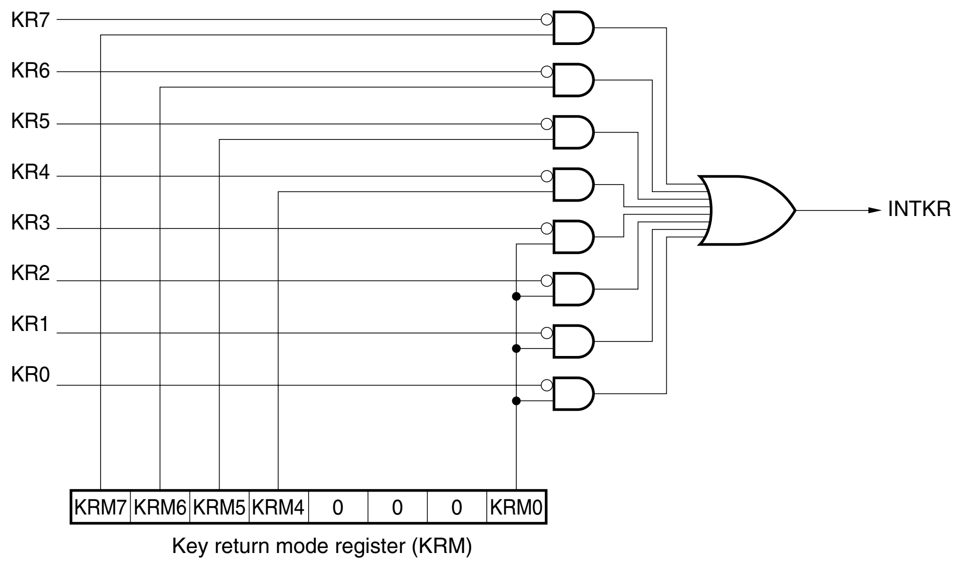
KRMn	Key return mode control
0	Do not detect key return signal
1	Detect key return signal

Caution If the key return mode register (KRM) is changed, an interrupt request flag may be set. To avoid setting this flag, change the KRM register after disabling interrupts, and then enable interrupts after clearing the interrupt request flag.

Table 5-4. Description of Key Return Detection Pin

Flag	Pin Description
KRM0	Controls KR0 to KR3 signals in 4-bit units
KRM4	Controls KR4 signal in 1-bit units
KRM5	Controls KR5 signal in 1-bit units
KRM6	Controls KR6 signal in 1-bit units
KRM7	Controls KR7 signal in 1-bit units

Figure 5-15. Key Return Block Diagram



CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Outline

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

The oscillator of V850/SB1 has an oscillation frequency of 2 to 20 MHz. The oscillator of the A and B versions of the V850/SB2 has an oscillation frequency of 2 to 12.58 MHz and the oscillator of the H version of the V850/SB2 has an oscillation frequency of 2 to 19 MHz. Oscillation can be stopped by executing a STOP instruction or by setting the processor clock control register (PCC). Oscillation is also stopped during a reset.

In IDLE mode, supplying the peripheral clock to the clock timer only is possible. Therefore, in IDLE mode, it is possible to operate the clock timer without using the subclock oscillator.

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

2. If the main clock halt 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.

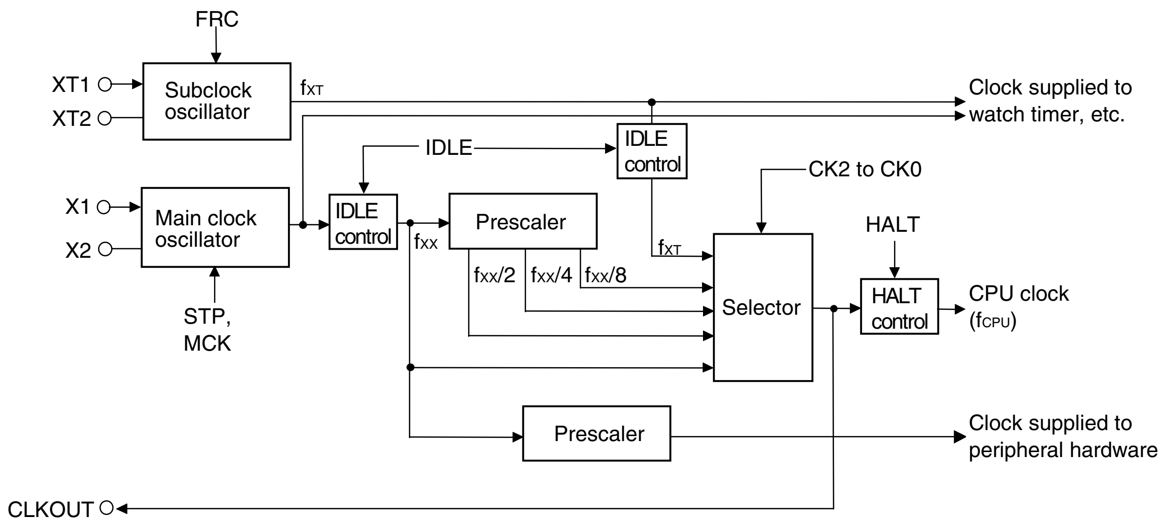
(2) Subclock oscillator

This circuit has an oscillation frequency of 32.768 kHz. Its oscillation is not stopped when the STOP instruction is executed, neither is it stopped when a reset is input.

When the subclock oscillator is not used, the FRC bit in the processor clock control register (PCC) can be set to disable use of the internal feedback resistor. This enables the current consumption to be kept low in the STOP mode.

6.2 Configuration

Figure 6-1. Clock Generator



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 STOP mode (fixed to low level).

This function is controlled via the DCLK1 and DCLK0 bits in the power save control register (PSC).

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 processor clock control register (PCC)).

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	FRC	MCK	0	0	0	CK2	CK1	CK0

FRC	Selection of internal feedback resistor for subclock
0	Use
1	Do not use

MCK	Operation of main clock (main clock)
0	Operating
1	Stopped

CK2[Notes 1, 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 HALT or 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. Be sure to set bits 3 to 5 to 0.**

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 most 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.
For details, 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's 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	STOP mode setting
0	Normal mode
1	STOP mode[Note 2]

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

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

(3) Oscillation stabilization time selection 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		
			Clock	f_{XX}	
				20 MHz[Note]	12.58 MHz
0	0	0	$2^{14}/f_{XX}$	819.2 μs	1.3 ms
0	0	1	$2^{16}/f_{XX}$	3.3 ms	5.2 ms
0	1	0	$2^{17}/f_{XX}$	6.6 ms	10.4 ms
0	1	1	$2^{18}/f_{XX}$	13.1 ms	20.8 ms
1	0	0	$2^{19}/f_{XX}$	26.2 ms	41.6 ms
Other than above			Setting prohibited		

Note Only in the V850/SB1.

6.4 Power Save Functions

6.4.1 Outline

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 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 saving 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 (leak current only) is set. 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 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 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 on-chip 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 on the main clock.

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 canceled 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)**

Item \ HALT Mode Setting		When CPU Operates with Main Clock	
		When Subclock Does Not Exist	When Subclock Exists
CPU		Stopped	
ROM correction		Stopped	
Clock generator		Oscillation for main clock and subclock Clock supply to CPU is stopped	
16-bit timer (TM0)		Operating	
16-bit timer (TM1)		Operating	
8-bit timer (TM2)		Operating	
8-bit timer (TM3)		Operating	
8-bit timer (TM4)		Operating	
8-bit timer (TM5)		Operating	
8-bit timer (TM6)		Operating	
8-bit timer (TM7)		Operating	
Watch timer		Operates when main clock is selected for count clock	Operating
Watchdog timer		Operating (interval timer only)	
Serial interface	CSI0 to CSI3	Operating	
	I^2C0 Note, I^2C1 Note	Operating	
	UART0, UART1	Operating	
	CSI4	Operating	
IEBus (V850/SB2 only)		Operating	
A/D converter		Operating	
DMA0 to DMA5		Operating	
Real-time output		Operating	

Note Available only in the Y versions (products with on-chip I^2C).

★

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

Item (HALT Mode Setting)		When CPU Operates with Main Clock	
		When Subclock Does Not Exist	When Subclock Exists
Port function		Held	
External bus interface		Only bus hold function operates	
External interrupt request	NMI	Operating	
	INTP0 to INTP3	Operating	
	INTP4 and INTP5	Operating	
	INTP6	Operating	
Key return function		Operating	
In external expansion mode	AD0 to AD15	High impedance[Note]	
	A16 to A21	Held[Note] (high impedance when $\overline{\text{HLDAK}}$ = 0)	
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	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 according to 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 saving 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, or $\overline{\text{RESET}}$ input.

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

<table>
<tr><th colspan="2">IDLE Mode Settings / Item</th><th>When Subclock Exists</th><th>When Subclock Does Not Exist</th></tr>
<tr><td colspan="2">CPU</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">ROM correction</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">Clock generator</td><td colspan="2">Both main clock and subclock oscillator
Clock supply to CPU and on-chip peripheral functions is stopped</td></tr>
<tr><td colspan="2">16-bit timer (TM0)</td><td>Operates when INTWTNI is selected as count clock (f_{XT} is selected for watch timer)</td><td>Stopped</td></tr>
<tr><td colspan="2">16-bit timer (TM1)</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">8-bit timer (TM2)</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">8-bit timer (TM3)</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">8-bit timer (TM4)</td><td>Operates when f_{XT} is selected for count clock</td><td>Stopped</td></tr>
<tr><td colspan="2">8-bit timer (TM5)</td><td>Operates when f_{XT} is selected for count clock</td><td>Stopped</td></tr>
<tr><td colspan="2">★ 8-bit timer (TM6)</td><td colspan="2">Operates when TO0 is selected as count clock (however, only when TM0 is operating)</td></tr>
<tr><td colspan="2">★ 8-bit timer (TM7)</td><td colspan="2">Operates when TO0 is selected as count clock (however, only when TM0 is operating)</td></tr>
<tr><td colspan="2">Watch timer</td><td colspan="2">Operating</td></tr>
<tr><td colspan="2">Watchdog timer</td><td colspan="2">Stopped</td></tr>
<tr><td rowspan="4">Serial interface</td><td>CSI0 to CSI3</td><td colspan="2">Operates when an external clock is selected as the serial clock</td></tr>
<tr><td>I^2C0 Note, I^2C1 Note</td><td colspan="2">Stopped</td></tr>
<tr><td>★ UART0, UART1</td><td colspan="2">Operates when an external clock is selected as the baud-rate clock (transmit only)</td></tr>
<tr><td>CSI4</td><td colspan="2">Operates when an external clock is selected as the serial clock</td></tr>
<tr><td colspan="2">IEBus (V850/SB2 only)</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">A/D converter</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">DMA0 to DMA5</td><td colspan="2">Stopped</td></tr>
<tr><td colspan="2">Real-time output</td><td>Operates when INTTM4 or INTTM5 is selected (when TM4 or TM5 is operating)</td><td>Stopped</td></tr>
<tr><td colspan="2">Port function</td><td colspan="2">Held</td></tr>
</table>

Note Available only in the Y versions (products with on-chip I^2C).

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

Item \ IDLE Mode Settings		When Subclock Exists	When Subclock Does Not Exist
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3	Operating	
	INTP4 and INTP5	Stopped	
	INTP6	Operates when f_{XT} is selected for sampling clock	Stopped
Key return		Operating	
In external expansion mode	AD0 to AD15	High impedance	
	A16 to A21		
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$		
	R/$\overline{\text{W}}$		
	$\overline{\text{DSTB}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$		
	ASTB		
	$\overline{\text{HLDAK}}$		

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 FRC bit in the processor clock control register (PCC) is set to 1, the subclock oscillator's on-chip feedback resistor is cut. This sets ultra low power consumption mode in which the only current is the device's leak current.

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 operating (peripheral functions operating on the subclock are not stopped).

★ External bus hold requests ($\overline{\text{HLDRQ}}$) 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 for software STOP mode are listed in Table 6-3.

(2) Release of software STOP mode

Software STOP mode can be released by an non-maskable interrupt, an unmasked interrupt request, or $\overline{\text{RESET}}$ input.

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

Table 6-3. Operating Statuses in Software STOP Mode (1/2)

	Item \ STOP Mode Settings		When Subclock Exists	When Subclock Does Not Exist
	CPU		Stopped	
	ROM correction		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 INTWTNI 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 (operation disabled)
★	8-bit timer (TM6)		Operates when TO0 is selected as count clock (however, only when TM0 is operating)	
★	8-bit timer (TM7)		Operates when TO0 is selected as count clock (however, only when TM0 is operating)	
	Watch timer		Operates when f_{XT} is selected for count clock	Stopped
	Watchdog timer		Stopped	
	Serial interface	CSI0 to CSI3	Operates when an external clock is selected as the serial clock	
		I^2C0^{Note}, I^2C1^{Note}	Stopped	
★		UART0, UART1	Operates when an external clock is selected as the baud-rate clock (transmit only)	
		CSI4	Operates when an external clock is selected as the serial clock	
	IEBus (V850/SB2 only)		Stopped	
	A/D converter		Stopped	

Note Available only in the Y versions (products with on-chip I^2C).

Table 6-3. Operating Statuses in Software STOP Mode (2/2)

Item \ Mode Settings		When Subclock Exists	When Subclock Does Not Exist
DMA0 to DMA5		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 and INTP5	Stopped	
	INTP6	Operates when f_{XT} is selected for the noise eliminator	Stopped
Key return		Operating	
In external expansion mode	AD0 to AD15	High impedance	
	A16 to A21		
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$		
	R/$\overline{\text{W}}$		
	$\overline{\text{DSTB}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$		
	ASTB		
	$\overline{\text{HLDAK}}$		

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 STOP mode.

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

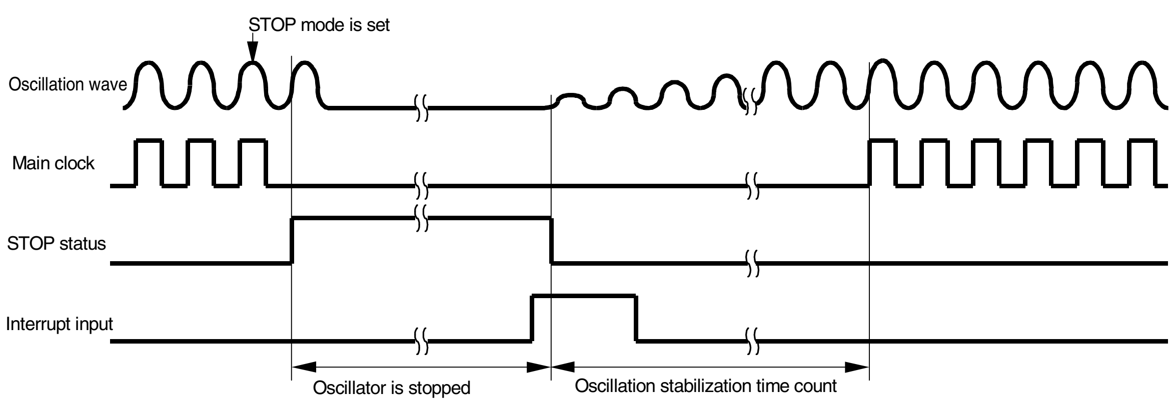
STOP mode is released by a non-maskable interrupt or an unmasked interrupt request. When an interrupt is input, the counter (watchdog timer) starts counting and the count time is the length of time that must elapse for stabilization of the oscillator's clock output.

★ 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 $2^{19}/f_{XX}$ according to the value of the OSTS register after reset.

6.6 Notes on Power Save Function

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

To set the power save mode (IDLE mode or 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 register 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 register 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 (<6>) if it is 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/STOP mode
:
```

Remark The above example assumes that rD (PSC set value), rX (value to be written to PSW), and rY (value rewritten to PSW) are already set.

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 (<4>) to the PSC register for setting the IDLE mode and STOP mode are executed before entering the power save mode.

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

If the V850/SB1 or V850/SB2 is used under the following conditions, the address indicated by the program counter (PC) differs from the address that actually reads an instruction after the power save mode has been released.

Of the instructions 4 to 16 bytes after the instruction that writes data to the PSC register, the CPU may ignore 4 or 8 bytes of the instruction and execute wrong instructions.

★ **Caution A PC discrepancy occurs only when all the conditions (i) to (iii) in [Conditions] below are met. It does not occur if even one condition is not met.**

[Conditions]

(i) If the power save mode (IDLE or STOP mode) is set while an instruction is being executed on the external ROM

(ii) If the power save mode is released by an interrupt request

(iii) If the subsequent instruction is executed while an interrupt request is being held pending after the power save mode has been released

Conditions in which an interrupt request is held pending:

- If the NP flag of the PSW register is "1" (during NMI servicing/set by software)
- If the ID flag of the PSW register is "1" (during interrupt request servicing/DI instruction/set by software)
- If the power save mode is released by an interrupt request with a priority the same as or lower than the interrupt request being serviced even though interrupts are enabled (EI status)

Therefore, use the V850/SB1 and V850/SB2 under the following conditions:

[Conditions]

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

(ii) Take the following measures using software if a power save mode is used while an instruction is being executed on the external ROM:

- Insert six NOP instructions 4 bytes after the instruction that writes data to the PSC register.
- Insert the BR $+2 instruction to eliminate the discrepancy in the address of the PC after the NOP instructions.

[Example of prevention program]

```
LDSR    rx, 5               ; Sets value of rX to PSW.
ST.B    r0, PRCMD[r0]       ; Writes data to PRCMD.
ST.B    rD, PSC[r0]         ; Sets PSC register.
LDSR    rY, 5               ; Returns value of PSW.
NOP                         ; Six NOP instructions or more
NOP
NOP
NOP
NOP
BR $+2                      ; Eliminates discrepancy of PC
```

Remark It is assumed that the following values have already been set:
rD: PSC set value, rX: Value written to PSW, rY: Value written back to PSW

CHAPTER 7 TIMER/COUNTER FUNCTION

7.1 16-Bit Timer (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 TO0 and TO1 (timer outputs), set the value of port 3 (P3) to 0 (port mode output) and the port 3 mode register (PM3) to 0. The logical sum (ORed) value of the output of the port and the timer is output.

Remark n = 0, 1

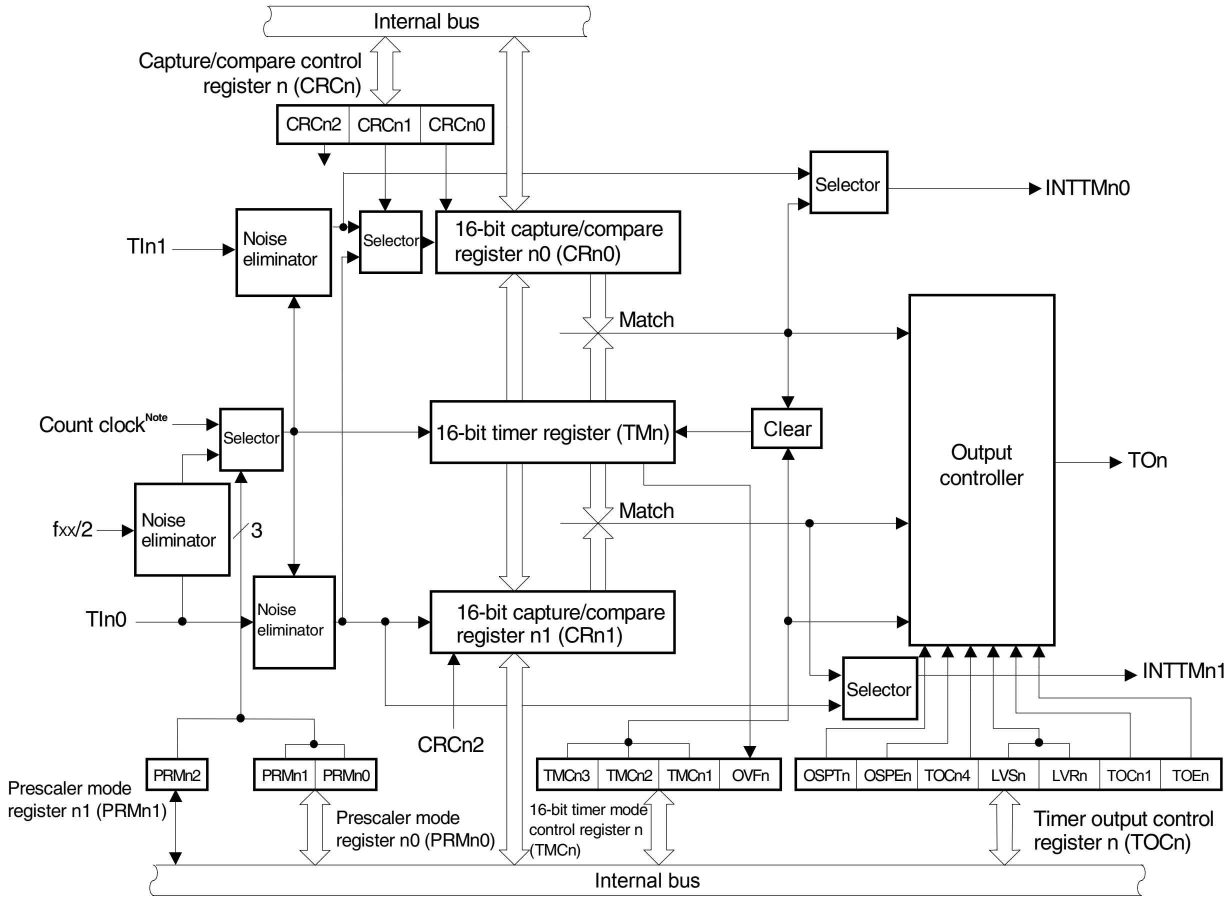
7.1.2 Function

TM0 and TM1 have the following functions.

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

Figure 7-1 shows the block diagram.

Figure 7-1. Block Diagram of TM0 and TM1



Note The count clock is set by the PRMn0 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 include the following hardware.

Table 7-1. Configuration of Timers 0 and 1

Item	Configuration
Timer registers	16 bits × 2 (TM0, TM1)
Registers	16-bit capture/compare registers: 16 bits × 2 each (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 registers n0, n1 (PRMn0, PRMn1)

(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 clear & start mode entered by inputting the valid edge of TIn0
<4> If TMn and CRn0 match in the clear & start mode entered upon 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) 16-bit capture/compare registers n0 (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 the CRCn0 bit 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 PRMn0 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 only be read by a 16-bit memory manipulation instruction when used as capture registers.

$\overline{\text{RESET}}$ input clears CRn0 to 0000H.

Caution Set CRn0 to a value other than 0000H in the clear & start mode entered upon a match between TMn and CRn0. In the free-running mode or the clear & start mode entered upon the TIn0 valid edge, however, an interrupt request (INTTMn0) is generated after an overflow (FFFFH) when CRn0 is set to 0000H.

(3) 16-bit capture/compare register n1 (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 the CRCn2 bit 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 TIn1 pin can be selected as a capture trigger. The valid edge of TIn1 is specified by the PRMn0 register.

When the capture trigger is specified as the valid edge of TIn0, the relationship between the TIn0 valid edge and the CRn1 capture trigger is as follows.

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

ESn01	ESn00	TIn0 Pin Valid Edge	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 only be read by a 16-bit memory manipulation instruction when used as capture registers.

$\overline{\text{RESET}}$ input clears CRn1 to 0000H.

Caution Set CRn1 to a value other than 0000H in the clear & start mode entered upon a match between TMn and CRn0. In the free-running mode or the clear & start mode entered upon the TIn0 valid edge, 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 registers to control timers 0 and 1 are shown below.

- 16-bit timer mode control register n (TMCn)
- Capture/compare control register n (CRCn)
- 16-bit timer output control register n (TOCn)
- Prescaler mode registers n0, n1 (PRMn0, PRMn1)

The following registers are also used.

- 16-bit timer register n (TMn)
- 16-bit capture/compare registers n0, n1 (CRn0, CRn1)

(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 (TMn).

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

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

Caution 16-bit timer register n starts operating when the TMCn2 and TMCn3 bits are set to values other than 0, 0 (operation stop mode). To stop the operation, set the TMCn2 and TMCn3 bits 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 selector	TOn output timing selector	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 bit is written, be sure to stop the timer operation.

2. The valid edge of the TIn0 pin is set by using prescaler mode register n0 (PRMn0).

3. When a mode in which the timer is cleared and started on a match between TMn and CRn0 is selected, the OVFn bit 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 16-bit capture/compare register n (CRn0 and CRn1).

CRCn is set by an 8-bit or 1-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 for the CRn0 register.

4. For the capture trigger, a pulse longer than twice the count clock selected by prescaler mode registers 0n, 1n (PRM0n, PRM1n) is required in order that the signals from TIn0 and T2n1 perform the capture operation correctly.

★ **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 an 8-bit or 1-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	0	OSPTn	OSPEn	TOCn4	LVSn	LVRn	TOCn1	TOEn

(n = 0, 1)

OSPTn	Control of output trigger of one-shot pulse by software
0	No one-shot pulse trigger
1	Uses one-shot pulse trigger

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 disabled

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 or TIn0 valid edge
0	Reverse timer output F/F disabled
1	Reverse timer output F/F enabled

TOEn	Controls output of timer n
0	Output disabled (output is fixed to 0 level)
1	Output enabled

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

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 (to 1) other than for one-shot pulse output (OSPEn bit = 0).

(4) Prescaler mode registers 00, 01 (PRM00, PRM01)

PRM0n selects the count clock of the 16-bit timer (TM0) and the valid edge of TI0n input. PRM00 and PRM01 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM00 and PRM01 to 00H.

After reset: 00H R/W Address: FFFFF206H

	7	6	5	4	3	2	1	0
PRM00	ES011	ES010	ES001	ES000	0	0	PRM01	PRM00

After reset: 00H R/W Address: FFFFF20EH

	7	6	5	4	3	2	1	0
PRM01	0	0	0	0	0	0	0	PRM02

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	fxx	
				20 MHz[Note 2]	12.58 MHz
0	0	0	fxx/2	100 ns	158 ns
0	0	1	fxx/16	800 ns	1.3 μs
0	1	0	INTWTNI	–	–
0	1	1	TI00 valid edge[Note 1]	–	–
1	0	0	fxx/4	200 ns	318 ns
1	0	1	fxx/64	3.2 μs	5.1 μs
1	1	0	fxx/256	12.8 μs	20.3 μs
1	1	1	Setting prohibited	–	–

Notes 1. The external clock requires a pulse longer than twice that of the internal clock (fxx/2).

2. Only in the V850/SB1.

Cautions 1. When selecting the valid edge of TI00 as the count clock, do not specify the clear & start mode entered on the valid edge of TI00 or TI00 as a capture trigger.

2. Before setting data to PRM0n, 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 10, 11 (PRM10, PRM11)

PRM1n selects the count clock of the 16-bit timer (TM1) and the valid edge of TI1n input. PRM10 and PRM11 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM10 and PRM11 to 00H.

After reset: 00H R/W Address: FFFFF216H

	7	6	5	4	3	2	1	0
PRM10	ES111	ES110	ES101	ES100	0	0	PRM11	PRM10

After reset: 00H R/W Address: FFFFF21EH

	7	6	5	4	3	2	1	0
PRM11	0	0	0	0	0	0	0	PRM12

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	PRM11	PRM10	Count clock selection		
			Count clock	fxx	
				20 MHz[Note 2]	12.58 MHz
0	0	0	fxx/2	100 ns	158 ns
0	0	1	fxx/4	200 ns	318 ns
0	1	0	fxx/16	800 ns	1.3 μs
0	1	1	TI10 valid edge[Note 1]	–	–
1	0	0	fxx/32	1.6 μs	2.5 μs
1	0	1	fxx/128	6.4 μs	10.2 μs
1	1	0	fxx/256	12.8 μs	20.3 μs
1	1	1	Setting prohibited	–	–

Notes 1. The external clock requires a pulse longer than twice that of the internal clock (fxx/2).

2. Only in the V850/SB1.

Cautions 1. When selecting the valid edge of TI10 as the count clock, do not specify the clear & start mode entered on the valid edge of TI10 or TI10 as a capture trigger.

2. Before setting data to PRM1n, always stop the timer operation.

3. If the 16-bit timer (TM1) operation is enabled by specifying the rising edge or both edges as 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 the 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 (n = 0, 1).

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 with 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 the PRMn0 and PRMn1 bits of prescaler mode register n0 (PRMn0) and by the PRMn2 bit of prescaler mode register n1 (PRMn1).

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	0	0	0	0	1	1	0/1	0

TMCn3, TMCn2: Clears and starts on match between TMn and CRn0.

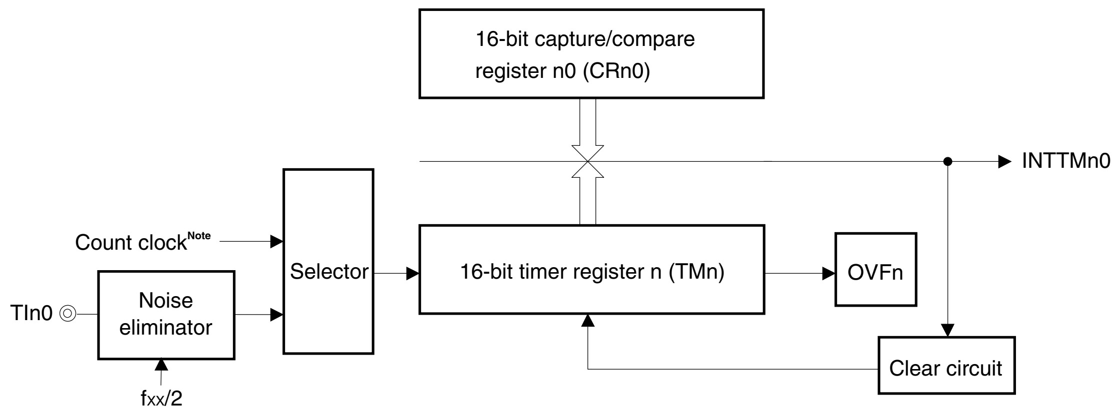
(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	0/1	0/1	0

CRCn0: 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

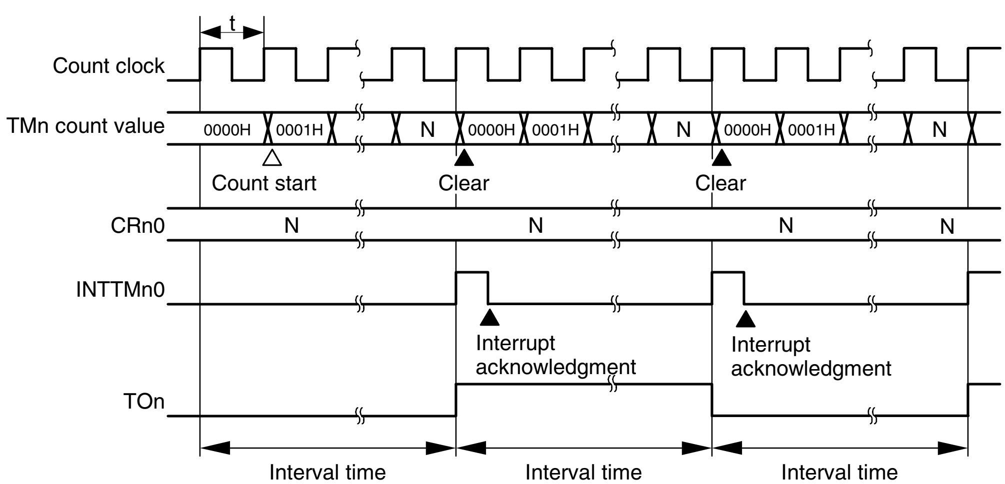


Note The count clock is set by the PRMn0 and PRMn1 registers.

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

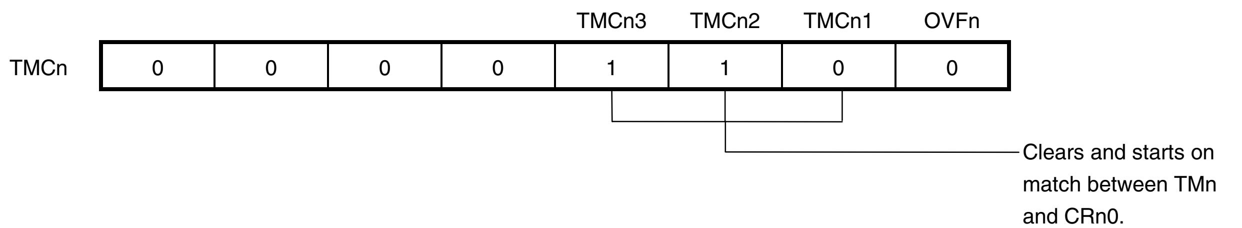
7.2.2 PPG output operation

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.

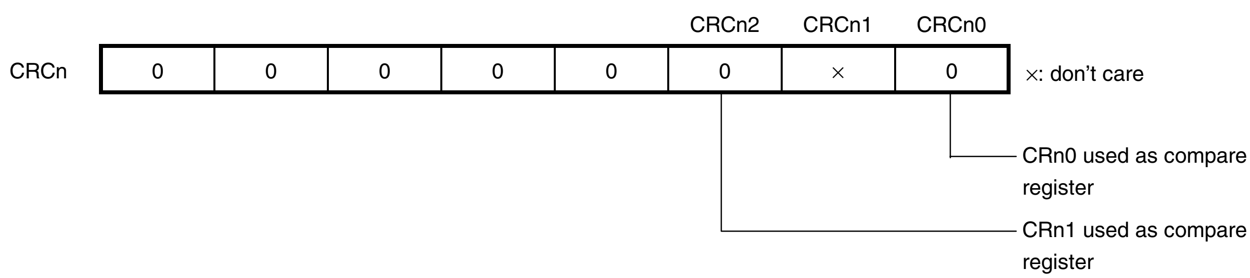
The PPG output function outputs a square wave from the TOn pin with a cycle specified by the count value set in advance to 16-bit capture/compare register n0 (CRn0) and a pulse width specified by the count value set in advance to 16-bit capture/compare register n1 (CRn1).

Figure 7-5. Control Register Settings in PPG Output Operation

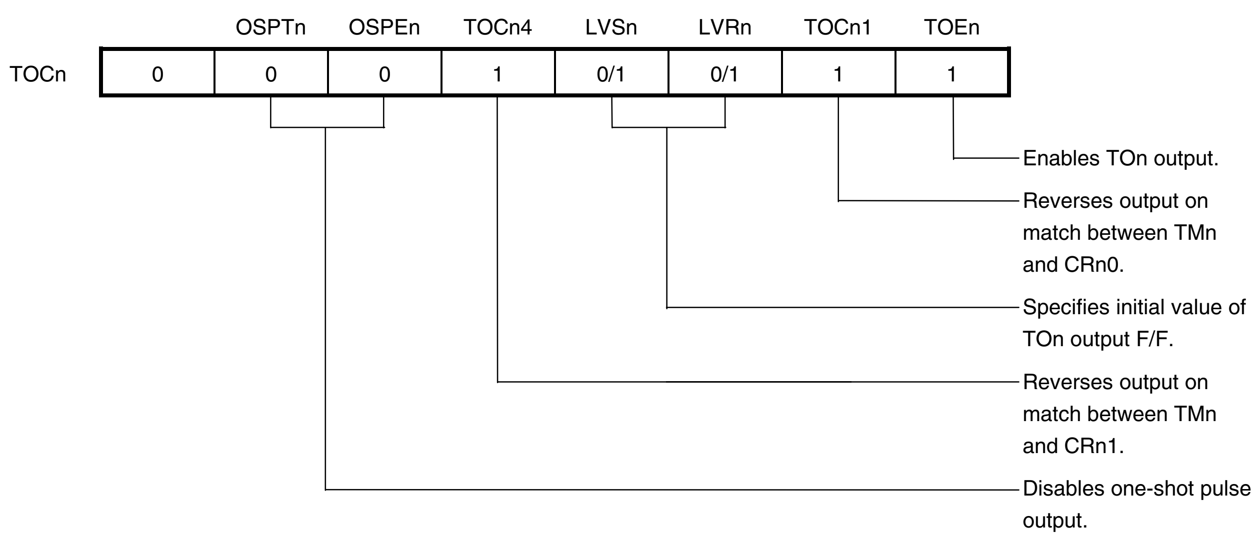
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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



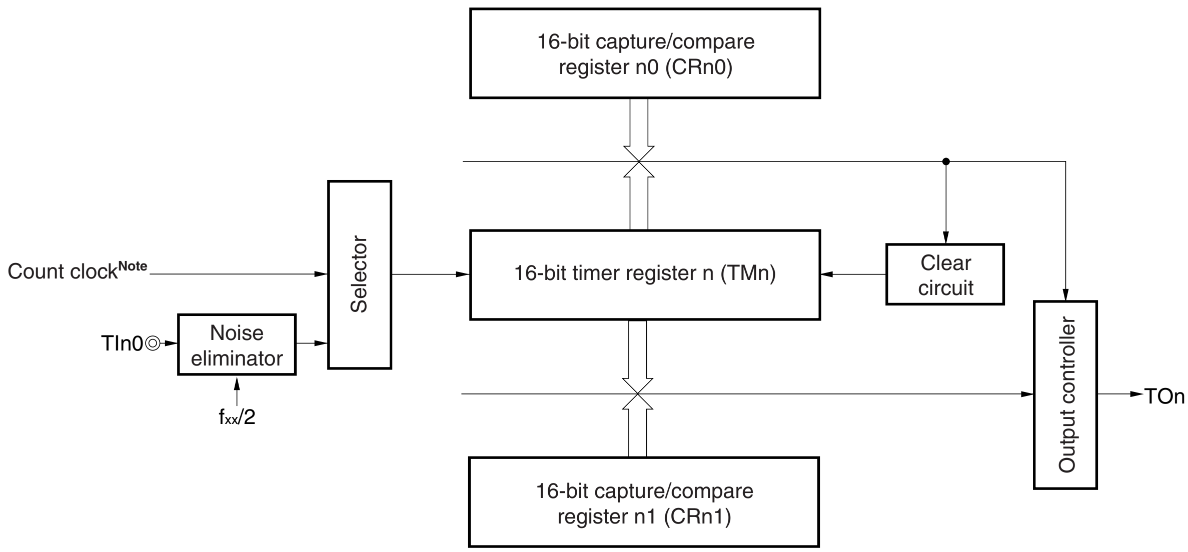
(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)



Cautions 1. Make sure that CRn0 and CRn1 are set to 0000H < CRn1 < CRn0 ≤ FFFFH.

2. PPG output sets the pulse cycle to (CRn0 set value + 1).
The duty factor is (CRn1 set value + 1)/(CRn0 set value + 1).

★ **Figure 7-6. Configuration of PPG Output**

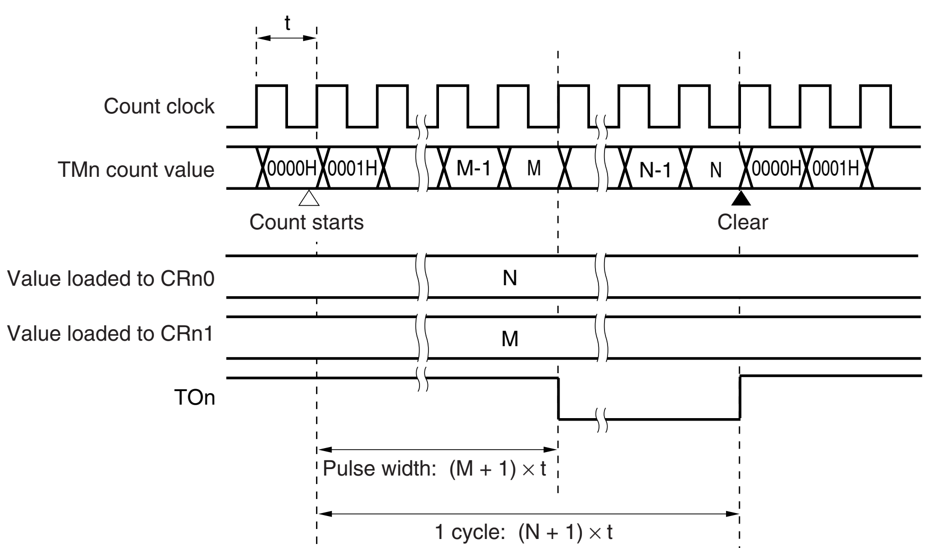


Note The count clock is set by the PRMn0 and PRMn1 registers.

Remarks 1. "◎—— " indicates a signal that can be directly connected to a port.

2. n = 0, 1

★ **Figure 7-7. PPG Output Operation Timing**



Remarks 1. $0000H \leq M < N \leq FFFFH$

2. n = 0, 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

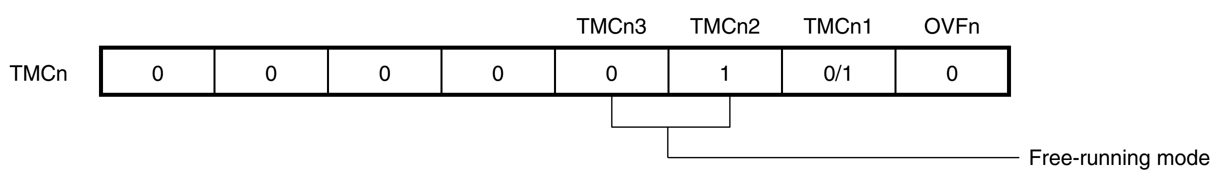
If the edge specified by prescaler mode register n0 (PRMn0) is input to the TIn0 pin when 16-bit timer register n (TMn) is used as a free-running counter (refer to **Figure 7-8**), the value of TMn is loaded to 16-bit capture/compare register n1 (CRn1), and an external interrupt request signal (INTTMn1) is set.

The edge is specified by using bits 6 and 7 (ESn10 and ESn11) of prescaler mode register n0 (PRMn0). The rising edge, falling edge, or both the rising and falling edges can be selected.

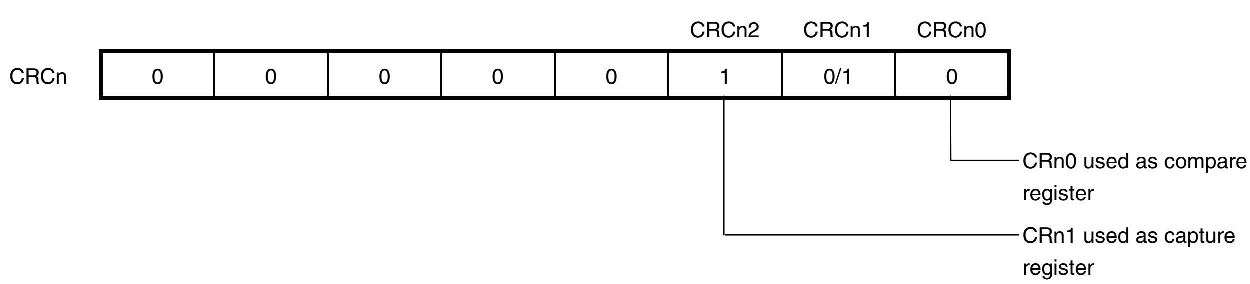
The valid edge is detected by sampling at the count clock cycle selected by prescaler mode register n0, n1 (PRMn0, PRMn1), and a capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Figure 7-8. 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)

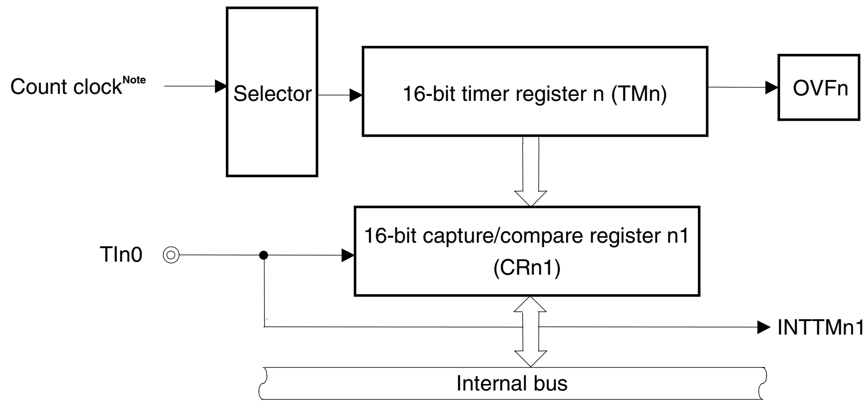


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



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 7-9. Configuration for Pulse Width Measurement with Free-Running Counter

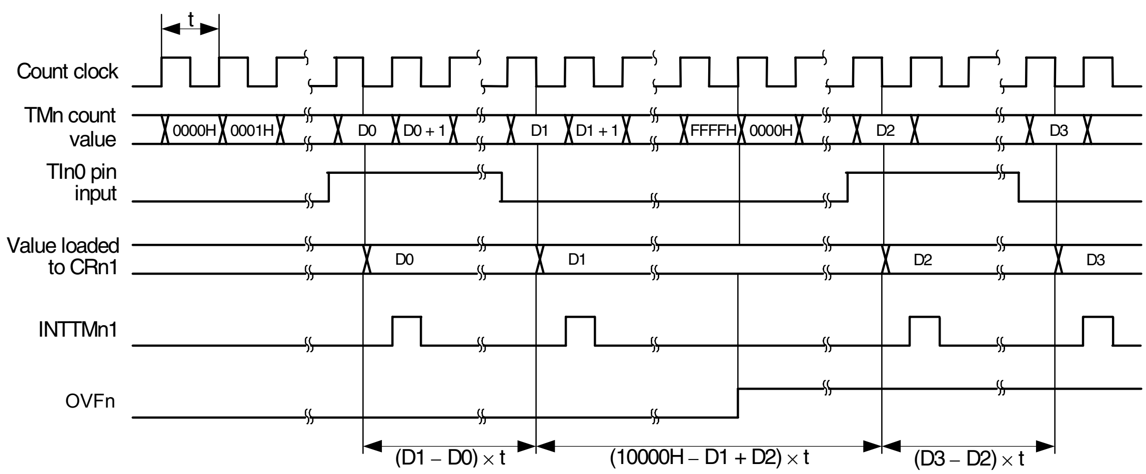


Note The count clock is set by the PRMn0 and PRMn1 registers.

Remarks 1. "◎——" indicates a signal that can be directly connected to a port.

2. n = 0, 1

Figure 7-10. 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 7-11**).

When the edge specified by the ESn00 and ESn01 bits of prescaler mode register n0 (PRMn0) is input to the TIn0 pin, the value of the TMn is loaded to 16-bit capture/compare register n1 (CRn1) and an external interrupt request signal (INTTMn1) is set.

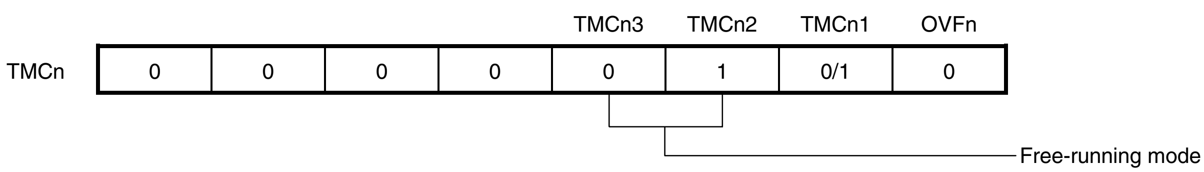
When the edge specified by the ESn10 and ESn11 bits in PRMn0 is input to the TIn1 pin, the value of TMn is loaded to 16-bit capture/compare register n0 (CRn0), and an external interrupt request signal (INTTMn0) is set.

The edges of the TIn0 and TIn1 pins are specified by the ESn00 and ESn01 bits and the ESn10 and ESn11 bits of PRMn0, respectively. The rising, falling, or both rising and falling edges can be specified.

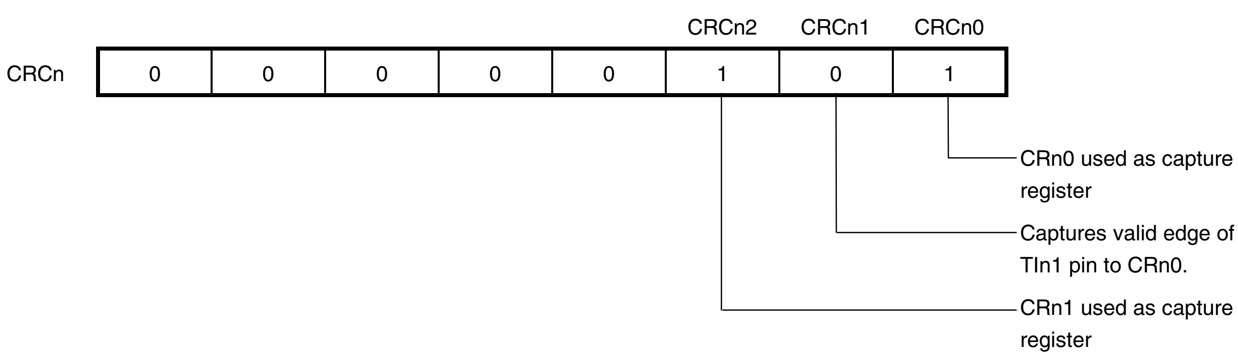
The valid edge is detected by sampling at the count clock cycle selected by prescaler mode register n0, n1 (PRMn0, PRMn1), and a capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be eliminated.

Figure 7-11. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

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



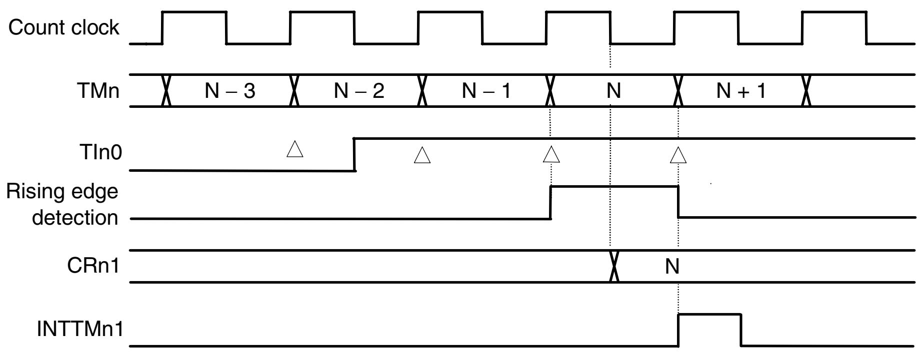
(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



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)**.

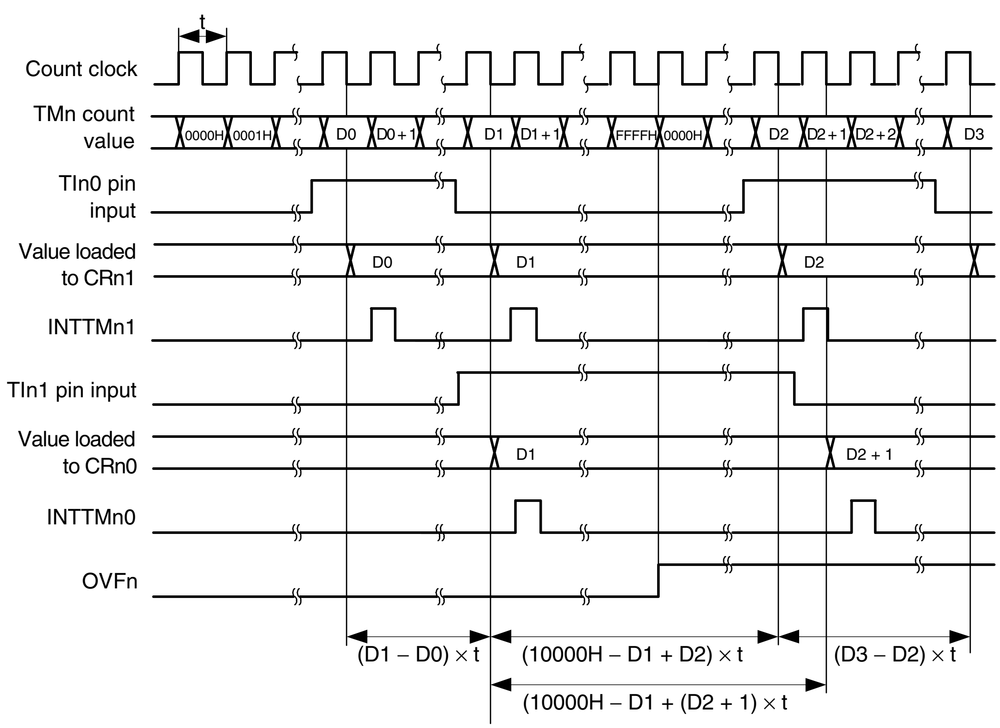
- **Capture operation (free-running mode)**
  The following figure illustrates the operation of the capture register when the capture trigger is input.

Figure 7-12. CRn1 Capture Operation with Rising Edge Specified



Remark n = 0, 1

Figure 7-13. 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 7-14**), the pulse width of the signal input to the TIn0 pin can be measured.

When the edge specified by the ESn00 and ESn01 bits of prescaler mode register n0 (PRMn0) 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.

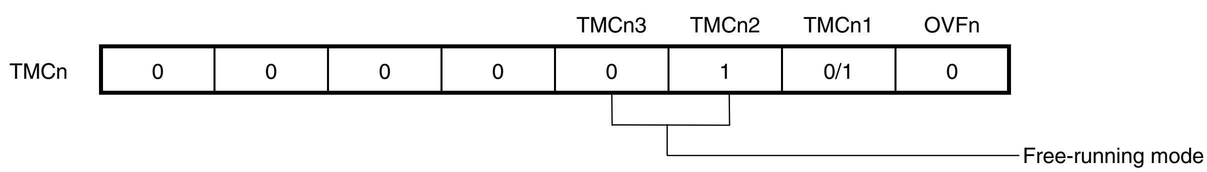
The edge of the TIn0 pin is specified by the ESn00 and ESn01 bits of prescaler mode register n (PRMn0). The rising or falling edge can be specified.

The valid edge of TIn0 is detected by sampling at the count clock cycle selected by prescaler mode register n0, n1 (PRMn0, 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.

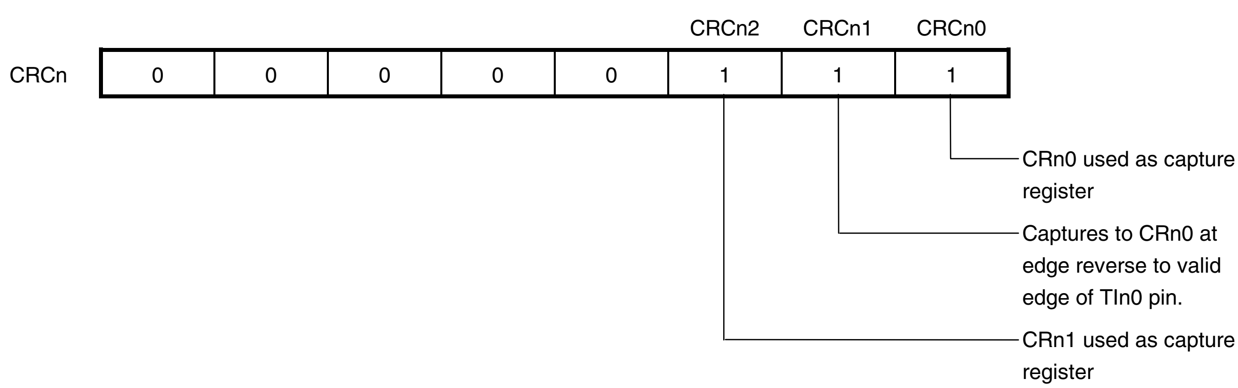
Caution If the valid edge of the TIn0 pin is specified to be both the rising and falling edges, capture/compare register n0 (CRn0) cannot perform its capture operation.

Figure 7-14. 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)

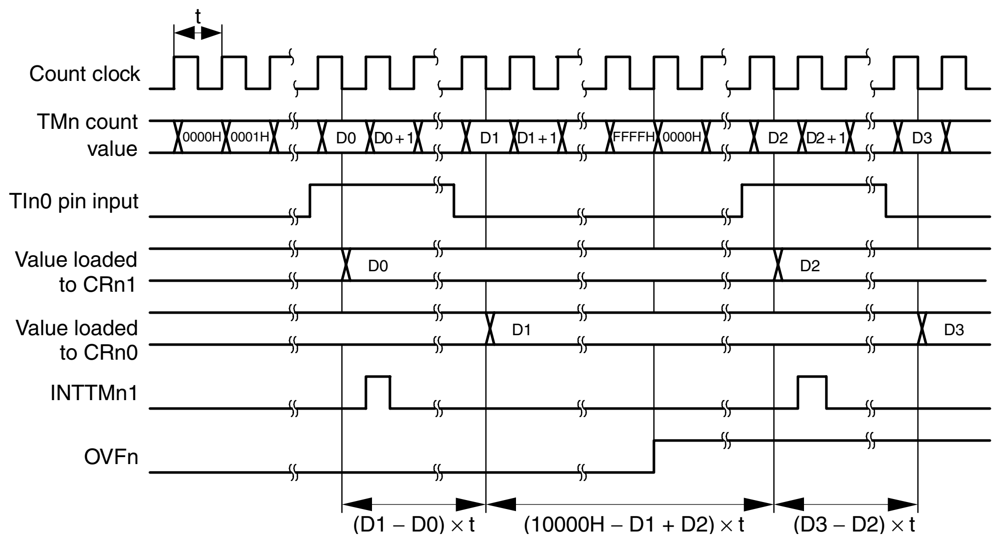


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



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 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 7-17**)

The edge is specified by the ESn00 and ESn01 bits of prescaler mode register n0 (PRMn0). 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 n0, n1 (PRMn0, 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.

Caution If the valid edge of the TIn0 pin is specified to be both the rising and falling edges, 16-bit capture/compare register n0 (CRn0) cannot perform its capture operation.

Figure 7-16. 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

TMCn3, TMCn2 (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

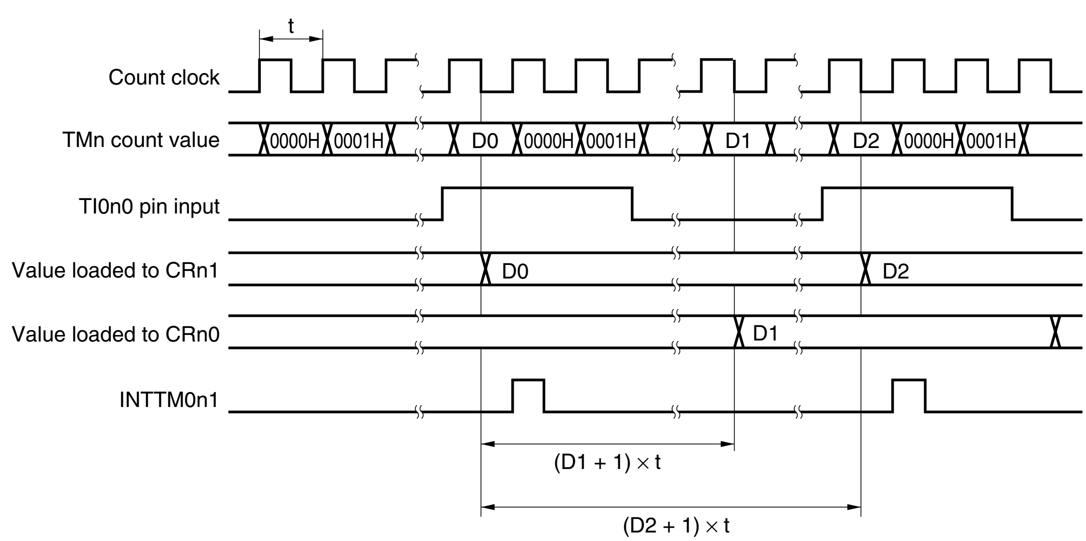
CRCn0 (1): CRn0 used as capture register

CRCn1 (1): Captures to CRn0 at edge reverse to valid edge of TIn0.

CRCn2 (1): 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 7-17. 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 n0 (PRMn0) has been input, TMn is incremented.

When the count value of TMn matches with 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 ESn00 and ESn01 bits of prescaler mode register n0 (PRMn0). The rising, falling, or both the rising and falling edges can be specified.

The valid edge is detected through sampling at a count clock cycle of $f_{XX}/2$, and the capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be removed.

Figure 7-18. 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

TMCn3, TMCn2 (1, 1): Clears and starts on match between TMn and CRn0.

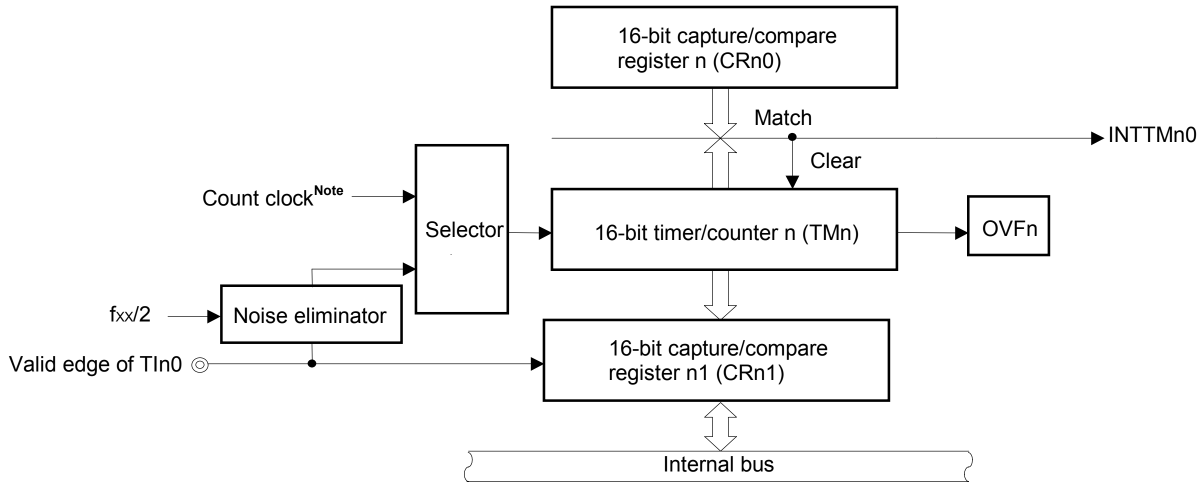
(b) Capture/compare control registers 0, 1 (CRC0, CRC1)

						CRCn2	CRCn1	CRCn0
CRCn	0	0	0	0	0	0/1	0/1	0

CRCn0 (0): CRn0 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 7-19. Configuration of External Event Counter

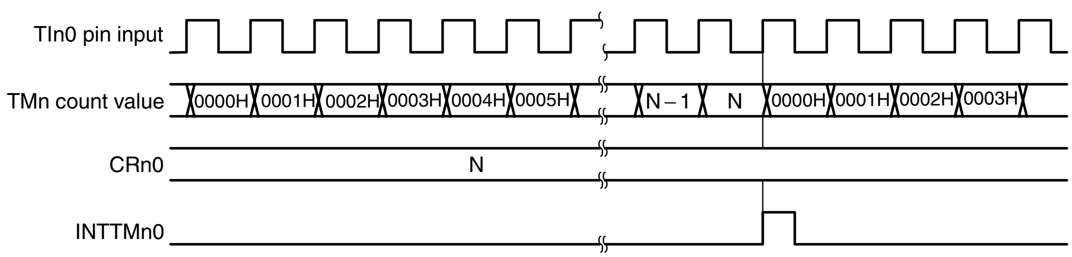


Note The count clock is set by the PRMn0 and PRMn1 registers.

Remarks 1. "◎—— " indicates a signal that can be directly connected to a port.

2. n = 0, 1

Figure 7-20. 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 the TOEn and TOCn1 bits 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.

Figure 7-21. 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.

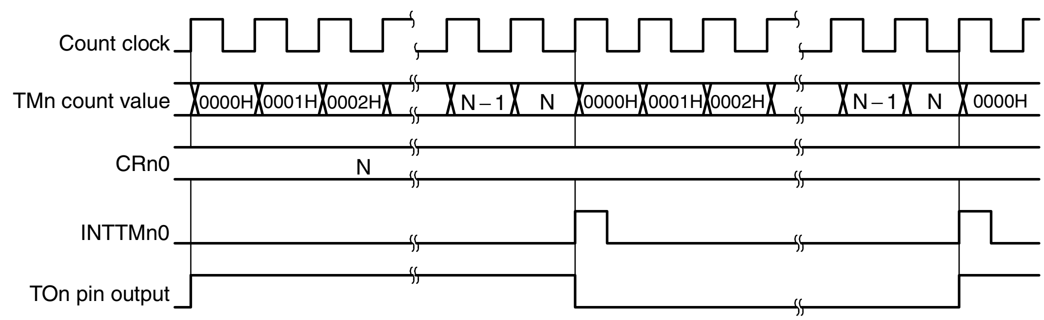
Specifies initial value of TOn output F/F.

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 7-22. 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 7-23, and by setting the OSPTn bit 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].

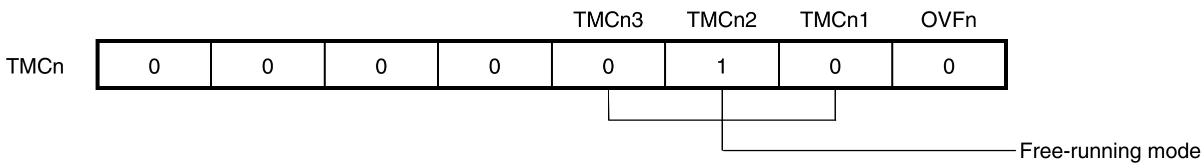
Even after the one-shot pulse has been output, TMn continues its operation. To stop TMn, TMCn must be reset to 00H.

Note This is an example when N < M. When N > M, the output is asserted by CRn0 and deasserted by CRn1.

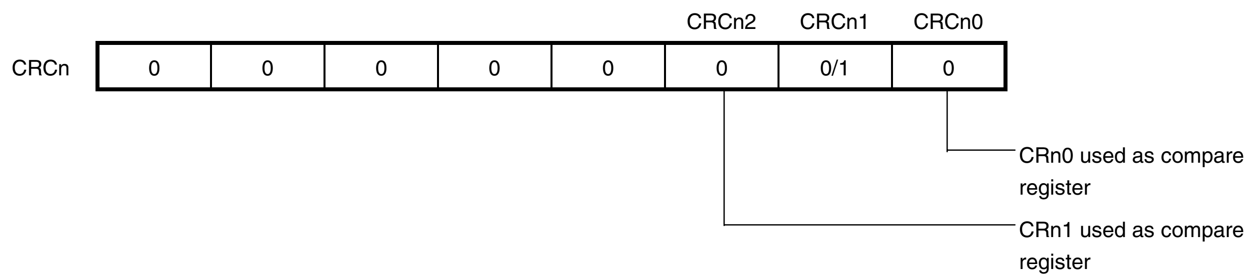
Caution Do not set OSPTn to 1 while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is complete.

Figure 7-23. Control Register Settings for One-Shot Pulse Output with Software Trigger

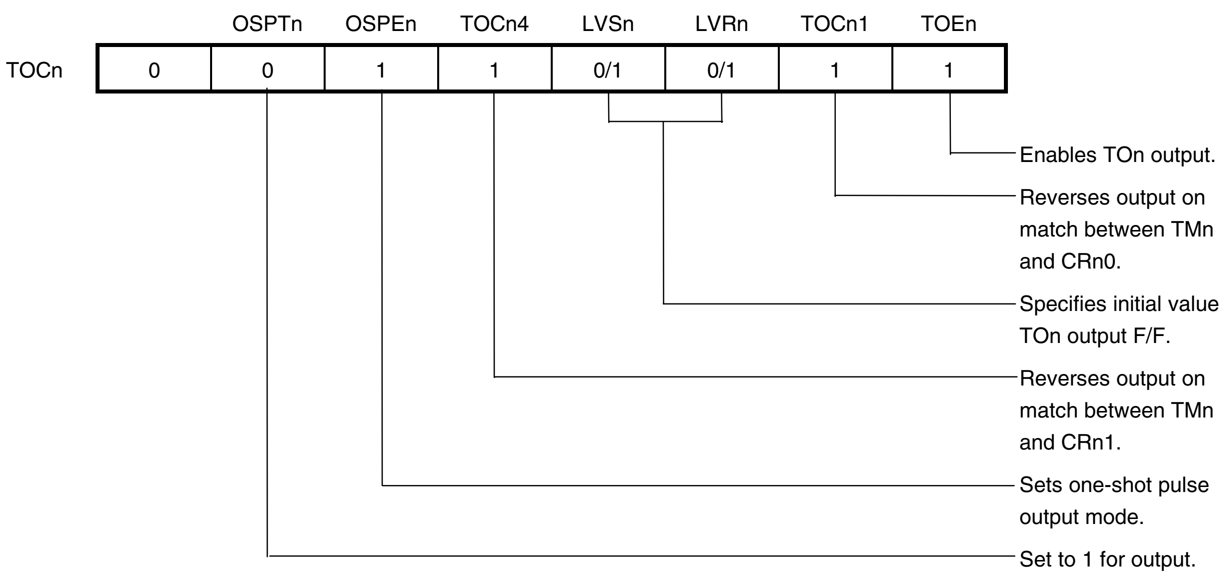
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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



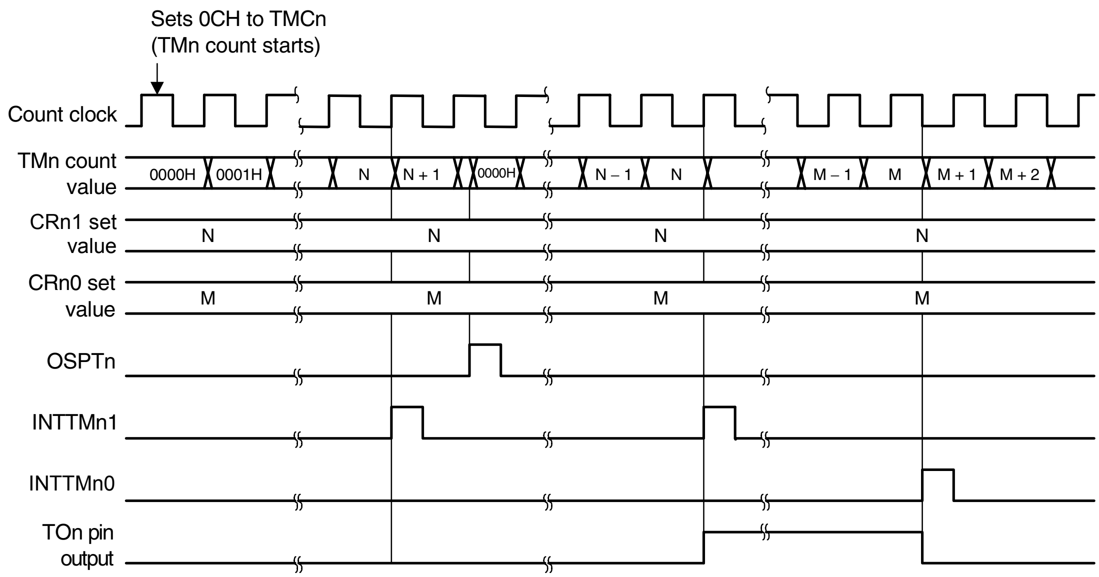
(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)



Caution Do not set CRn0 and CRn1 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 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 7-25, 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 n0 (PRMn0). 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).

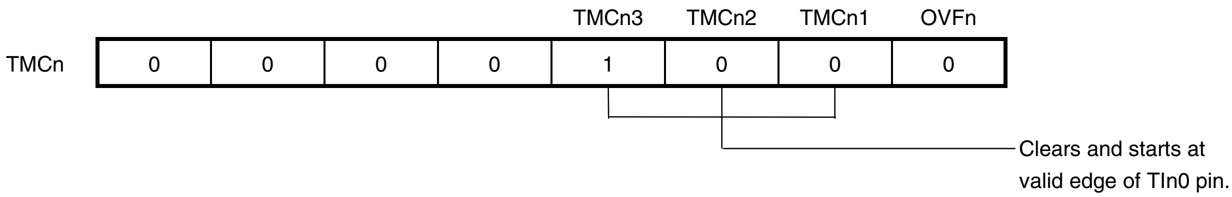
After that, the output is deasserted at the count value (M) set in advance to 16-bit capture/compare register n0 (CRn0)[Note].

Note This is an example when N < M. When N > M, the output is asserted by CRn0 and deasserted by CRn1.

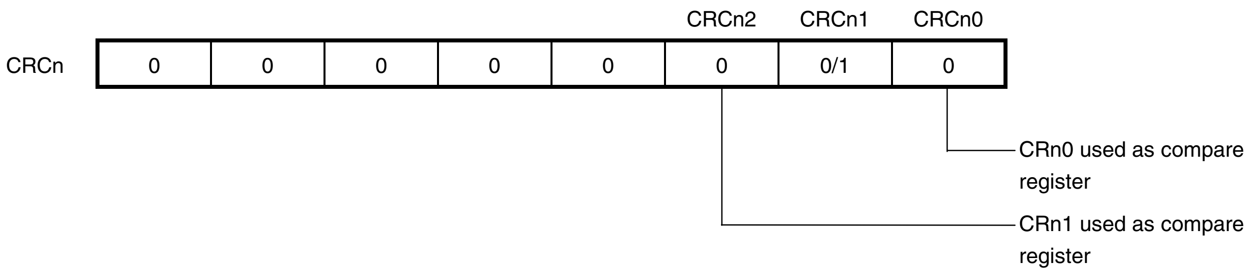
★ **Caution** **If an external trigger occurs while a one-shot pulse is being output, the 16-bit timer/event counter is cleared and started and a one-shot pulse is output again.**

Figure 7-25. Control Register Settings for One-Shot Pulse Output with External Trigger

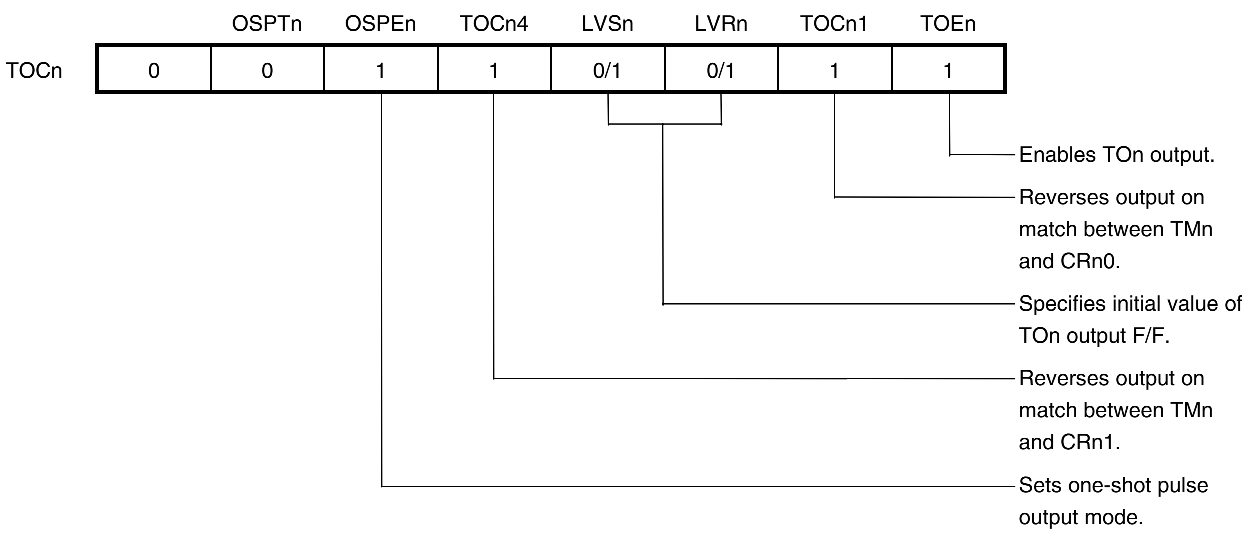
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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



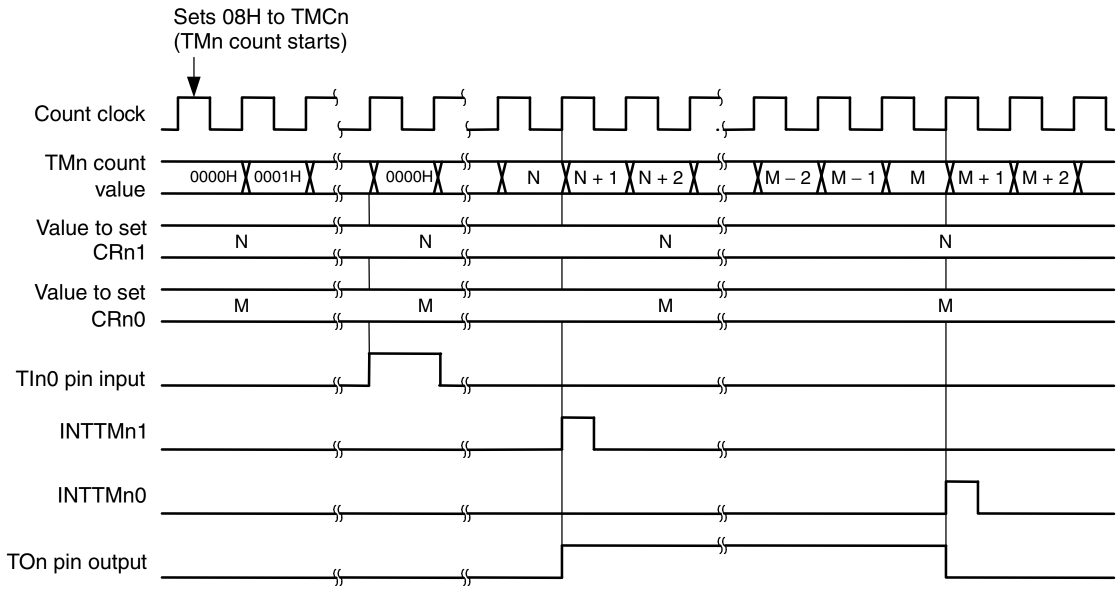
(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)



Caution Do not set CRn0 and CRn1 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 7-26. 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).**

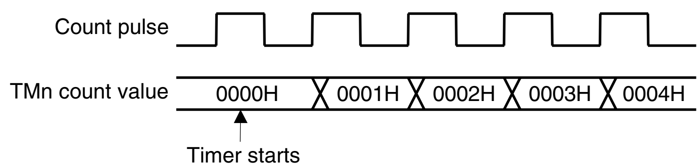
Remark n = 0, 1
N < M

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 7-27. Start Timing of 16-Bit Timer Register n



Remark n = 0, 1

(2) 16-bit capture/compare register setting (clear & start mode on match between TMn and CRn0)

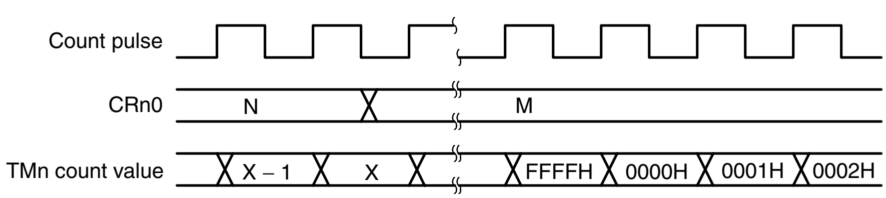
Set 16-bit capture/compare registers n0, n1 (CRn0, CRn1) to a value other than 0000H (the 1-pulse count operation is disabled when these registers are used as event counters).

(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 then restarted after the value of CRn0 has been changed.

Figure 7-28. Timing After Changing Compare Register During Timer Count Operation



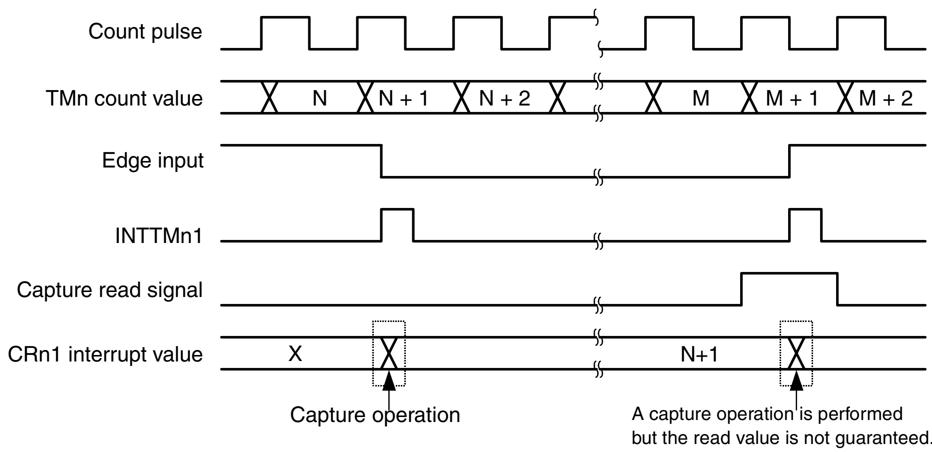
Remarks 1. N > X > M

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 the capture operation, but this read value is not guaranteed. However, the interrupt request signal (INTTMn1) is set as a result of detection of the valid edge.

Figure 7-29. 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 the TMCn2 and TMCn3 bits of 16-bit timer mode control register n to 0, 0. Set the valid edge by using the ESn00 and ESn01 bits of prescaler mode register n0 (PRMn0).

(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. Do not output the one-shot pulse again until the current one-shot pulse output ends.

(b) One-shot pulse output with external trigger

★ If an external trigger occurs while a one-shot pulse is being output, the 16-bit timer/event counter is cleared and started and a one-shot pulse is output again.

(c) One-shot pulse output function

When using the one-shot pulse output function of timer 0 or 1 by software trigger, do not change the level of the TIn0 pin or the pin multiplexed with it.

Even in this case, the external trigger remains valid. Consequently, the timer is cleared and started by the level of the TIn0 pin or the pin multiplexed with it, and a pulse is output when it is not expected.

(7) Operation of OVFn bit

(a) OVFn bit set

The OVFn bit 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 or the mode in which it is cleared and started by the valid edge of TIn0.

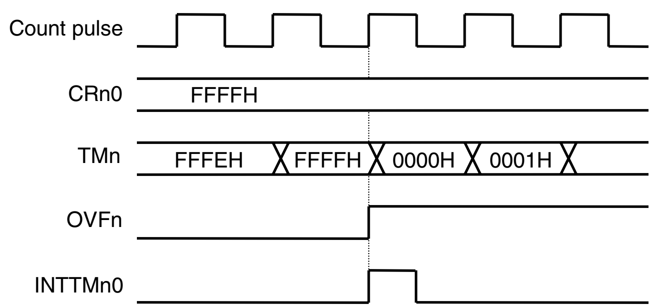
↓

Set CRn0 to FFFFH.

↓

When TMn is cleared from FFFFH to 0000H on a match with the CRn0 register

Figure 7-30. Operation Timing of OVFn Bit



Remark n = 0, 1

(b) Clear OVFn bit

Even if the OVFn bit is cleared before the next count clock is counted (before TMn become 0001H) after TMn has overflowed, the OVFn flag is set again and the clear becomes invalid.

(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 CRn0 and CRn1 is undefined.

(b) If the match timings 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 timings of the write period and 16-bit timer register n (TMn) conflict, do not write to CRn0 and CRn1 close to the match timing.

(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) Acknowledgment 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.

(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 valid edge of TIn0

If the CRn0 register capture trigger is set to the inverse phase of the valid edge of TIn0 and both the 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.

(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.

(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 registers 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.

7.3 8-Bit Timer (TM2 to TM7)

★ 7.3.1 Outline

- 8-bit compare registers: 8 (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 (TIm) count enabled
- Timer outputs that operate on match detection: 1 each (TOm)
  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 2 and 3 mode registers (PM2, 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 7,
m = 2 to 5

7.3.2 Functions

8-bit timer n has the following two modes (n = 2 to 7).

- Mode using timer alone (individual mode)
- Mode using cascade connection (16-bit resolution: cascade connection mode)

Caution Do not access following registers when not using the cascade connection.
- **16-bit counters (TM23, TM45, TM67)**
- **16-bit compare registers (CR23, CR45, CR67)**

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)

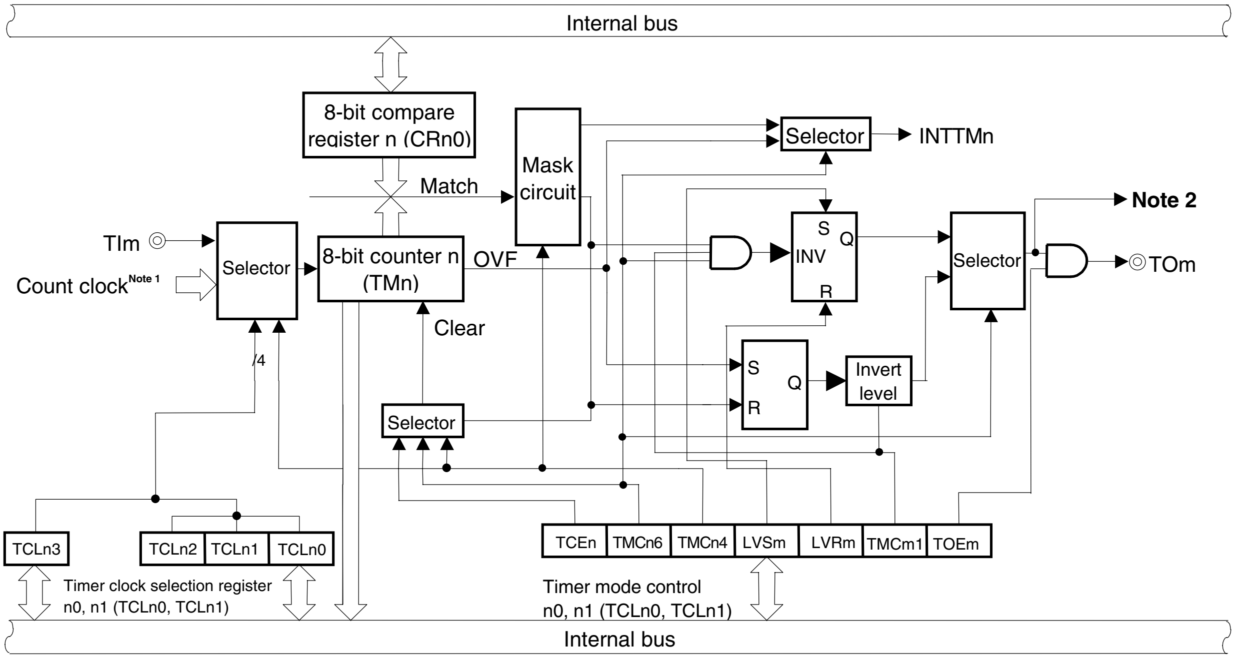
The timer operates as a 16-bit timer/event counter by connecting TM2 and TM3 or TM4 and TM5 in cascade.
It can have the following functions.

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

The timer operates as a 16-bit timer/event counter by connecting TM6 and TM7 in cascade.

- Interval timer with 16-bit resolution

Figure 7-31. Block Diagram of TM2 to TM7



Notes 1. The count clock is set by the TCLn register.

2. Clock of serial interface (TM2 and TM3 only)

Remarks 1. "——◎" is a signal that can be directly connected to a port.

2. n = 2 to 7, m = 2 to 5

7.3.3 Configuration

Timer n includes the following hardware.

Table 7-5. Configuration of Timers 2 to 7

Item	Configuration
Timer registers	8-bit counters 2 to 7 (TM2 to TM7) 16-bit counters 23, 45, 67 (TM23, TM45, TM67): Only when connecting in cascade
Registers	8-bit compare registers 2 to 7 (CR20 to CR70) 16-bit compare registers 23, 45, 67 (CR23, CR45, CR67): Only when connecting in cascade
Timer outputs	TO2 to TO5
Control registers	Timer clock selection registers 20 to 70 and 21 to 71 (TCL20 to TCL70 and TCL21 to TCL71) 8-bit timer mode control registers 2 to 7 (TMC2 to TMC7)

(1) 8-bit counters 2 to 7 (TM2 to TM7)

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.

TM2 and TM3 or TM5 and TM6 can be connected in cascade and used as 16-bit timers.

When TMm and TMm+1 are connected in cascade and used as a 16-bit timer, they can be read by a 16-bit memory manipulation instruction. However, since they are connected via the internal 8-bit bus, TMm and TMm+1 are read separately. Consequently, they should be read twice and compared to allow for count variation.

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 lower timers (TM2, TM4, TM6) is cleared.

Remark n = 2 to 7
m = 2, 4, 6

(2) 8-bit compare registers 2 to 7 (CR20 to CR70)

The CRn0 register is set by an 8-bit memory manipulation instruction.

The value set in CRn0 is always compared to the count 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 TMm and TMm+1 are connected in cascade and used 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. This register generates an interrupt request (INTTMm) when the counter value and register value are compared as 16 bits and match. Since the INTTMm+1 interrupt request is also generated at that time, mask the INTTMm+1 interrupt request when TMm and TMm+1 are used connected in cascade.

$\overline{\text{RESET}}$ 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 7
m = 2, 4, 6

7.3.4 Timer n control register

The following two types of registers control timer n.

- Timer clock selection registers n0, n1 (TCLn0, TCLn1)
- 8-bit timer mode control register n (TMCn)

(1) Timer clock selection registers 20 to 71 and 21 to 71 (TCL20 to TCL70 and TCL21 to TCL71)

These registers set the count clock of timer n.

TCLn0 and TCLn1 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF244H, FFFFF254H

	7	6	5	4	3	2	1	0
TCLn0	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	fxx	
					20 MHz[Note]	12.58 MHz
0	0	0	0	TIn falling edge	–	–
0	0	0	1	TIn rising edge	–	–
0	0	1	0	fxx/4	200 ns	318 ns
0	0	1	1	fxx/8	400 ns	636 ns
0	1	0	0	fxx/16	800 ns	1.3 μs
0	1	0	1	fxx/32	1.6 μs	2.5 μs
0	1	1	0	fxx/128	6.4 μs	10.2 μs
0	1	1	1	fxx/512	25.6 μs	40.7 μs
1	0	0	0	Setting prohibited	–	–
1	0	0	1	Setting prohibited	–	–
1	0	1	0	fxx/64	3.2 μs	5.1 μs
1	0	1	1	fxx/256	12.8 μs	20.3 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	–	–

Note Only in the V850/SB1.

Cautions 1. When TCLn0 and TCLn1 are overwritten by different data, write after temporarily stopping the timer.

2. Always set bits 3 to 7 to in TCLn0 to 0, and bits 1 to 7 in TCLn1 to 0.

Remark When connected in cascade, the settings of TCL33 to TCL30 of TM3 are invalid.

After reset: 00H R/W Address: FFFFF264H, FFFFF274H

	7	6	5	4	3	2	1	0
TCLn0	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[Note]	12.58 MHz
0	0	0	0	TIn falling edge	–	–
0	0	0	1	TIn rising edge	–	–
0	0	1	0	$f_{XX}/4$	200 ns	318 ns
0	0	1	1	$f_{XX}/8$	400 ns	636 ns
0	1	0	0	$f_{XX}/16$	800 ns	1.3 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	2.5 μs
0	1	1	0	$f_{XX}/128$	6.4 μs	10.2 μs
0	1	1	1	f_{XT} (Subclock)	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	5.1 μs
1	0	1	1	$f_{XX}/256$	12.8 μs	20.3 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	–	–

Note Only in the V850/SB1.

Cautions 1. When TCLn0 and TCLn1 are overwritten by different data, write after temporarily stopping the timer.

2. Always set bits 3 to 7 of TCLn0 and bits 1 to 7 of TCLn1 to 0.

Remark When connected in cascade, the settings of TCL53 to TCL50 of TM5 are invalid.

After reset: 00H R/W Address: FFFFF284H, FFFFF294H

	7	6	5	4	3	2	1	0
TCLn0 (n = 6, 7)	0	0	0	0	0	TCLn2	TCLn1	TCLn0

After reset: 00H R/W Address: FFFFF28EH, FFFFF29EH

	7	6	5	4	3	2	1	0
TCLn1 (n = 6, 7)	0	0	0	0	0	0	0	TCLn3

TCLn3	TCLn2	TCLn1	TCLn0	Count clock selection		
				Count clock	fxx	
					20 MHz[Note]	12.58 MHz
0	0	0	0	Setting prohibited	–	–
0	0	0	1	Setting prohibited	–	–
0	0	1	0	fxx/4	200 ns	318 ns
0	0	1	1	fxx/8	400 ns	636 ns
0	1	0	0	fxx/16	800 ns	1.3 μs
0	1	0	1	fxx/32	1.6 μs	2.5 μs
0	1	1	0	fxx/64	3.2 μs	5.1 μs
0	1	1	1	fxx/128	6.4 μs	10.2 μs
1	0	0	0	Setting prohibited	–	–
1	0	0	1	Setting prohibited	–	–
1	0	1	0	fxx/256	12.8 μs	20.3 μs
1	0	1	1	fxx/512	25.6 μs	40.7 μs
1	1	0	0	Setting prohibited	–	–
1	1	0	1	Setting prohibited	–	–
1	1	1	0	Setting prohibited	–	–
1	1	1	1	TM0 overflow signal	–	–

Note Only in the V850/SB1.

Cautions 1. When TCLn0 and TCLn1 are overwritten by different data, write after temporarily stopping the timer.

2. Always set bits 3 to 7 of TCLn0 and bits 1 to 7 of TCLn1 to 0.

Remark When connected in cascade, the settings of TCL73 to TCL70 of TM7 are invalid.

(2) 8-bit timer mode control registers 2 to 7 (TMC2 to TMC7)

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 an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 04H (although the state of hardware is initialized to 04H, 00H is read when reading).

After reset: 04H R/W Address: TMC2 FFFFF246H TMC5 FFFFF276H
TMC3 FFFFF256H TMC6 FFFFF286H
TMC4 FFFFF266H TMC7 FFFFF296H

	<7>	6	5	4	<3>	<2>	1	<0>
TMCn	TCEn	TMCn6	0	TMCn4	LVSm	LVRm	TMCm1	TOEm

(n = 2 to 7, m = 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 & 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, 6)
1	Cascade connection mode (connection to lower timer)

LVSm	LVRm	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

TMCm1	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

TOEm	Timer output control
0	Disable output (port mode)
1	Enable output

Cautions 1. When using as the timer output pin (TOm), set the port value to 0 (port mode output). An ORed value of the timer output value is output.

2. Since TOm and TIm share the same pin, only one of the functions can be used.

Remarks 1. In the PWM mode, the PWM output is set to the inactive level by TCEm = 0.

2. If LVSm and LVRm 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 TMn continues counting. At the same time, an interrupt request signal (INTTMn) is generated.

The TMn count clock can be selected by the TCLn0 to TCLn2 bits of timer clock selection register n0 (TCLn0) and by the TCLn3 bit of timer clock selection register n1 (TCLn1) (n = 2 to 7).

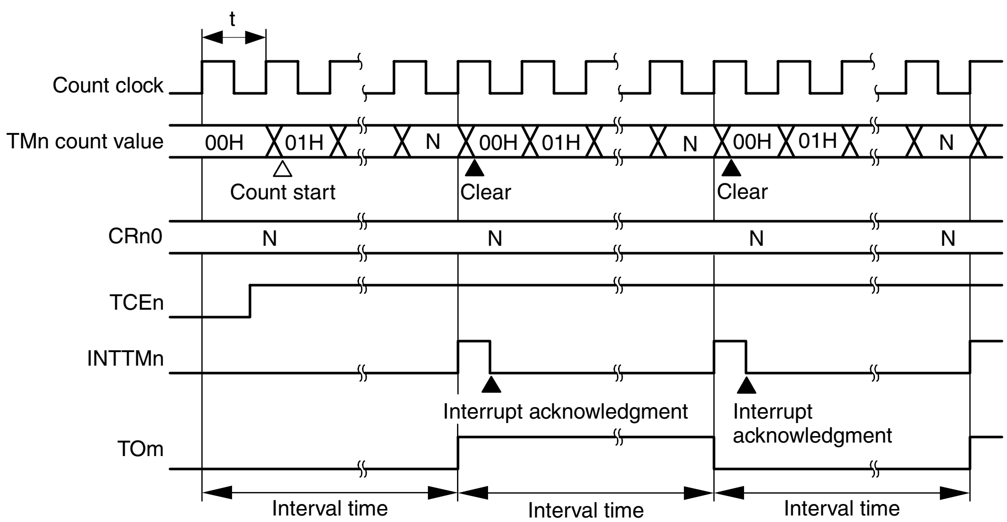
Setting method

(1) Set each register.
- TCLn0, TCLn1: Select the count clock.
- CRn0: Compare value
- TMCn: Selects the clear and start mode when TMn and CRn0 match.
  (TMCn = 0000xx11B, 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 7-32. Timing of Interval Timer Operation (1/3)

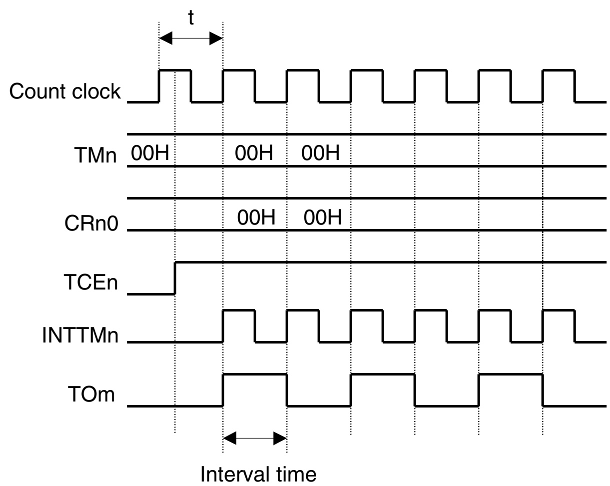
Basic operation



Remarks 1. Interval time = (N + 1) × t; N = 00H to FFH
2. n = 2 to 7,
m = 2 to 5

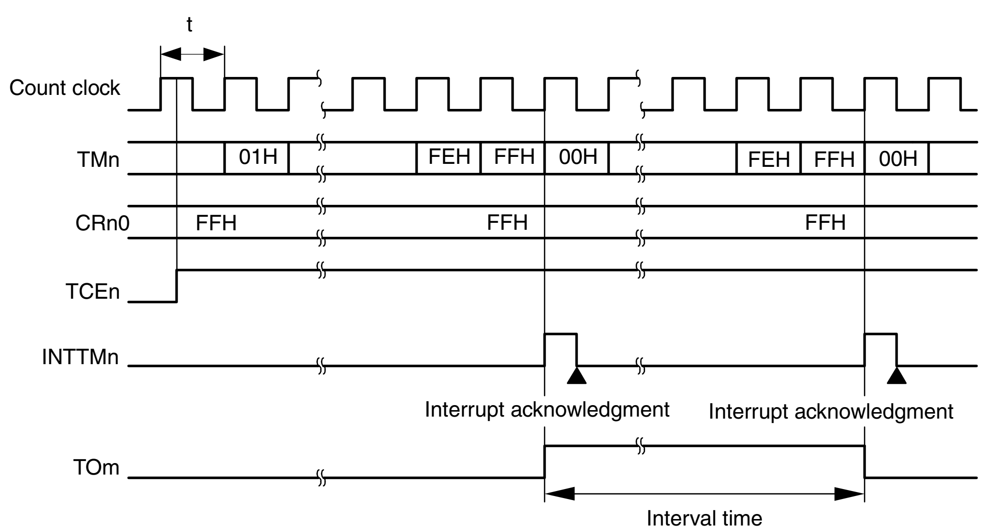
Figure 7-32. Timing of Interval Timer Operation (2/3)

When CRn0 = 00H



Remark n = 2 to 7,
m = 2 to 5

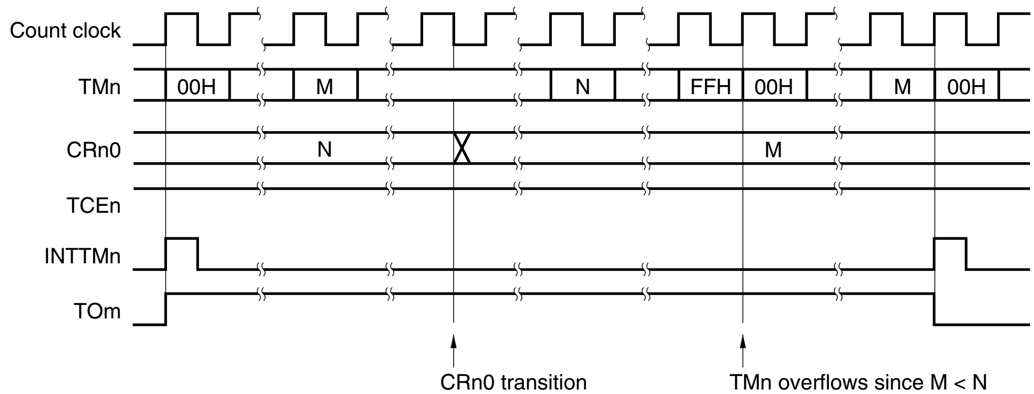
When CRn0 = FFH



Remark n = 2 to 7,
m = 2 to 5

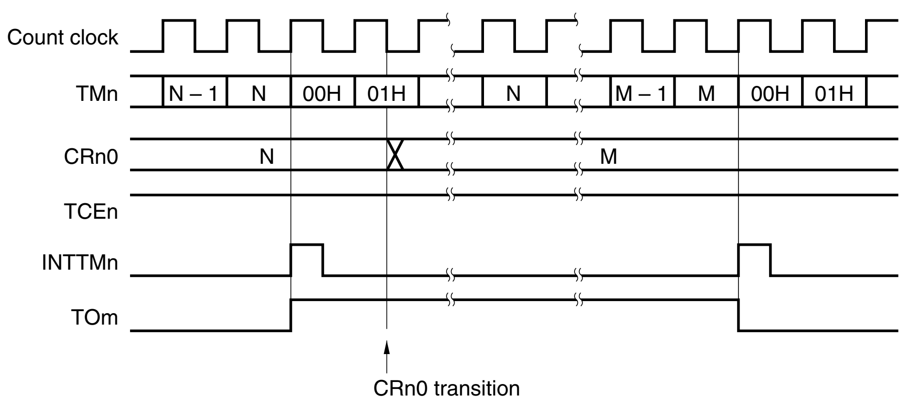
★ **Figure 7-32. Timing of Interval Timer Operation (3/3)**

Operated by CRn0 transition (M < N)



Remark n = 2 to 7,
m = 2 to 5

Operated by CRn0 transition (M > N)



Remark n = 2 to 7,
m = 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.

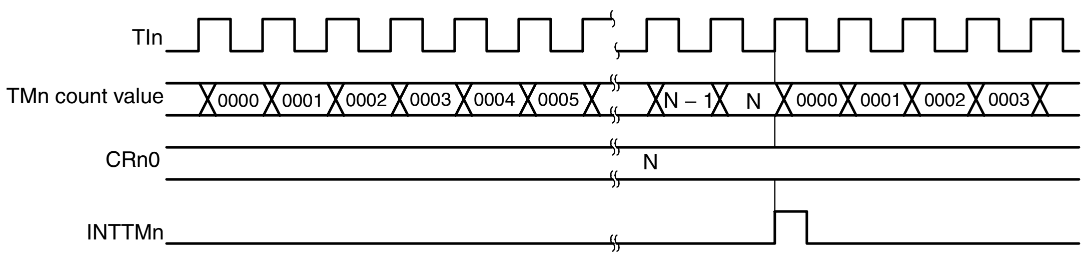
Each time a valid edge specified by timer clock selection register n0, n1 (TCLn0, TCLn1) is input, TMn is incremented. The edge setting can be selected as either the rising or falling edge.

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 7-33. Timing of External Event Counter Operation (with Rising Edge Specified)



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
- TCLn0, TCLn1: Select the count clock
- CRn0: Compare value
- TMCn: Clear and start mode when TMn and CRn0 match

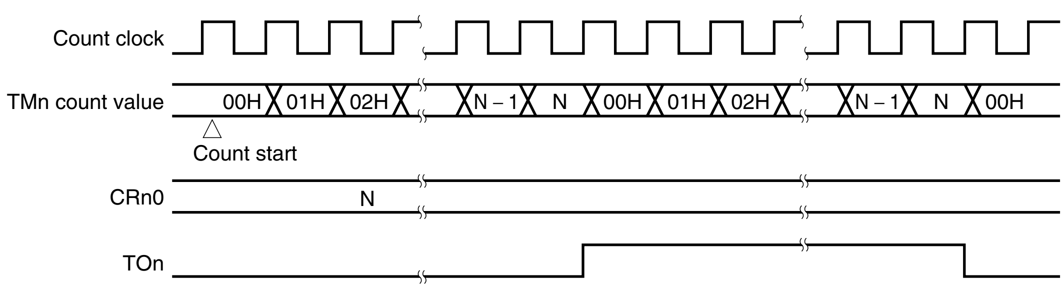
LVSn	LVRn	Setting State of Timer Output Flip-Flop
1	0	High level output
0	1	Low level output

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.

Remark n = 2 to 5

Figure 7-34. Square Wave Output Operation Timing



Note The initial value of TOn output can be set by the LVSn and LVRn bits of the TMCn register.

★ **Remarks 1.** Square-wave output frequency = 1/2 (N + 1)
2. n = 2 to 5

7.4.4 Operation as 8-bit PWM output

By setting the TMCn6 bit 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 the TMCn1 bit in TMCn.

The count clock can be selected by the TCLn0 to TCLn2 bits of timer clock selection register n0 (TCLn0) and by the TCLn3 bit of timer clock selection register n1 (TCLn1).

The PWM output can be enabled and disabled by the TOEn bit 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 selection register n0, n1 (TCLn0, TCLn1).
(4) Set the active level in TMCn1 bit of TMCn.
(5) If TCEn bit of TMCn is set to 1, counting starts. When counting stops, set TCEn to 0.

PWM output operation

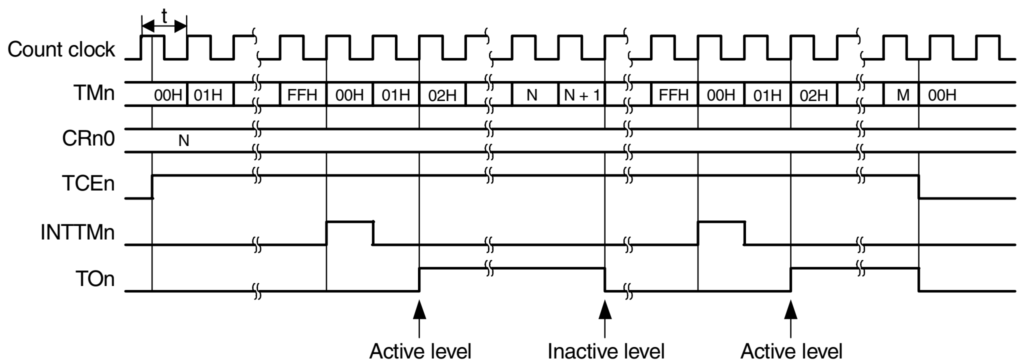
(1) When counting starts, the PWM output (output from TOn) outputs the inactive level until an overflow occurs.
(2) When the overflow occurs, the active level specified in step (1) in the setting method is output. The active level is output until CRn0 and the count of 8-bit counter n (TMn) match.
(3) 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

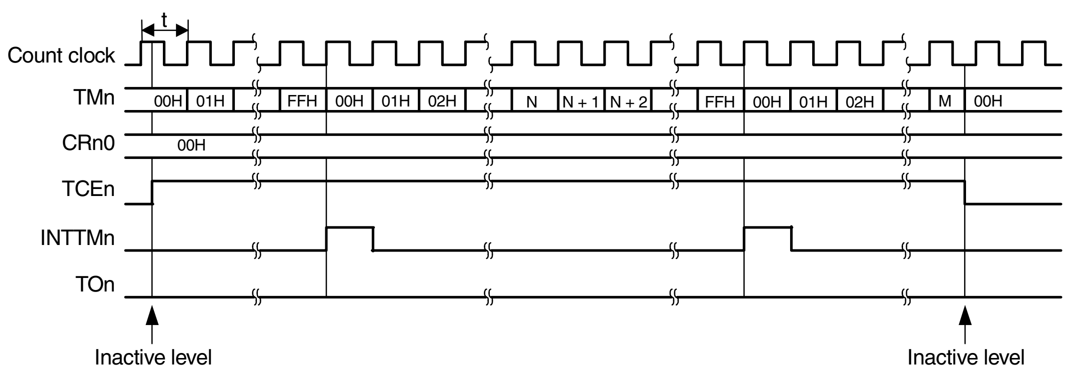
(a) Basic operation of PWM output

Figure 7-35. Timing of PWM Output

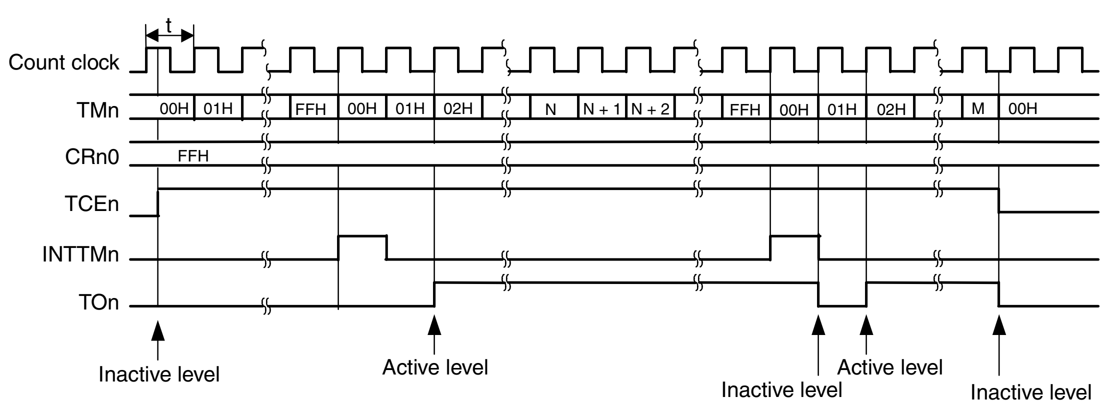
Basic operation (active level = H)



When CRn0 = 00H



When CRn0 = FFH



★ **Remarks 1.** PWM frequency = 2^8t

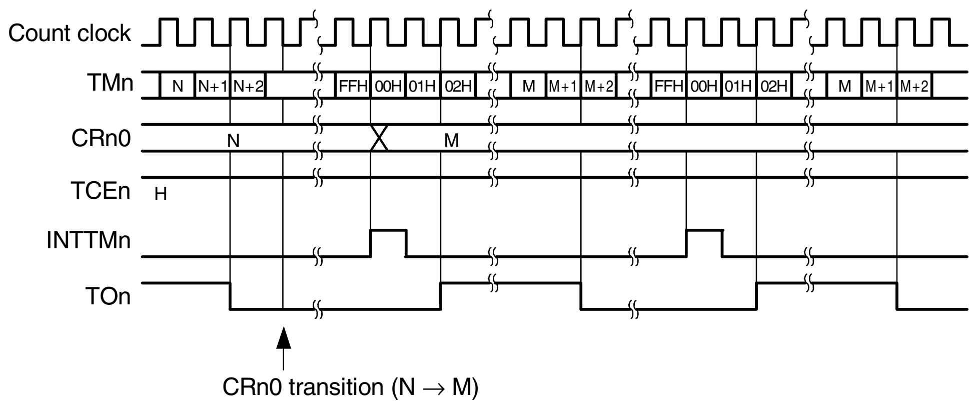
Duty = N/256 (N: CRn0 set value)

2. n = 2 to 5

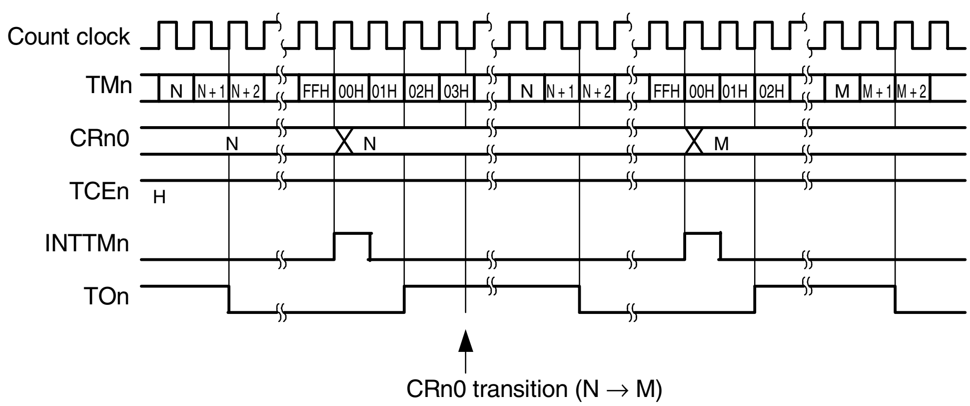
(b) Operation based on CRn0 transitions

Figure 7-36. Timing of Operation Based on CRn0 Transitions

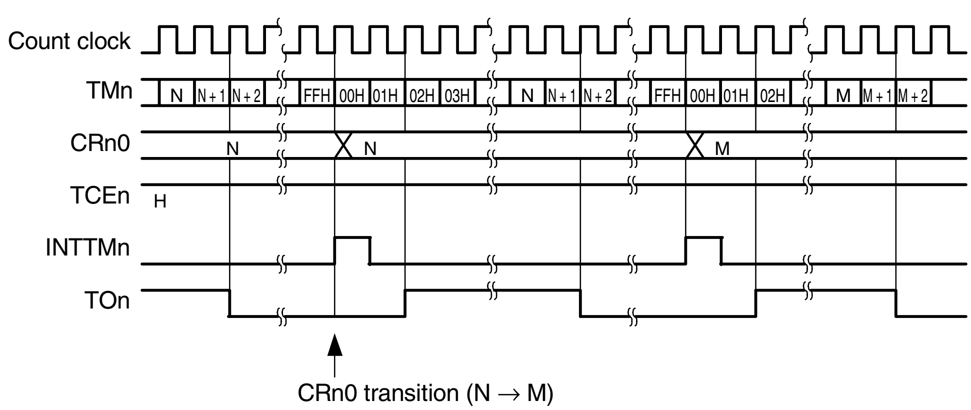
When the CRn0 value changes from N to M before TMn overflows



When the CRn0 value changes from N to M after TMn overflows



When the CRn0 value changes from N to M within 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/SB1 and V850/SB2 provide 16-bit registers that can be used only when connected in cascade.
The following registers are available.

TM2, TM3 cascade connection:	16-bit counter TM23 (Address: FFFFF24AH)
	16-bit compare register CR23 (Address: FFFFF24CH)
TM4, TM5 cascade connection:	16-bit counter TM45 (Address: FFFFF26AH)
	16-bit compare register CR45 (Address: FFFFF26CH)
TM6, TM7 cascade connection:	16-bit counter TM67 (Address: FFFFF28AH)
	16-bit compare register CR67 (Address: FFFFF28CH)

By setting the TMCm4 bit 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, 7).
With the count preset in 8-bit compare register n (CRn0) as the interval, the timer operates as an interval timer by repeatedly generating interrupts (n = 2 to 7).
The following shows a setting method when using TM2 and TM3. When using TM4 and TM5 or TM6 and TM7, substitute them for TM2 and TM3.

Setting method (TM2, TM3 cascade connection)

(1) Setting registers
- TCL20, TCL21: Select the count clock for TM2 (setting not necessary for TM3 because of cascade connection).
- CR20, CR30: Compare value (00H to FFH can be set for compare values)
- TMC2: Selects clear & start mode on a match of TM2 and CR2 (x: don't care)
  [TM2 → TMC2 = 0000xxx0B, TM3 → TMC3 = 0001xxx0B]

(2) Set the TCE3 bit of TMC3 to 1. After that, set the TCE2 bit of TMC2 to 1 to start the count operation.
(3) When the TM23 and CR23 values of the timer connected in cascade match, an interrupt request signal (INTTM2) is generated (TM2 and TM3 are cleared to 00H).
(4) IMTTM2 is then repeatedly generated at the same interval.

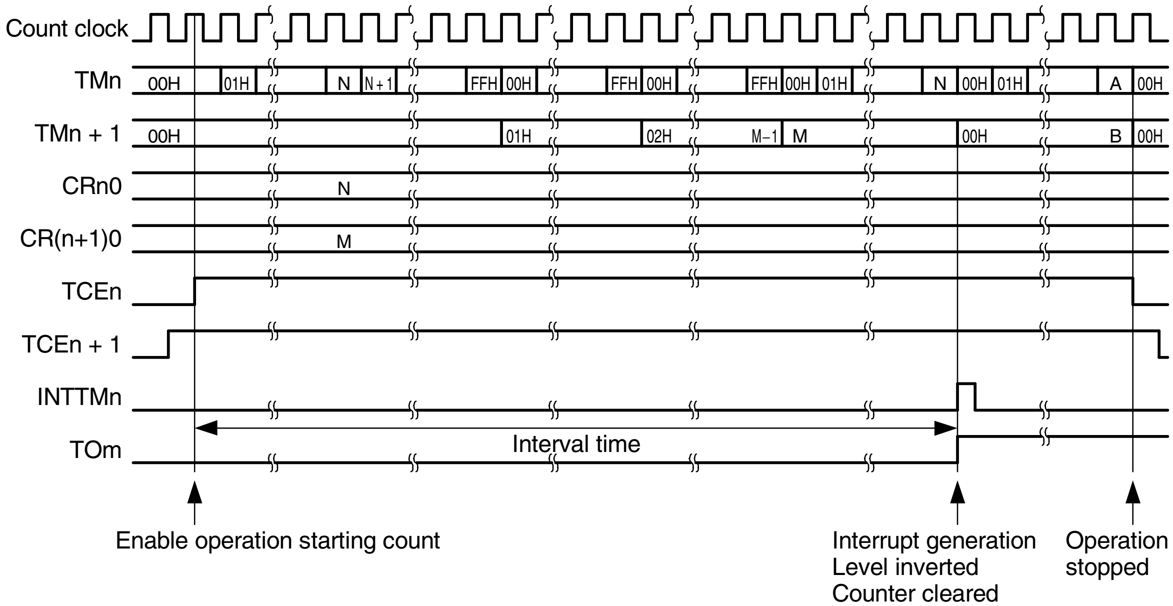
Cautions 1. The count operation can be started or stopped just by setting the TCE2 bit of TMC2.

2. When 8-bit timers (TM2, TM3) are connected in cascade and used as a 16-bit timer (TM23), change the setting value of the compare register (CR23) after stopping the count operation of the 8-bit timers connected in cascade.
If the value of CR23 is changed without stopping the timers, the values of the higher 8 bits (TM3) become undefined.

3. Even during cascade connection, the interrupt request signal (INTTM3) of higher timer 3 (TM3) is generated when the count value of higher timer 3 (TM3) matches CR30. Be sure to mask TM3 to disable this interrupt.

A timing example of the cascade connection mode with 16-bit resolution is shown below.

Figure 7-37. Cascade Connection Mode with 16-Bit Resolution



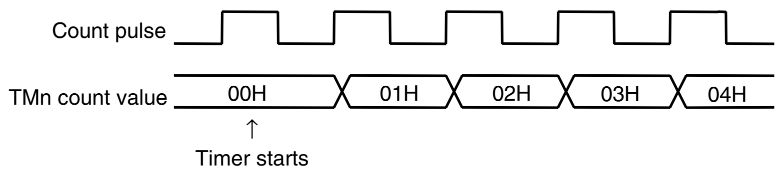
Remark n = 2, 4, 6,
m = 2, 4

7.4.6 Cautions

(1) Error when the 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 asynchronous to the count pulse.

Figure 7-38. Start Timing of Timer n

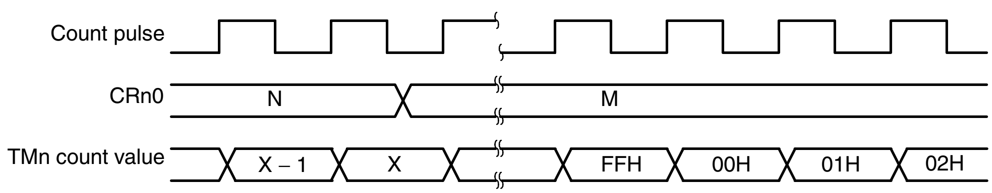


Remark n = 2 to 7

(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 counting 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 7-39. Timing After Compare Register Changes During Timer Count Operation



Remarks 1. N > X > M

2. n = 2 to 7

Caution Except when the TIm input is selected, always set TCEn = 0 before setting the stop state (m = 2 to 5).

(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 7).

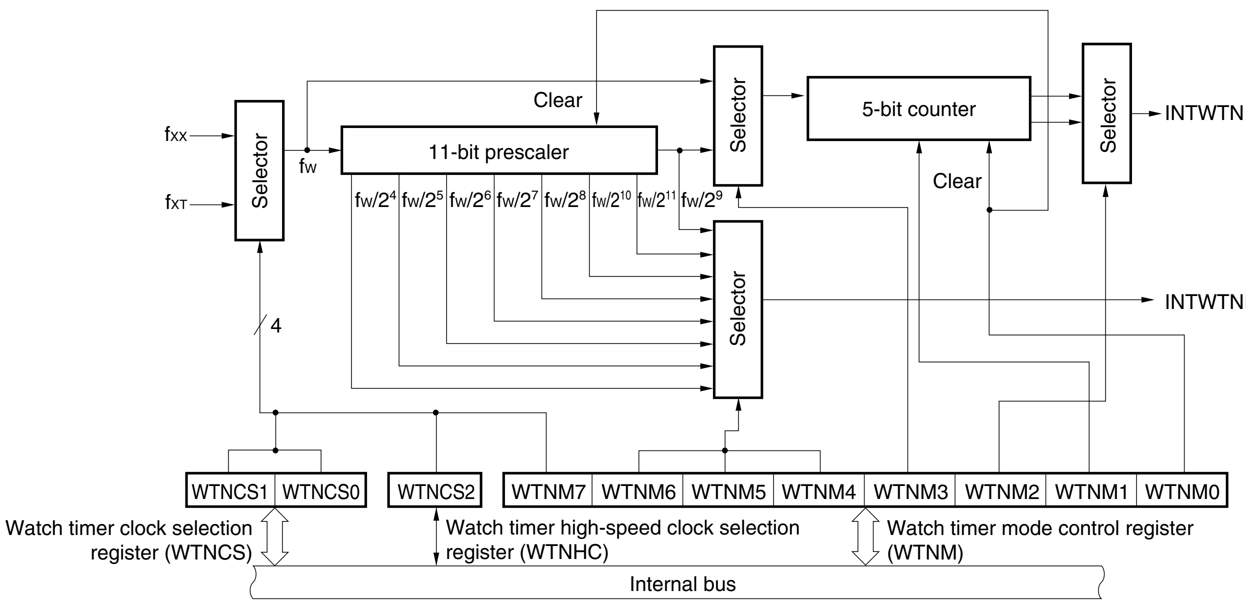
8.1 Function

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.

Figure 8-1. Block Diagram of Watch Timer



★ **Caution The WTNHC register is available only in the B versions of the V850/SB1, μPD703036H, 703036HY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037H, and 70F3037HY.**

Remark f_{XX}: Main clock frequency

f_{XT}: Subclock frequency

f_W: Watch timer clock frequency

(1) Watch timer

The watch timer generates an interrupt request (INTWTN) at time intervals of 0.5 seconds or 0.25 seconds by using the main clock or subclock.

(2) Interval timer

The watch timer generates an interrupt request (INTWTNI) at time intervals specified in advance.

Table 8-1. Interval Time of Interval Timer

Interval Time	f_{XT} = 32.768 kHz
$2^4 \times 1/f_W$	488 μs
$2^5 \times 1/f_W$	977 μs
$2^6 \times 1/f_W$	1.95 ms
$2^7 \times 1/f_W$	3.91 ms
$2^8 \times 1/f_W$	7.81 ms
$2^9 \times 1/f_W$	15.6 ms
$2^{10} \times 1/f_W$	31.2 ms
$2^{11} \times 1/f_W$	62.4 ms

Remark f_W: Watch timer clock frequency

8.2 Configuration

The watch timer includes the following hardware.

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control registers	Watch timer mode control register (WTNM) ★ Watch timer high-speed clock selection register (WTNHC)[Note] Watch timer clock selection register (WTNCS)

★ **Note** The WTNHC register is available only in the B versions of the V850/SB1, μPD703036H, 703036HY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037H, and 70F3037HY.

8.3 Watch Timer Control Registers

★ The watch timer mode control register (WTNM), watch timer high-speed clock selection register (WTNHC)Note, and watch timer clock selection register (WTNCS) control the watch timer. The watch timer should be operated after setting the count clock and interval time.

★ **Caution The WTNHC register is available only in the B versions of the V850/SB1, μPD703036H, 703036HY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037H, and 70F3037HY.**

(1) Watch timer mode control register (WTNM)

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 set time of the watch flag.

WTNM is set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears WTNM to 00H.

After reset: 00H R/W Address: FFFFF360H

	7	6	5	4	3	2	<1>	<0>
WTNM	WTNM7	WTNM6	WTNM5	WTNM4	WTNM3	WTNM2	WTNM1	WTNM0

WTNM6	WTNM5	WTNM4	Selection interval time of prescaler
0	0	0	2^4/fw (488 μs)
0	0	1	2^5/fw (977 μs)
0	1	0	2^6/fw (1.95 ms)
0	1	1	2^7/fw (3.91 ms)
1	0	0	2^8/fw (7.81 ms)
1	0	1	2^9/fw (15.6 ms)
1	1	0	2^{10}/fw (31.2 ms)
1	1	1	2^{11}/fw (62.4 ms)

WTNM3	WTNM2	Watch timer interrupt
0	0	2^{14}/fw (0.5 s)
0	1	2^{13}/fw (0.25 s)
1	0	2^5/fw (977 μs)
1	1	2^4/fw (488 μs)

WTM1	Control of operation of 5-bit counter
0	Clears after operation stops
1	Starts

WTNM0	Watch timer operation enable
0	Operation stopped (clears both prescaler and 5-bit counter)
1	Operation enabled

Remarks 1. fw: Watch timer clock frequency

2. Values in parentheses apply when fw = 32.768 kHz.

3. For the settings of WTNM7, refer to **(3) Watch timer clock selection register (WTNCS)**.

★ **(2) Watch timer high-speed clock selection register (WTNHC)**

This register selects the count clock of the watch timer.

The count clock is determined in combination with the WTNM7 bit of the WTNM register and the WTNCS1 and WTNCS0 bits of the watch timer clock selection register (WTNCS).

WTNHC is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears WTNHC to 00H.

After reset: 00H R/W Address: FFFFF366H

	7	6	5	4	3	2	1	0
WTNHC	0	0	0	0	0	0	0	WTNCS2

Caution The WTNHC register is available only in the B versions of the V850/SB1, μPD703036H, 703036HY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037H, and 70F3037HY.

★ **(3) Watch timer clock selection register (WTNCS)**

This register selects the count clock of the watch timer.

WTNCS is set using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears WTNCS to 00H.

Caution Do not change the contents of the WTNM, WTNHC, and WTNCS registers (interval time, watch timer interrupt time, count clock) during a watch timer operation.

After reset: 00H R/W Address: FFFFF364H

	7	6	5	4	3	2	1	0
WTNCS	0	0	0	0	0	0	WTNCS1	WTNCS0

WTNCS2[Note]	WTNCS1	WTNCS0	WTNM7	Selection of count clock	Main clock frequency
0	0	0	0	$f_{XX}/2^7$	4.194 MHz
0	0	0	1	f_{XT} (subclock)	–
0	0	1	0	$f_{XX}/3 \times 2^6$	6.291 MHz
0	0	1	1	$f_{XX}/2^8$	8.388 MHz
0	1	0	0	Setting prohibited	–
0	1	0	1	Setting prohibited	–
0	1	1	0	$f_{XX}/3 \times 2^7$	12.582 MHz
0	1	1	1	$f_{XX}/2^9$	16.777 MHz
1	0	1	0	$f_{XX}/3^2 \times 2^6$	18.874 MHz
Other than above				Setting prohibited	–

Note The WTNCS2 bit is available only in the B versions of the V850/SB1, μPD703036H, 703036HY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037H, and 70F3037HY.

Remark WTNM7 is bit 7 of the WTNM register.

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).

The watch timer generates an interrupt request at fixed time intervals.

The count operation of the watch timer is started when bits 0 (WTNM0) and 1 (WTNM1) of the watch timer mode control register (WTNM) are set to 1. When these bits are cleared to 0, the 11-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 WTNM1 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 WTNM0 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 (INTWTN).

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.

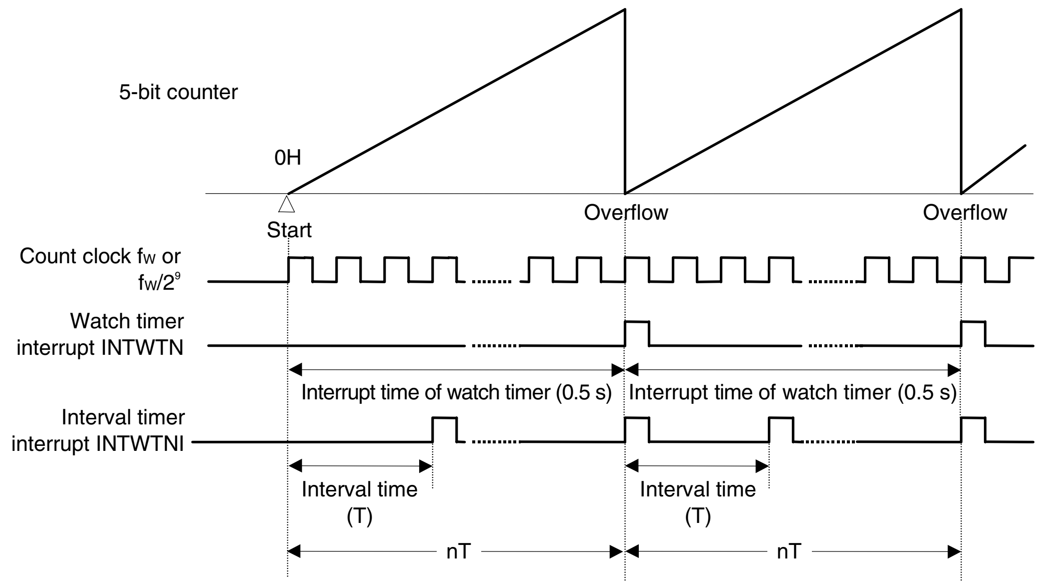
The interval time can be selected by bits 4 to 6 (WTNM4 to WTNM6) of the watch timer mode control register (WTNM).

Table 8-3. Interval Time of Interval Timer

WTNM6	WTNM5	WTNM4	Interval Time	f_W = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	488 μs
0	0	1	$2^5 \times 1/f_W$	977 μs
0	1	0	$2^6 \times 1/f_W$	1.95 ms
0	1	1	$2^7 \times 1/f_W$	3.91 ms
1	0	0	$2^8 \times 1/f_W$	7.81 ms
1	0	1	$2^9 \times 1/f_W$	15.6 ms
1	1	0	$2^{10} \times 1/f_W$	31.2 ms
1	1	1	$2^{11} \times 1/f_W$	62.4 ms

Remark f_W: Watch timer clock frequency

Figure 8-2. Operation Timing of Watch Timer/Interval Timer



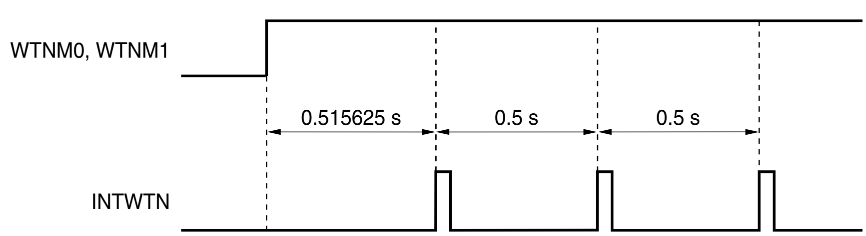
Remark fw: Watch timer clock frequency
( ): fw = 32.768 kHz
n: Interval timer operation count

8.4.3 Cautions

It takes some time to generate the first watch timer interrupt request (INTWTN) after operation is enabled (WRNM1 and WTNM0 bits of WTNM register = 1).

Figure 8-3. Watch Timer Interrupt Request (INTWTN) Generation (Interrupt Period = 0.5 s)

It takes 0.515625 s to generate the first INTWTN ($2^9 \times 1/32.768 = 0.015625$ s longer). INTWTN is then generated every 0.5 s.



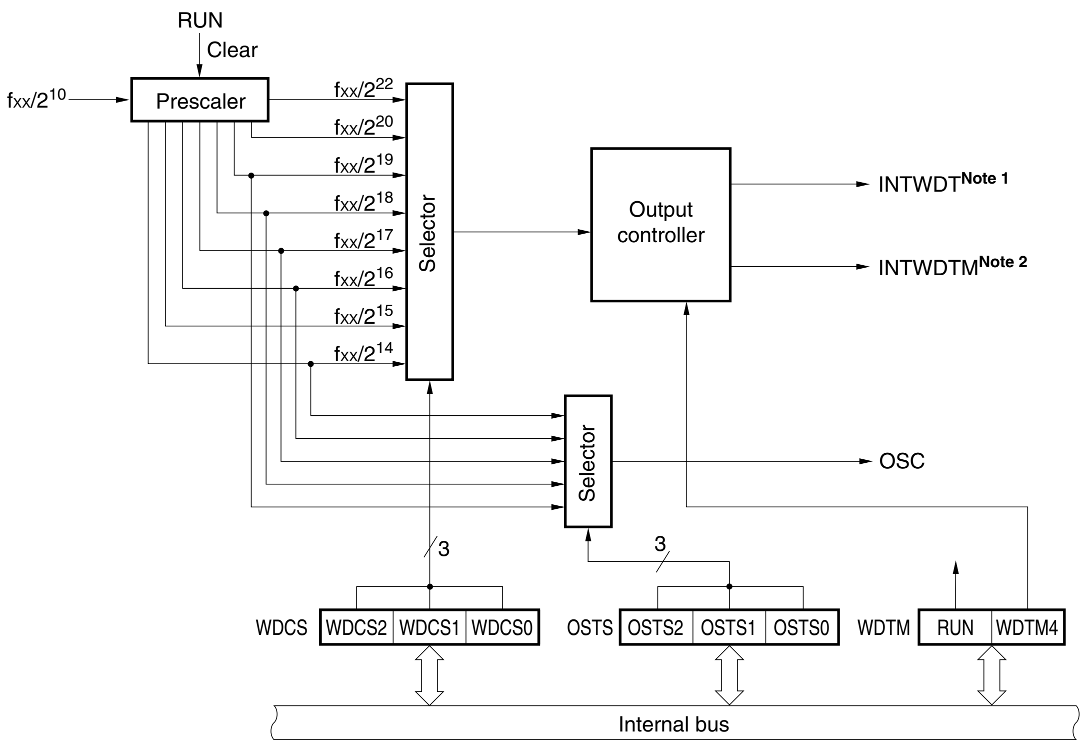
9.1 Functions

The watchdog timer has the following functions.

- 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 an inadvertent program loop. When an inadvertent program loop is detected, a non-maskable interrupt can be generated.

Table 9-1. Inadvertent Program Loop Detection Time of Watchdog Timer

Clock	Inadvertent Program Loop Detection Time	
	f_{XX} = 20 MHz[Note]	f_{XX} = 12.58 MHz
$2^{14}/f_{XX}$	819.2 μs	1.3 ms
$2^{15}/f_{XX}$	1.6 ms	2.6 ms
$2^{16}/f_{XX}$	3.3 ms	5.2 ms
$2^{17}/f_{XX}$	6.6 ms	10.4 ms
$2^{18}/f_{XX}$	13.1 ms	20.8 ms
$2^{19}/f_{XX}$	26.2 ms	41.6 ms
$2^{20}/f_{XX}$	52.4 ms	83.3 ms
$2^{22}/f_{XX}$	209.7 ms	333.4 ms

Note Only in the V850/SB1.

(2) Interval timer mode

Interrupts are generated at a preset time interval.

Table 9-2. Interval Time of Interval Timer

Clock	Interval Time	
	f_{XX} = 20 MHz[Note]	f_{XX} = 12.58 MHz
$2^{14}/f_{XX}$	819.2 μs	1.3 ms
$2^{15}/f_{XX}$	1.6 ms	2.6 ms
$2^{16}/f_{XX}$	3.3 ms	5.2 ms
$2^{17}/f_{XX}$	6.6 ms	10.4 ms
$2^{18}/f_{XX}$	13.1 ms	20.8 ms
$2^{19}/f_{XX}$	26.2 ms	41.6 ms
$2^{20}/f_{XX}$	52.4 ms	83.3 ms
$2^{22}/f_{XX}$	209.7 ms	333.4 ms

Note Only in the V850/SB1.

9.2 Configuration

The watchdog timer includes the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Oscillation stabilization time selection register (OSTS) Watchdog timer clock selection register (WDCS) Watchdog timer mode register (WDTM)

9.3 Watchdog Timer Control Register

The registers to control the watchdog timer are shown below.

- Oscillation stabilization time selection register (OSTS)
- Watchdog timer clock selection register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Oscillation stabilization time selection register (OSTS)

This register selects the oscillation stabilization time after a reset is applied or the 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[Note]	12.58 MHz
0	0	0	2^{14}/fxx	819.2 μs	1.3 ms
0	0	1	2^{16}/fxx	3.3 ms	5.2 ms
0	1	0	2^{17}/fxx	6.6 ms	10.4 ms
0	1	1	2^{18}/fxx	13.1 ms	20.8 ms
1	0	0	2^{19}/fxx (after reset)	26.2 ms	41.6 ms
Other than above			Setting prohibited		

Note Only in the V850/SB1.

(2) Watchdog timer clock selection 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 clears 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[Note]	12.58 MHz
0	0	0	2^{14}/fxx	819.2 μs	1.3 ms
0	0	1	2^{15}/fxx	1.6 ms	2.6 ms
0	1	0	2^{16}/fxx	3.3 ms	5.2 ms
0	1	1	2^{17}/fxx	6.6 ms	10.4 ms
1	0	0	2^{18}/fxx	13.1 ms	20.8 ms
1	0	1	2^{19}/fxx	26.2 ms	41.6 ms
1	1	0	2^{20}/fxx	52.4 ms	83.3 ms
1	1	1	2^{22}/fxx	209.7 ms	333.4 ms

Note Only in the V850/SB1.

★ **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 an 8-bit or 1-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears 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 the watchdog timer[Note 1]
0	Disable count
1	Clear count and start counting

WDTM4	Operating mode selection for the 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 may be up to $2^{10}/f_{XX}$ seconds shorter than the set time.

9.4 Operation

9.4.1 Operation as watchdog timer

Set bit 4 (WDTM4) of the watchdog timer mode register (WDTM) to 1 to operate as a watchdog timer to detect an inadvertent 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 inadvertent program loop detection, the watchdog timer is cleared and counting starts again.

If RUN is not set to 1 and the inadvertent program loop detection time has elapsed, a non-maskable interrupt (INTWDT) is generated (no reset functions).

The watchdog timer stops running in the IDLE mode and STOP mode. Consequently, set RUN to 1 and clear the watchdog timer before entering the IDLE mode or STOP mode. Do not set the watchdog timer when operating the HALT mode since the watchdog timer running in HALT mode.

Cautions 1. The actual inadvertent program loop detection time may be up to $2^{10}/f_{XX}$ seconds less than the set time.

2. When the subclock is selected for the CPU clock, the watchdog timer stops (retains) counting.

Table 9-4. Inadvertent Program Loop Detection Time of Watchdog Timer

Clock	Inadvertent Program Loop Detection Time	
	f_{XX} = 20 MHz[Note]	f_{XX} = 12.58 MHz
$2^{14}/f_{XX}$	819.2 μs	1.3 ms
$2^{15}/f_{XX}$	1.6 ms	2.6 ms
$2^{16}/f_{XX}$	3.3 ms	5.2 ms
$2^{17}/f_{XX}$	6.6 ms	10.4 ms
$2^{18}/f_{XX}$	13.1 ms	20.8 ms
$2^{19}/f_{XX}$	26.2 ms	41.6 ms
$2^{20}/f_{XX}$	52.4 ms	83.3 ms
$2^{22}/f_{XX}$	209.7 ms	333.4 ms

Note Only in the V850/SB1.

9.4.2 Operation 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 STOP mode. Therefore, before entering the IDLE mode/STOP mode, set the RUN bit of the WDTM register to 1 and clear the interval timer. Then set the IDLE mode/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 being set by WDTM may be up to $2^{10}/f_{XX}$ seconds less than the set time.

3. When the subclock is selected for the CPU clock, the watchdog timer stops (retains) counting.

Table 9-5. Interval Time of Interval Timer

Clock	Interval Time	
	f_{XX} = 20 MHz[Note]	f_{XX} = 12.58 MHz
$2^{14}/f_{XX}$	819.2 μs	1.3 ms
$2^{15}/f_{XX}$	1.6 ms	2.6 ms
$2^{16}/f_{XX}$	3.3 ms	5.2 ms
$2^{17}/f_{XX}$	6.6 ms	10.4 ms
$2^{18}/f_{XX}$	13.1 ms	20.8 ms
$2^{19}/f_{XX}$	26.2 ms	41.6 ms
$2^{20}/f_{XX}$	52.4 ms	83.3 ms
$2^{22}/f_{XX}$	209.7 ms	333.4 ms

Note Only in the V850/SB1.

9.5 Standby Function Control Register

(1) Oscillation stabilization time selection register (OSTS)

The wait time from releasing the stop mode until the oscillation stabilizes is controlled by the oscillation stabilization time selection register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H.

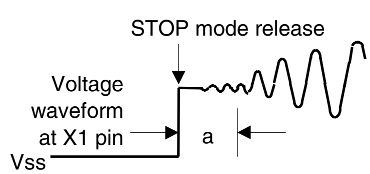
After reset: 04H R/W Address: FFFFF380H

	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[Note]	12.58 MHz
0	0	0	2^{14}/fxx	819.2 μs	1.3 ms
0	0	1	2^{16}/fxx	3.3 ms	5.2 ms
0	1	0	2^{17}/fxx	6.6 ms	10.4 ms
0	1	1	2^{18}/fxx	13.1 ms	20.8 ms
1	0	0	2^{19}/fxx (after reset)	26.2 ms	41.6 ms
Other than above			Setting prohibited		

Note Only in the V850/SB1.

Caution The wait time at the release of the STOP mode does not include the time ("a" in the figure below) until clock oscillation starts after releasing the STOP mode when $\overline{\text{RESET}}$ is input or an interrupt is generated.



CHAPTER 10 SERIAL INTERFACE FUNCTION

10.1 Overview

The V850/SB1 and V850/SB2 incorporate the following serial interfaces.

- Channel 0: 3-wire serial I/O (CSI0)/I^{2}C0[Note]
- Channel 1: 3-wire serial I/O (CSI1)/Asynchronous serial interface (UART0)
- Channel 2: 3-wire serial I/O (CSI2)/I^{2}C1[Note]
- Channel 3: 3-wire serial I/O (CSI3)/Asynchronous serial interface (UART1)
- Channel 4: 8 to 16-bit variable-length 3-wire serial I/O (CSI4)

Note I^{2}C0 and I^{2}C1 support multimasters (Y versions (products with on-chip I^{2}C) only).
Either 3-wire serial I/O or I^{2}C can be used as a serial interface.

10.2 3-Wire Serial I/O (CSI0 to CSI3)

CSIn (n = 0 to 3) 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 ($\overline{\text{SCKn}}$), 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 $\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$ 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.

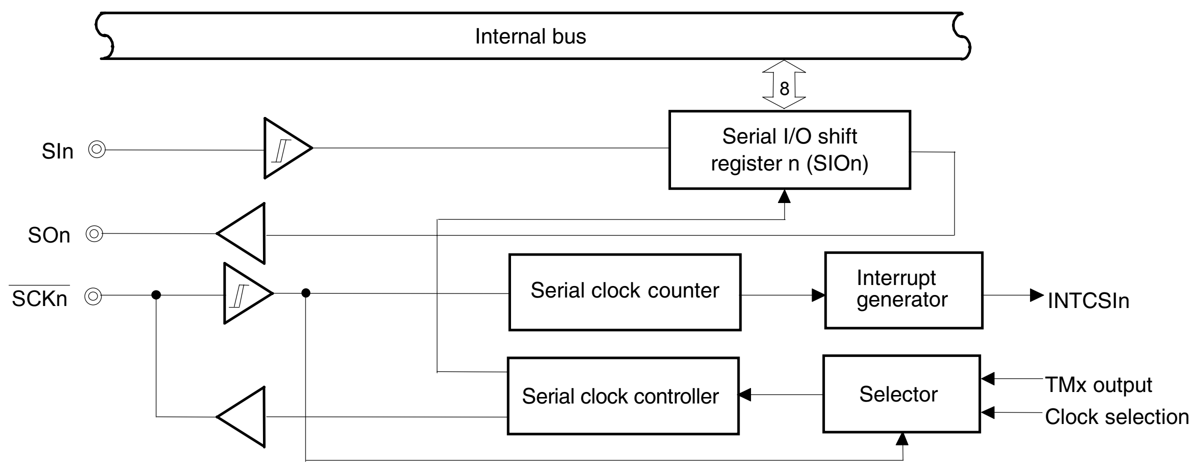
10.2.1 Configuration

CSIn includes the following hardware.

Table 10-1. Configuration of CSIn

Item	Configuration
Registers	Serial I/O shift registers 0 to 3 (SIO0 to SIO3)
Control registers	Serial operation mode registers 0 to 3 (CSIM0 to CSIM3)
	Serial clock selection registers 0 to 3 (CSIS0 to CSIS3)

Figure 10-1. Block Diagram of 3-Wire Serial I/O



Remarks 1. n = 0 to 3

2. TMx output is as follows:
When n = 0 or 3: TM2
When n = 1 or 2: TM3

(1) Serial I/O shift registers 0 to 3 (SIO0 to SIO3)

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 clears 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 MODE = 0 and write is disabled when MODE = 1).

10.2.2 CSIn control registers

CSIn is controlled by the following registers.

- Serial operation mode register n (CSIMn)
- Serial clock selection register n (CSISn)

(1) Serial clock selection registers 0 to 3 (CSIS0 to CSIS3) and serial operation mode registers 0 to 3 (CSIM0 to CSIM3)

The CSISn register is used to set serial interface channel n's serial clock.
The CSISn register can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears the CSISn register 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 an 8-bit or 1-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears the CSIMn register to 00H.

After reset : 00H R/W Address: CSIS0 FFFFF2A4H CSIS2 FFFFF2C4H
CSIS1 FFFFF2B4H CSIS3 FFFFF2D4H

	7	6	5	4	3	2	1	0
CSISn	0	0	0	0	0	0	0	SCLn2

(n = 0 to 3)

After reset: 00H R/W Address: CSIM0 FFFFF2A2H CSIM2 FFFFF2C2H
CSIM1 FFFFF2B2H CSIM3 FFFFF2D2H

	<7>	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

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 2]

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	at n = 0, 3: Output of TO2 at n = 1, 2: Output of TO3
0	1	0	fxx/8
0	1	1	fxx/16
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	fxx/32
1	1	1	fxx/64

Notes 1. The SIn, SOn, and $\overline{\text{SCKn}}$ pins are used as port function pins when CSIEn = 0 (SIOn operation stop status).

2. When CSIEn = 1 (SIOn operation enable status), the port function is available for the SIn pin when only using the transmit function and SOn pin when only using the receive function.

Cautions 1. Do not perform bit manipulation of SCLn1 and SCLn0.

2. Be sure to set bits 6 to 3 of the CSIMn register to 0.

Remark When the output of the timer is selected as the clock, it is not necessary to set the P26/TO2/TI2 and P27/TO3/TI3 pins in the 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 SIn, SOn, and $\overline{\text{SCKn}}$ pin 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. CSIMn Setting (Operation Stop Mode)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

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 ($\overline{SCKn}$), 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. CSIMn Setting (3-Wire Serial I/O Mode)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn (n = 0 to 3)	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

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	Write to SIOn	Normal output
1	Receive-only mode	Read from SIOn	Port function

SCLn2	SCLn1	SCLn0	Clock selection
0	0	0	External clock input ($\overline{SCKn}$)
0	0	1	When n = 0, 3: TO2 When n = 1, 2: TO3
0	1	0	fxx/8
0	1	1	fxx/16
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	fxx/32
1	1	1	fxx/64

Remark Refer to **10.2.2 (1) Serial clock selection registers 0 to 3 (CSIS0 to CSIS3) and serial operation mode registers 0 to 3 (CSIM0 to CSIM3)** for the SCLn2 bit.

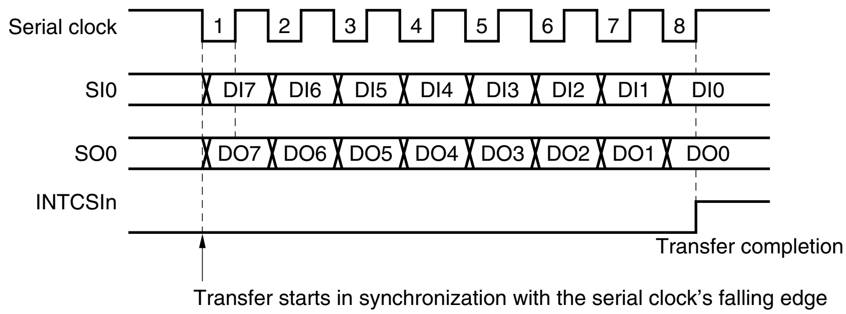
(b) Communication operations

In 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is sent 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.

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 I^2C Bus (A Versions)

To use the I^2C bus function, set the P10/SDA0, P12/SCL0, P20/SDA1, and P22/SCL1 pins to N-ch open drain output.

The products with an on-chip I^2C bus are shown below.

- V850/SB1: μPD703031AY, 703032AY, 703033AY, 70F3032AY, 70F3033AY
- V850/SB2: μPD703034AY, 703035AY, 703037AY, 70F3035AY, 70F3037AY

The I^2C0 and I^2C1 have the following two modes.

- Operation stop mode
- I^2C (Inter IC) bus mode (multimasters 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) I^2C bus mode (multimaster support)

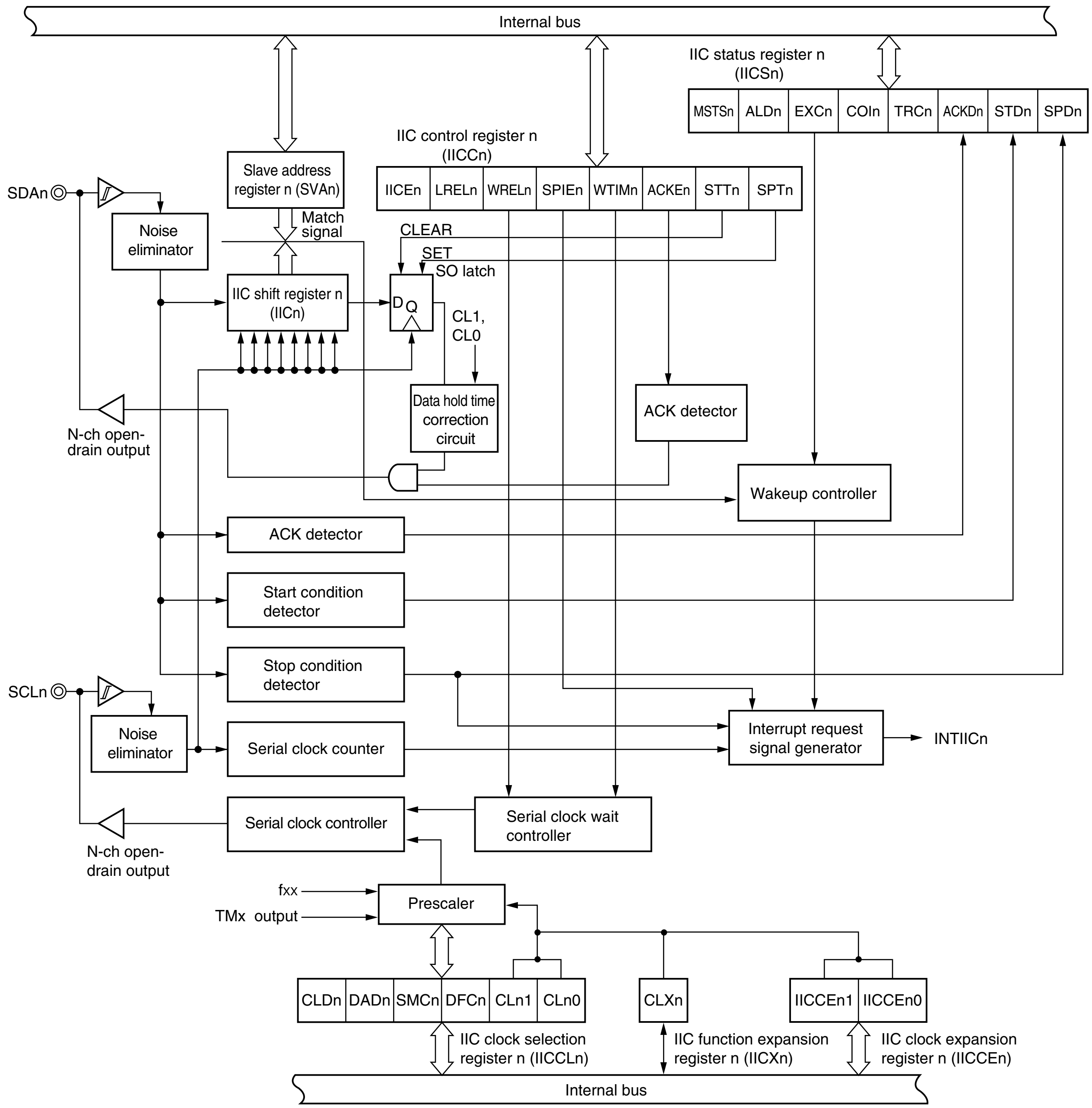
This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLn) line and a serial data bus (SDAn) line.

This mode complies with the I^2C 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 I^2C bus.

Since SCLn and SDAn are open drain outputs, the I^2Cn requires pull-up resistors for the serial clock line and the serial data bus line.

Remark n = 0, 1

Figure 10-5. Block Diagram of I²C



Remarks 1. n = 0, 1

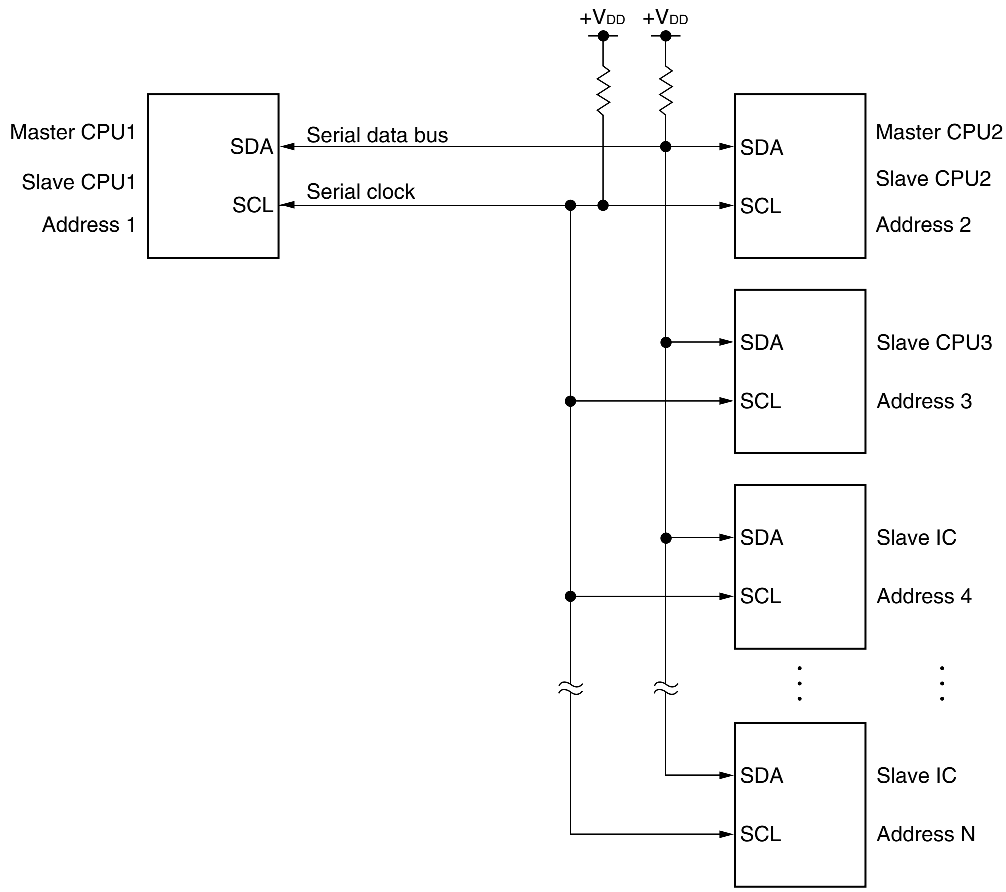
2. TMx output

n = 0: TM2 output

n = 1: TM3 output

A serial bus configuration example is shown below.

Figure 10-6. Serial Bus Configuration Example Using I²C Bus



10.3.1 Configuration

I^2Cn includes the following hardware (n = 0, 1).

Table 10-2. Configuration of I^2Cn

Item	Configuration
Registers	IIC shift registers 0 and 1 (IIC0, IIC1) Slave address registers 0 and 1 (SVA0, SVA1)
Control registers	IIC control registers 0 and 1 (IICC0, IICC1) IIC status registers 0 and 1 (IICS0, IICS1) IIC clock selection registers 0 and 1 (IICCL0, IICCL1) IIC clock expansion registers 0 and 1 (IICCE0, IICCE1) IICC function expansion registers 0 and 1 (IICX0, IICX1)

(1) IIC shift registers 0 and 1 (IIC0, IIC1)

IICn is used to convert 8-bit serial data into 8-bit parallel data and vice versa. IICn can be used for both transmission and reception (n = 0, 1).

Write and read operations to IICn are used to control the actual transmit and receive operations.

IICn is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IIC0 and IIC1 to 00H.

(2) Slave address registers 0 and 1 (SVA0, SVA1)

SVAn sets local addresses when in slave mode.

SVAn is set by an 8-bit memory manipulation instruction (n = 0, 1).

$\overline{\text{RESET}}$ input clears SVA0 and SVA1 to 00H.

(3) SO latch

The SO latch is used to retain the SDAn pin's output level (n = 0, 1).

(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 n (SVAn) or when an extension code is received (n = 0, 1).

(5) Clock selector

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICn).

An I^2C interrupt is generated following either of two triggers.

- Eighth or ninth clock of the serial clock (set by WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by SPIEn bit)

Remarks 1. n = 0, 1

2. WTIMn bit: Bit 3 of IIC control register n (IICCn)

SPIEn bit: Bit 4 of IIC control register n (IICCn)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCLn pin from a sampling clock (n = 0, 1).

(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 I²C control registers

I²C0 and I²C1 are controlled by the following registers.

- IIC control registers 0, 1 (IICC0, IICC1)
- IIC status registers 0, 1 (IICS0, IICS1)
- IIC clock selection registers 0, 1 (IICCL0, IICCL1)
- IIC clock expansion registers 0, 1 (IICCE0, IICCE1)
- IIC function expansion registers 0, 1 (IICX0, IICX1)

The following registers are also used.

- IIC shift registers 0, 1 (IIC0, IIC1)
- Slave address registers 0, 1 (SVA0, SVA1)

(1) IIC control registers 0, 1 (IICC0, IICC1)

IICCn is used to enable/disable I²C operations, set wait timing, and set other I²C operations.
IICCn can be set by an 8-bit or 1-bit memory manipulation instruction (n = 0, 1).
$\overline{\text{RESET}}$ input clears IICCn to 00H.

★ **Caution In I²C0, I²C1 bus mode, set the port 1 mode register (PM1), port 2 mode register (PM2), port 1 function register (PF1), and port 2 function register (PF2) as follows. In addition, set each output latch to 0.**

Pin	Port Mode Register	Port Function Register
P10/SI0/SDA0	PM10 of PM1 register = 0	PF10 of PF1 register = 1
P12/$\overline{\text{SCK0}}$/SCL0	PM12 of PM1 register = 0	PF12 of PF1 register = 1
P20/SI2/SDA1	PM20 of PM2 register = 0	PF20 of PF2 register = 1
P22/$\overline{\text{SCK2}}$/SCL1	PM22 of PM2 register = 0	PF22 of PF2 register = 1

(1/4)

After reset: 00H R/W Address: FFFFF340H, FFFFF350H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

(n = 0, 1)

IICEn	I^{2}Cn operation enable/disable specification
0	Operation stopped. IIC status register n (IICSn) preset. Internal operation stopped.
1	Enables operation.
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

LRELn	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 SCLn and SDAn lines are set to high impedance. The following flags are cleared. • STDn • ACKDn • TRCn • COIn • EXCn • MSTSn • STTn • SPTn
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 (LRELn = 0)[Note]	Condition for setting (LRELn = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark STDn: Bit 1 of IIC state register n (IICSn)
ACKDn: Bit 2 of IIC state register n (IICSn)
TRCn: Bit 3 of IIC state register n (IICSn)
COIn: Bit 4 of IIC state register n (IICSn)
EXCn: Bit 5 of IIC state register n (IICSn)
MSTSn: Bit 7 of IIC state register n (IICSn)

(2/4)

WRELn	Wait cancellation control
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.

Condition for clearing (WRELn = 0)[Note]	Condition for setting (WRELn = 1)
• Automatically cleared after execution • When $\overline{RESET}$ is input	• Set by instruction

SPIEn	Enable/disable generation of interrupt request when stop condition is detected
0	Disabled
1	Enabled

Condition for clearing (SPIEn = 0)[Note]	Condition for setting (SPIEn = 1)
• Cleared by instruction • When $\overline{RESET}$ is input	• Set by instruction

WTIMn	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 $\overline{ACK}$ 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 (WTIMn = 0)[Note]	Condition for setting (WTIMn = 1)
• Cleared by instruction • When $\overline{RESET}$ is input	• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark n = 0, 1

(3/4)

ACKEn	Acknowledge control
0	Disable acknowledgment.
1	Enable acknowledgment. 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 EXCn = 1.
Condition for clearing (ACKEn = 0)[Note]	Condition for setting (ACKEn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

STTn	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in STOP mode): Generate a start condition (for starting as master). The SDAn 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, SCLn 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 ACKEn has been set to 0 and slave has been notified of final reception. • For master transmission: A start condition cannot be generated normally during the ACKn period. Set during the wait period. • Cannot be set at the same time as SPTn	
Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by instruction • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When LRELn = 1 • When IICEn = 0 • Cleared when $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark Bit 1 (STTn) is 0 if it is read immediately after data setting.

(4/4)

<table>
<tr><th>SPTn</th><th colspan="2">Stop condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device's transfer).
After the SDAn line goes to low level, either set the SCLn line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDAn line is changed from low level to high level and a stop condition is generated.</td></tr>
<tr><td colspan="3">Cautions concerning setting timing
• For master reception: Cannot be set during transfer.
Can be set only when ACKEn has been set to 0 and during the wait period after slave has been notified of final reception.
• For master transmission: A stop condition cannot be generated normally during the ACKn period. Set during the wait period.
• Cannot be set at the same time as STTn.
• SPTn can be set only when in master mode[Note]
• When WTIMn has been set to 0, if SPTn 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, WTIMn should be changed from 0 to 1 during the wait period following output of eight clocks, and SPTn should be set during the wait period that follows output of the ninth clock.</td></tr>
<tr><td colspan="2">Condition for clearing (SPTn = 0)</td><td>Condition for setting (SPTn = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• When LRELn = 1
• When IICEn = 0
• Cleared when $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

Note Set SPTn only in master mode. However, SPTn 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 (TRCn) of IIC status register n (IICSn) is set to 1, WRELn is set during the ninth clock and wait is canceled, after which TRCn is cleared and the SDAn line is set to high impedance.

Remarks 1. Bit 0 (SPTn) is 0 if it is read immediately after data setting.

2. n = 0, 1

(2) IIC status registers 0, 1 (IICS0, IICS1)

IICSn indicates the status of the I^2Cn bus.

IICSn can be set by an 8-bit or 1-bit memory manipulation instruction. IICSn is a read-only register (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICSn to 00H.

(1/3)

After reset: 00H R Address: FFFFF342H, FFFFF352H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

(n = 0, 1)

MSTSn	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is generated

ALDn	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”. MSTSn is cleared.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after IICSn is read[Note] • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ 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 IICSn.

Remark LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

(2/3)

EXCn	Detection of extension code reception	
0	Extension code was not received.	
1	Extension code was received.	
Condition for clearing (EXCn = 0)		Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 • When IICEn 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).

COIn	Detection of matching addresses	
0	Addresses do not match.	
1	Addresses match.	
Condition for clearing (COIn = 0)		Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input		• When the received address matches the local address (SVAn) (set at the rising edge of the eighth clock).

TRCn	Detection of transmit/receive status	
0	Receive status (other than transmit status). The SDAn line is set to high impedance.	
1	Transmit status. The value in the SO latch is enabled for output to the SDAn line (valid starting at the falling edge of the first byte’s ninth clock).	
Condition for clearing (TRCn = 0)		Condition for setting (TRCn = 1)
• When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • Cleared by WRELn = 1[Note] • When $\overline{\text{ALDn}}$ 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 TRCn is cleared and SDAn line become high impedance when bit 5 (WRELn) of IIC control register n (IICCn) is set and the wait state is released at the ninth clock with bit 3 (TRCn) of IIC status register n (IICSn) = 1.

Remarks 1. WRELn: Bit 5 of IIC control register n (IICCn)
LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

2. n = 0, 1

(3/3)

ACKDn	Detection of $\overline{\text{ACK}}$
0	$\overline{\text{ACK}}$ was not detected.
1	$\overline{\text{ACK}}$ was detected.
Condition for clearing (ACKDn = 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 LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• After the SDAn line is set to low level at the rising edge of the SCLn’s ninth clock

STDn	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 (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the next byte’s first clock following address transfer • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	When a start condition is detected

SPDn	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 (SPDn = 0)	Condition for setting (SPDn = 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 IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	When a stop condition is detected

Remarks 1. LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

2. n = 0, 1

(3) IIC clock selection registers 0, 1 (IICCL0, IICCL1)

IICCLn is used to set the transfer clock for the I^2Cn bus.

IICCLn can be set by an 8-bit or 1-bit memory manipulation instruction. Bits SMCn, CLn1 and CLn0 are set using the CLXn bit of IIC function expansion register n (IICXn) in combination with bits IICCEn1 and IICCEn0 of IIC clock expansion register n (IICCEn) (n = 0, 1) (see **10.3.2 (6) I^2Cn transfer clock setting method**).

$\overline{\text{RESET}}$ input clears IICCLn to 00H.

After reset: 00H   R/W[Note]   Address: FFFFF344H, FFFFF354H

	7	6	<5>	<4>	3	2	1	0
IICCLn	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0

(n = 0, 1)

CLDn	Detection of SCLn line level (valid only when IICEn = 1)
0	SCLn line was detected at low level.
1	SCLn line was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLn line is at low level • When IICEn = 0 • When $\overline{\text{RESET}}$ is input	• When the SCLn line is at high level

DADn	Detection of SDAn line level (valid only when IICEn = 1)
0	SDAn line was detected at low level.
1	SDAn line was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAn line is at low level • When IICEn = 0 • When $\overline{\text{RESET}}$ is input	• When the SDAn line is at high level

SMCn	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
A digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFCn switching (on/off).	

Note Bits 4 and 5 are read only bits.

★ **Caution Be sure to set bits 7 and 6 to 0.**

Remark IICEn: Bit 7 of IIC control register n (IICCn)

(4) IIC function expansion registers 0, 1 (IICX0, IICX1)

These registers set the function expansion of I^2Cn (valid only in high-speed mode).

IICXn is set with a 1-bit or 8-bit memory manipulation instruction. Set the CLXn bit in combination with the SMCn, DFCn, CLn1, and the CLn0 bits of IIC clock selection register n (IICCLn) and the IICCEn1 and IICCEn0 bits of IIC clock expansion register n (IICCEn) (see **10.3.2 (6) I^2Cn transfer clock setting method**) (n = 0, 1).

$\overline{\text{RESET}}$ input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF34AH, FFFFF35AH

	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(n = 0, 1)

(5) IIC clock expansion registers 0, 1 (IICCE0, IICCE1)

These registers set the transfer clock expansion of I^2Cn.

IICCEn is set with an 8-bit memory manipulation instruction. Set the IICCEn1 and IICCEn0 bits in combination with the SMCn, CLn1, and CLn0 bits of IIC clock selection register n (IICCLn) and the CLXn bit of IIC function expansion register n (IICXn) (see **10.3.2 (6) I^2Cn transfer clock setting method**) (n = 0, 1).

$\overline{\text{RESET}}$ input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF34CH, FFFFF35CH

	7	6	5	4	3	2	1	0
IICCEn	0	0	0	0	0	0	IICCEn1	IICCEn0

(n = 0, 1)

(6) I^2Cn transfer clock setting method

The I^2Cn transfer clock frequency (f_{SCL}) is calculated using the following expression (n = 0, 1).

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 24, 48, 36, 54, 44, 86, 172, 132, 198 (see **Table 10-3 Selection Clock Setting**.)

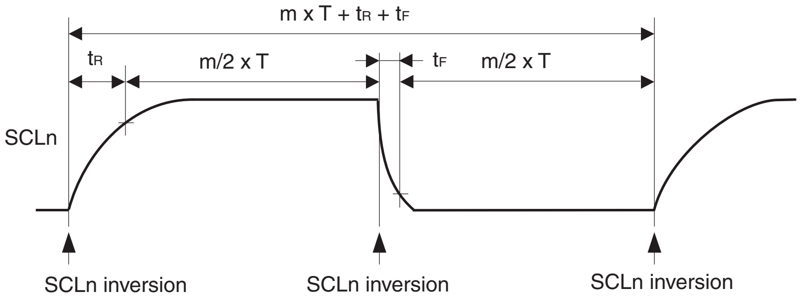
T: $1/f_{XX}$

t_R: SCLn rise time

t_F: SCLn fall time

For example, the I^2Cn transfer clock frequency (f_{SCL}) when f_{XX} = 20 MHz, m = 198, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(198 \times 50 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 98.5 \text{ kHz}$$



The selection clock is set using a combination of the SMCn, CLn1, and CLn0 bits of IIC clock selection register n (IICCLn), the CLXn bit of IIC function expansion register n (IICXn), and IICCEn1 and the IICCEn0 bits of IIC clock expansion register n (IICCEn) (n = 0, 1).

Table 10-3. Selection Clock Setting

IICCEn		IICXn	IICCLn			Selection Clock (f_{XX}/m)		Settable Main Clock Frequency (f_{XX}) Range	Operation Mode
Bit 1	Bit 0	Bit 0	Bit 3	Bit 1	Bit 0				
IICCEn1	IICCEn0	CLXn	SMCn	CLn1	CLn0				
x	x	1	1	0	x	f_{XX}/12		4.0 MHz to 4.19 MHz	High-speed mode (SMCn = 1)
x	x	0	1	0	x	f_{XX}/24		4.0 MHz to 8.38 MHz	
x	x	0	1	1	0	f_{XX}/48		8.0 MHz to 16.67 MHz	
0	1	0	1	1	1	f_{XX}/36		12.0 MHz to 13.4 MHz	
1	0	0	1	1	1	f_{XX}/54		16.0 MHz to 20.0 MHz[Note]	
0	0	0	1	1	1	n = 0	TM2 output/18	TM2 setting	
						n = 1	TM3 output/18	TM3 setting	
x	x	0	0	0	0	f_{XX}/44		2.0 MHz to 4.19 MHz	Normal mode (SMCn = 0)
x	x	0	0	0	1	f_{XX}/86		4.19 MHz to 8.38 MHz	
x	x	0	0	1	0	f_{XX}/172		8.38 MHz to 16.67 MHz	
0	1	0	0	1	1	f_{XX}/132		12.0 MHz to 13.4 MHz	
1	0	0	0	1	1	f_{XX}/198		16.0 MHz to 20.0 MHz[Note]	
0	0	0	0	1	1	n = 0	TM2 output/66	TM2 setting	
						n = 1	TM3 output/66	TM3 setting	
Other than above						Setting prohibited			

Note Only in the V850/SB1 and the H versions of the V850/SB2.

Remarks 1. n = 0, 1

2. x: don't care

3. When the output of the timer is selected as the clock, it is not necessary to set the P26/TO2/TI2 and P27/TO3/TI3 pins in the timer output mode.

(7) IIC shift registers 0, 1 (IIC0, IIC1)

IICn is used for serial transmission/reception (shift operations) that is synchronized with the serial clock. It can be read from or written to in 8-bit units, but data should not be written to IICn during a data transfer (n = 0, 1).

After reset: 00H R/W Address: FFFFF348H, FFFFF358H

	7	6	5	4	3	2	1	<0>
IICn								

(n = 0, 1)

(8) Slave address registers 0, 1 (SVA0, SVA1)

SVAn holds the I^{2}C 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, FFFFF356H

	7	6	5	4	3	2	1	0
SVAn								0

(n = 0, 1)

10.3.3 I^{2}C bus mode functions

(1) Pin configuration

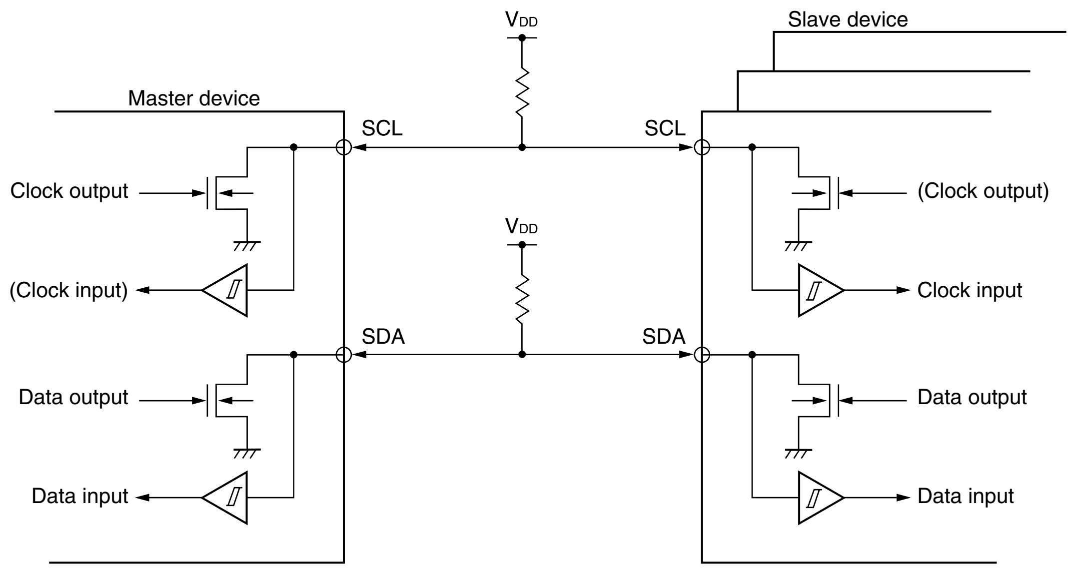
The serial clock pin (SCLn) and serial data bus pin (SDAn) are configured as follows (n = 0, 1).

SCLn .... 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.

SDAn .... 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.

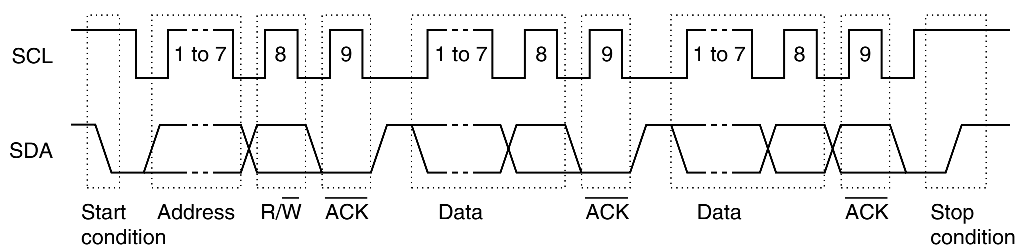
Figure 10-7. Pin Configuration Diagram



10.3.4 I²C bus definitions and control methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. The transfer timing for the "start condition", "data", and "stop condition" output via the I²C bus's serial data bus is shown below.

Figure 10-8. I²C Bus's Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

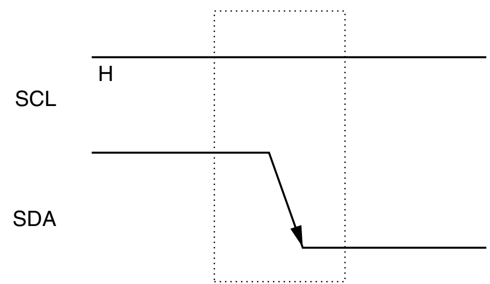
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLn) is continuously output by the master device. However, in the slave device, the SCLn's low-level period can be extended and a wait can be inserted (n = 0, 1).

(1) Start condition

The start condition is met when the SCLn pin is at high level and the SDAn pin changes from high level to low level. The start conditions for the SCLn pin and SDAn 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 (n = 0, 1).

Figure 10-9. Start Condition



A start condition is output when bit 1 (STTn) of IIC control register n (IICCn) is set to 1 after a stop condition has been detected (SPDn: Bit 0 = 1 in the IIC status register n (IICSn)). When a start condition is detected, bit 1 (STDn) of IICSn is set to 1 (n = 0, 1).

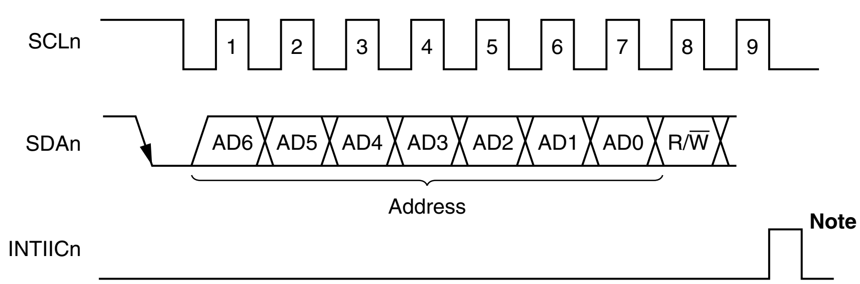
(2) Address

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 n (SVAn). If the address data matches the SVAn values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition (n = 0, 1).

Figure 10-10. Address



Note INTIICn is generated if a local address or extension code is received during slave device operation.

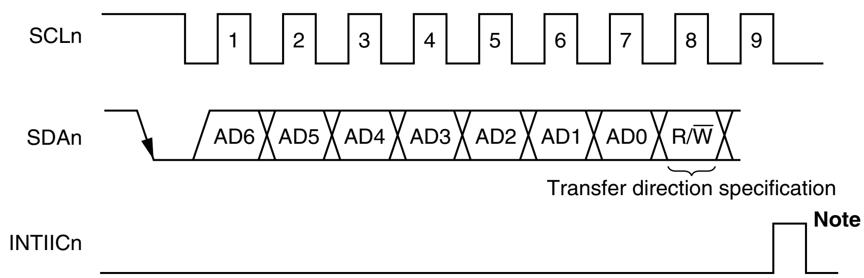
Remark n = 0, 1

The slave address and the eighth bit, which specifies the transfer direction as described in **(3) Transfer direction specification** below, are written together to the IIC shift register (IICn) and are then output. Received addresses are written to IICn (n = 0, 1).
The slave address is assigned to the higher 7 bits of IICn.

(3) Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 10-11. Transfer Direction Specification



Note INTIICn is generated if a local address or extension code is received during slave device operation.

Remark n = 0, 1

(4) Acknowledge signal ($\overline{ACK}$)

The acknowledge signal ($\overline{ACK}$) is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{ACK}$ signal for every 8 bits of data it receives. The transmitting device normally receives an $\overline{ACK}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{ACK}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{ACK}$ signal is returned after it transmits 8 bits of data. When an $\overline{ACK}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{ACK}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{ACK}$ 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 SDAn line to low level during the ninth clock, the $\overline{ACK}$ signal becomes active (normal receive response).

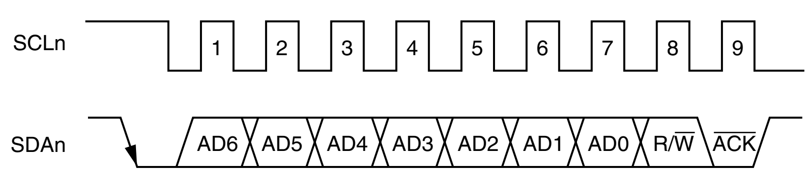
When bit 2 (ACKEn) of IIC control register n (IICCn) is set to 1, automatic $\overline{ACK}$ signal generation is enabled (n = 0, 1).

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRCn) of IIC status register n (IICSn) to be set. When this TRCn bit's value is 0, it indicates receive mode. Therefore, ACKEn should be set to 1 (n = 0, 1).

When the slave device is receiving (when TRCn = 0), if the slave device does not need to receive any more data after receiving several bytes, setting ACKEn to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRCn = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting ACKEn to 0 will prevent the $\overline{ACK}$ signal from being returned. This prevents the MSB data from being output via the SDAn line (i.e., stops transmission) during transmission from the slave device.

Figure 10-12. $\overline{ACK}$ Signal



Remark n = 0, 1

When the local address is received, an $\overline{ACK}$ signal is automatically output in synchronization with the falling edge of the SCLn's eighth clock regardless of the ACKEn value. No $\overline{ACK}$ signal is output if the received address is not a local address (n = 0, 1).

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 SCLn's eighth clock if ACKEn is set to 1 before wait cancellation.

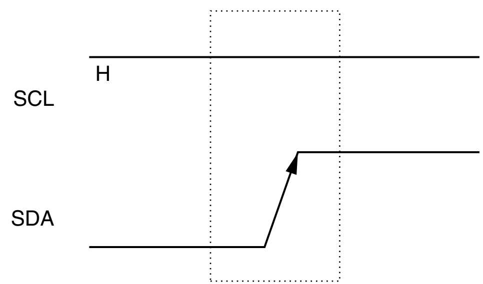
When 9-clock wait is selected: $\overline{ACK}$ signal is automatically output at the falling edge of the SCLn's eighth clock if ACKEn has already been set to 1.

(5) Stop condition

When the SCLn pin is at high level, changing the SDAn pin from low level to high level generates a stop condition (n = 0, 1).

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



Remark n = 0, 1

A stop condition is generated when bit 0 (SPTn) of IIC control register n (IICCn) is set to 1. When the stop condition is detected, bit 0 (SPDn) of IIC status register n (IICSn) is set to 1 and INTIICn is generated when bit 4 (SPIEn) of IICCn is set to 1 (n = 0, 1).

(6) Wait signal ($\overline{\text{WAIT}}$)

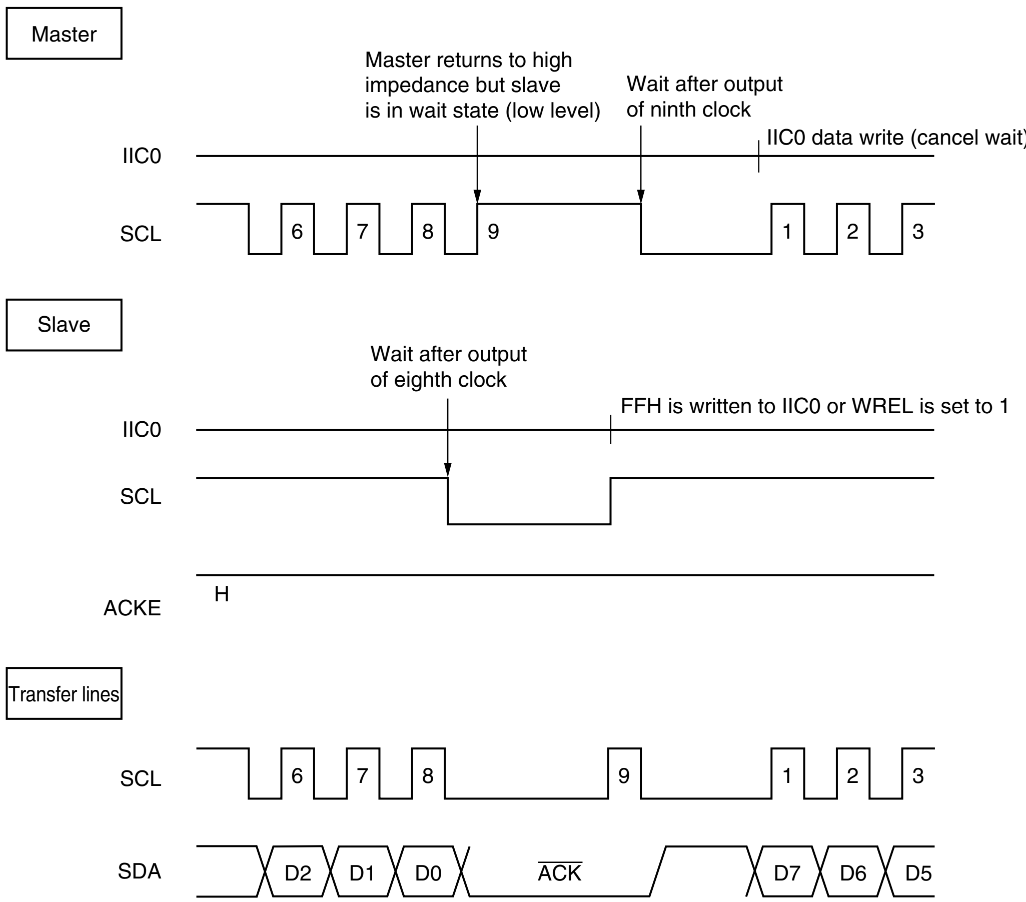
The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCLn 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 (n = 0, 1).

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 ACKEn = 1)

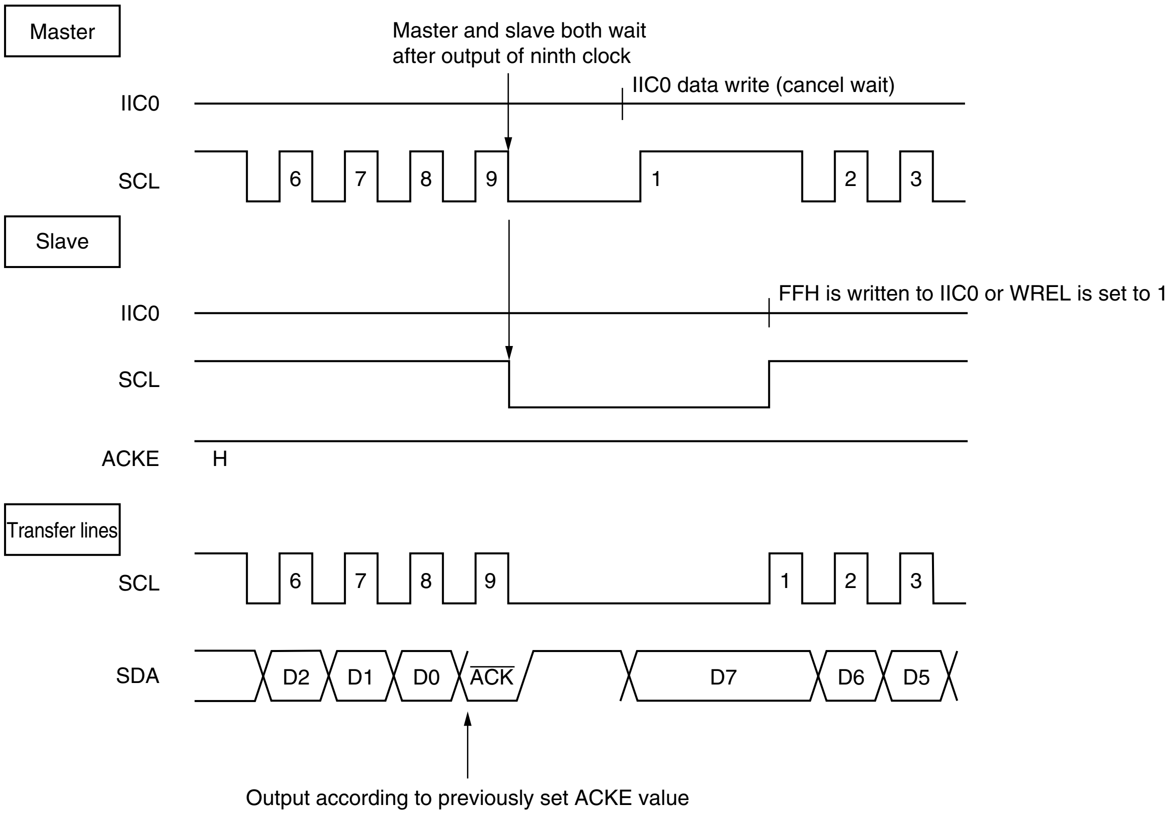


Remark n = 0, 1

Figure 10-14. Wait Signal (2/2)

(b) When master and slave devices both have a nine-clock wait

(master: transmission, slave: reception, and ACKEn = 1)



Remarks 1. ACKEn: Bit 2 of IIC control register n (IICCn)
WRELn: Bit 5 of IIC control register n (IICCn)

2. n = 0, 1

A wait may be automatically generated depending on the setting for bit 3 (WTIMn) of IIC control register n (IICCn) (n = 0, 1).

Normally, when bit 5 (WRELn) of IICCn is set to 1 or when FFH is written to IIC shift register n (IICn), the wait status is canceled and the transmitting side writes data to IICn to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STTn) of IICCn to 1
- By setting bit 0 (SPTn) of IICCn to 1

10.3.5 I²C interrupt requests (INTIICn)

The following shows the value of IIC status register n (IICSn) at the INTIICn interrupt request generation timing and at the INTIICn interrupt timing (n = 0, 1).

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When WTIMn = 0

SPTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX000B

▲3: IICSn = 10XXX000B (WTIMn = 0)

▲4: IICSn = 10XXXX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

SPTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX100B

▲3: IICSn = 10XXXX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIMn = 0

STTn = 1 ↓ (before the second ST); SPTn = 1 ↓ (before SP)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2	▲3				▲4	▲5	▲6	Δ7

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX000B (WTIMn = 1)

▲3: IICSn = 10XXXX00B (WTIMn = 0)

▲4: IICSn = 10XXX110B (WTIMn = 0)

▲5: IICSn = 10XXX000B (WTIMn = 1)

▲6: IICSn = 10XXXX00B

Δ 7: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

STTn = 1 ↓ (before the second ST); SPTn = 1 ↓ (before SP)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3		▲4	Δ5

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXXX00B

▲3: IICSn = 10XXX110B

▲4: IICSn = 10XXXX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIMn = 0

SPTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 1010X110B

▲2: IICSn = 1010X000B

▲3: IICSn = 1010X000B (WTIMn = 1)

▲4: IICSn = 1010XX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

SPTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 1010X110B

▲2: IICSn = 1010X100B

▲3: IICSn = 1010XX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(2) Slave device operation (when receiving slave address data (match with SVAn))

(a) Start ~ Address ~ Data ~ Data ~ Stop

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 0001X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X100B

▲3: IICSn = 0001XX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
				▲1	▲2					▲3	▲4		△5

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 0001X110B

▲4: IICSn = 0001X000B

△ 5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
				▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0001X110B

▲4: IICSn = 0001XX00B

△ 5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2				▲3		▲4		Δ5

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X000B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

<2> When WTIMn = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2			▲3	▲4		▲5	Δ6

▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X110B

▲5: IICSn = 0010XX00B

Δ 6: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 00000X10B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 00000X10B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(3) Slave device operation (when receiving extension code)

(a) Start ~ Code ~ Data ~ Data ~ Stop

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	△4

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 0010X000B

△ 4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		▲4 △5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010X100B

▲4: IICSn = 0010XX00B

△ 5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1		▲2					▲3	▲4		△5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 0001X110B

▲4: IICSn = 0001X000B

△ 5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3				▲4		▲5	△6

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0001X110B

▲5: IICSn = 0001XX00B

△ 6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1		▲2				▲3		▲4		Δ5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X000B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

<2> When WTIMn = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3			▲4	▲5		▲6	Δ7

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0010X010B

▲5: IICSn = 0010X110B

▲6: IICSn = 0010XX00B

Δ 7: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
		▲1		▲2					▲3			Δ4

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 00000X10B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
		▲1	▲2		▲3				▲4			Δ5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 00000X10B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
								Δ1

Δ 1: IICSn = 00000001B

Remark Δ: Generated only when SPIEn = 1
n = 0, 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3		Δ4

▲1: IICSn = 0101X110B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0001X000B

▲3: IICSn = 0001X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICSn = 0101X110B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0001X100B

▲3: IICSn = 0001XX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) When arbitration loss occurs during transmission of extension code

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0010X000B

▲3: IICSn = 0010X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0010X110B

▲3: IICSn = 0010X100B

▲4: IICSn = 0010XX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1					Δ2

▲1: IICSn = 01000110B (Example: when ALDn is read during interrupt servicing)

Δ 2: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
n = 0, 1

(b) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
		▲1						Δ2

▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

IICCn’s LRELn is set to 1 by software

Δ 2: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

(c) When arbitration loss occurs during data transfer

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2				Δ3

▲1: IICSn = 10001110B

▲2: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2			Δ3

▲1: IICSn = 10001110B

▲2: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

n = 0, 1

(d) When loss occurs due to restart condition during data transfer

<1> Not extension code (Example: mismatches with SVAn)

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1					▲2			∆3

▲1: IICSn = 1000X110B

▲2: IICSn = 01000110B (Example: when ALDn is read during interrupt servicing)

∆ 3: IICSn = 00000001B

Remark ▲: Always generated
∆: Generated only when SPIEn = 1
X: don't care
Dn = D6 to D0
n = 0, 1

<2> Extension code

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1				▲2				∆3

▲1: IICSn = 1000X110B

▲2: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

IICCn's LRELn is set to 1 by software

∆ 3: IICSn = 00000001B

Remark ▲: Always generated
∆: Generated only when SPIEn = 1
X: don't care
Dn = D6 to D0
n = 0, 1

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	RW	AK	D7 to Dn	SP

▲1 (after AK) △2 (after SP)

▲1: IICSn = 1000X110B

△ 2: IICSn = 01000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
Dn = D6 to D0
n = 0, 1

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

When WTIMn = 1

STTn = 1 ↓ (after the second AK)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 △4

▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

▲3: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

△ 4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

When WTIMn = 1

STTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 Δ3

▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

Δ 3: IICSn = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

When WTIMn = 1

SPTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

▲3: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

10.3.6 Interrupt request (INTIICn) generation timing and wait control

The setting of bit 3 (WTIMn) of IIC control register n (IICCn) determines the timing by which INTIICn is generated and the corresponding wait control, as shown below (n = 0, 1).

Table 10-4. INTIICn Generation Timing and Wait Control

WTIMn	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 INTIICn signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register n (SVAn).
At this point, $\overline{\text{ACK}}$ is output regardless of the value set to bit 2 (ACKEn) of IICCn. For a slave device that has received an extension code, INTIICn occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of slave address register n (SVAn), neither INTIICn nor a wait occurs.

Remarks 1. 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.

2. n = 0, 1

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined regardless of the WTIMn bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WRELn) of IIC control register n (IICCn) to 1
- By writing to IIC shift register n (IICn)
- By start condition setting (bit 1 (STTn) of IIC control register n (IICCn) = 1)
- By step condition setting (bit 0 (SPTn) of IIC control register n (IICCn) = 1)

When an 8-clock wait has been selected (WTIMn = 0), the output level of $\overline{\text{ACK}}$ must be determined prior to wait cancellation.

Remark n = 0, 1

(5) Stop condition detection

INTIICn is generated when a stop condition is detected.

Remark n = 0, 1

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 (INTIICn) occurs when a local address has been set to slave address register n (SVAn) and when the address set to SVAn matches the slave address sent by the master device, or when an extension code has been received (n = 0, 1).

10.3.8 Error detection

In I^2C bus mode, the status of the serial data bus (SDAn) during data transmission is captured by IIC shift register n (IICn) of the transmitting device, so the IICn data prior to transmission can be compared with the transmitted IICn data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match (n = 0, 1).

10.3.9 Extension code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXCn) is set for extension code reception and an interrupt request (INTIICn) is issued at the falling edge of the eighth clock (n = 0, 1).
The local address stored in slave address register n (SVAn) is not affected.

(2) If 11110xx0 is set to SVAn by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that INTIICn occurs at the falling edge of the eighth clock (n = 0, 1).

- Higher 4 bits of data match: EXCn = 1 [Note]
- 7 bits of data match: COIn = 1 [Note]

Note EXCn: Bit 5 of IIC status register n (IICSn)
COIn: Bit 4 of IIC status register n (IICSn)

(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 (LRELn) of IIC control register n (IICCn) to 1 and the CPU will enter the next communication wait state.

Table 10-5. 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 STTn is set to 1 before STDn 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 (n = 0, 1).

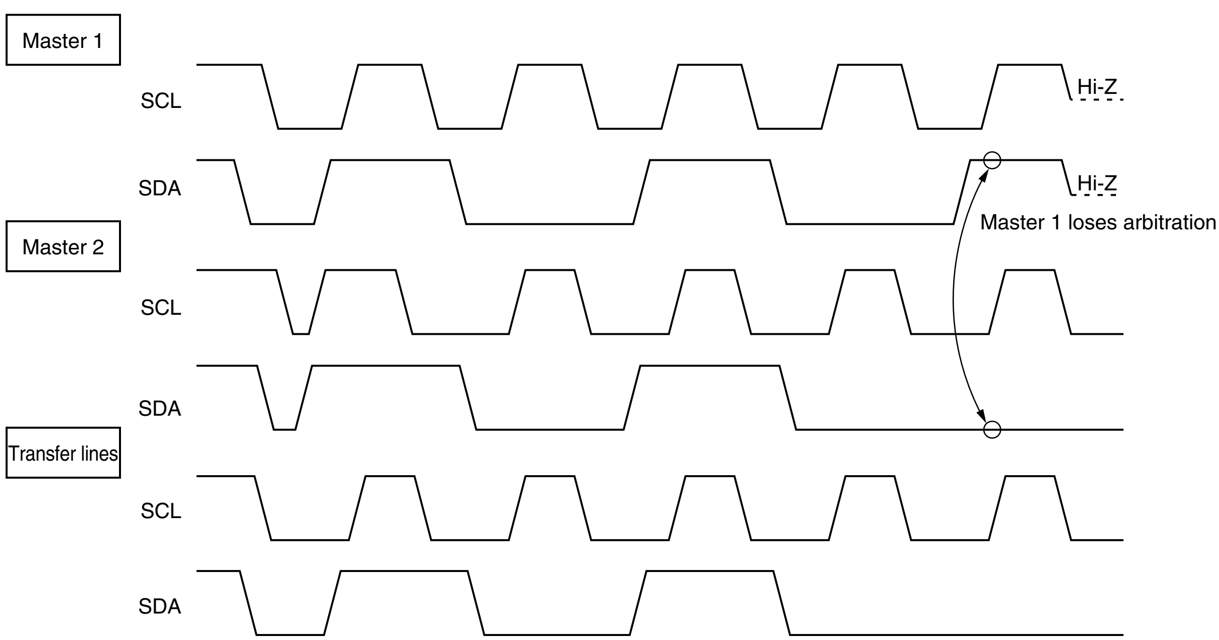
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in IIC status register n (IICSn) is set at the timing by which the arbitration loss occurred, and the SCLn and SDAn lines are both set for high impedance, which releases the bus (n = 0, 1).

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 ALDn = 1 setting that has been made by software (n = 0, 1).

For details of interrupt request timing, see **10.3.5 I²C interrupt requests (INTIICn)**.

Note STDn: Bit 1 of IIC status register n (IICSn)
STTn: Bit 1 of IIC control register n (IICCn)

Figure 10-15. Arbitration Timing Example



Remark n = 0, 1

Table 10-6. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer Note 1
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ signal transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIEn = 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 SPIEn = 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 SCLn is at low level while attempting to output a restart condition	

Notes 1. When WTIMn (bit 3 of the IIC control register n (IICCn)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock (n = 0, 1).

2. When there is a possibility that arbitration will occur, set SPIEn = 1 for master device operation (n = 0, 1).

Remark SPIEn: Bit 5 of IIC control register n (IICCn)

10.3.11 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request (INTIICn) 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 (SPIEn) of IIC control register n (IICCn) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled (n = 0, 1).

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 ($\overline{\text{ACK}}$ is not returned and the bus was released when bit 6 (LRELn) of IIC control register n (IICCn) was set to 1) (n = 0, 1).

If bit 1 (STTn) of IICCn 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 n (IICn) causes the master's address transfer to start. At this point, bit 4 (SPIEn) of IICCn should be set (n = 0, 1).

When STTn has been set, the operation mode (as start condition or as communication reservation) is determined according to the bus status (n = 0, 1).

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 STTn, set STTn, wait for the wait period, then check the MSTSn (bit 7 of IIC status register n (IICSn)) (n = 0, 1).

Wait periods, which should be set via software, are listed in Table 10-7. These wait periods can be set via the settings for bits 3, 1, and 0 (SMCn, CLn1, and CLn0) of IIC clock selection register n (IICCLn) (n = 0, 1).

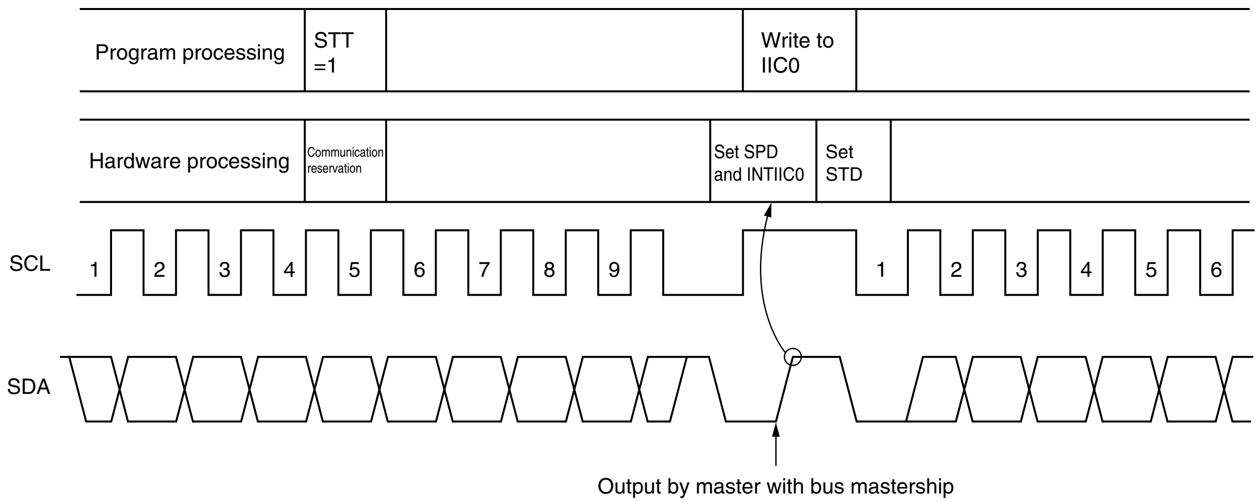
Table 10-7. Wait Periods

<table>
<tr><th>SMCn</th><th>CLn1</th><th>CLn0</th><th>Wait Period</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>26 clocks</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>46 clocks</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>92 clocks</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>37 clocks</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td rowspan="2">16 clocks</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>32 clocks</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>13 clocks</td></tr>
</table>

Remark n = 0, 1

The communication reservation timing is shown below.

Figure 10-16. Communication Reservation Timing

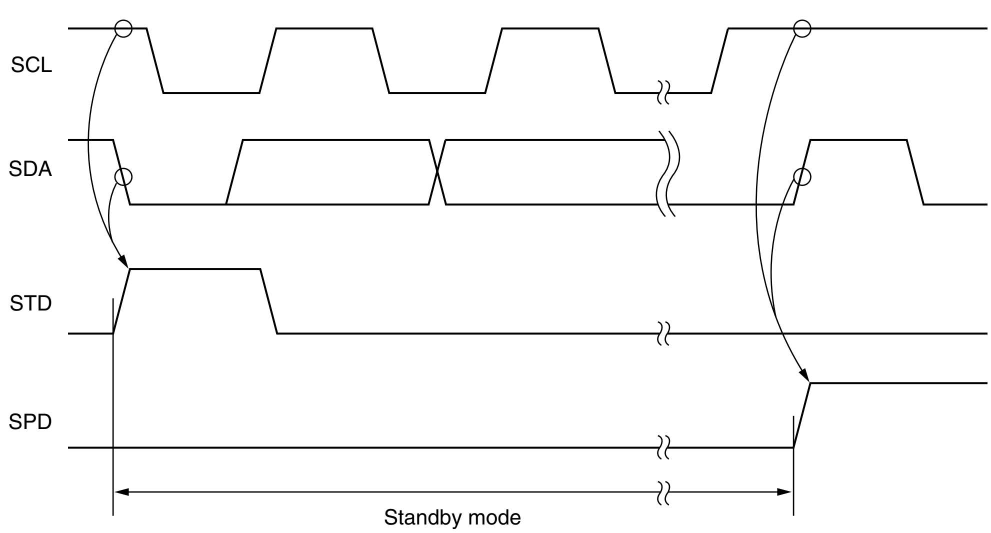


IICn: IIC shift register n
STTn: Bit 1 of IIC control register n (IICCn)
STDn: Bit 1 of IIC status register n (IICSn)
SPDn: Bit 0 of IIC status register n (IICSn)

Remark n = 0, 1

Communication reservations are acknowledged at the following timing. After bit 1 (STDn) of IIC status register n (IICSn) is set to 1, a communication reservation can be made by setting bit 1 (STTn) of IIC control register n (IICCn) to 1 before a stop condition is detected (n = 0, 1).

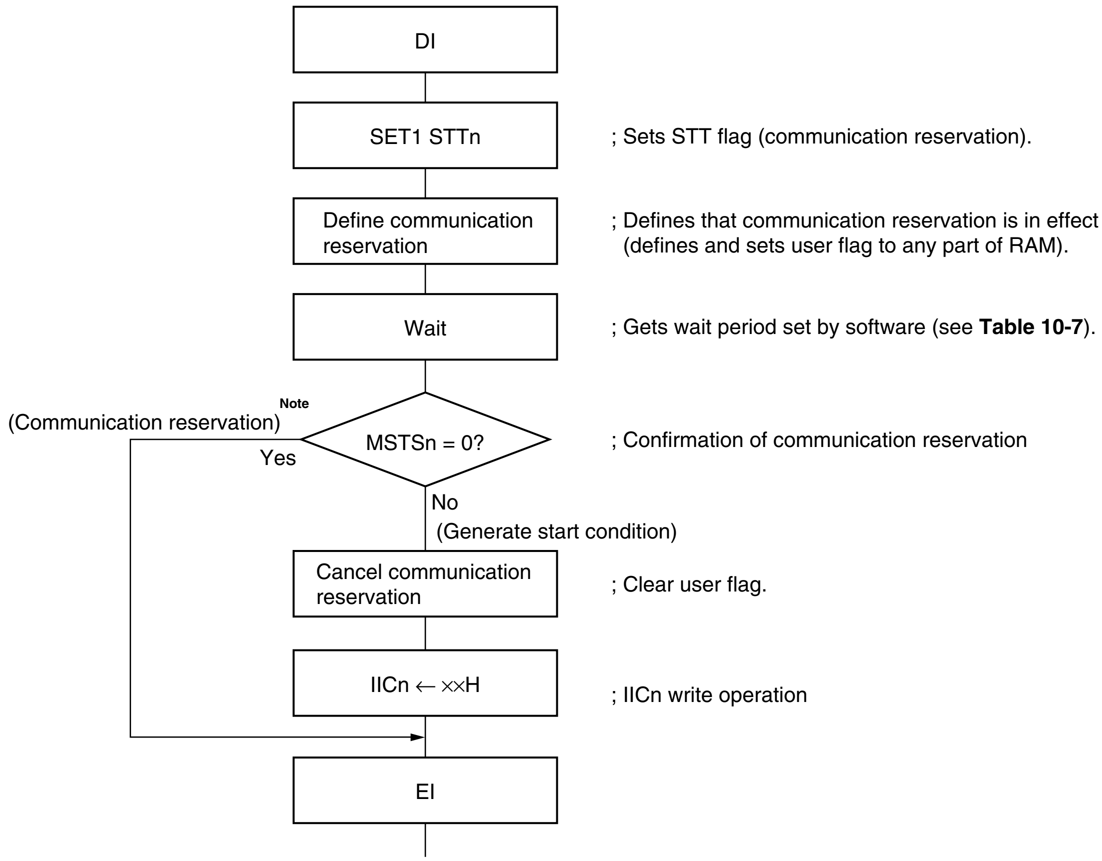
Figure 10-17. Timing for Acknowledging Communication Reservations



Remark n = 0, 1

The communication reservation flowchart is illustrated below.

Figure 10-18. Communication Reservation Flowchart



Note The communication reservation operation executes a write to IIC shift register n (IICn) when a stop condition interrupt request occurs.

Remark n = 0, 1

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 selection register n (IICCLn).
(b) Set bit 7 (IICEn) of IIC control register n (IICCn).
(c) Set bit 0 of IICCn.

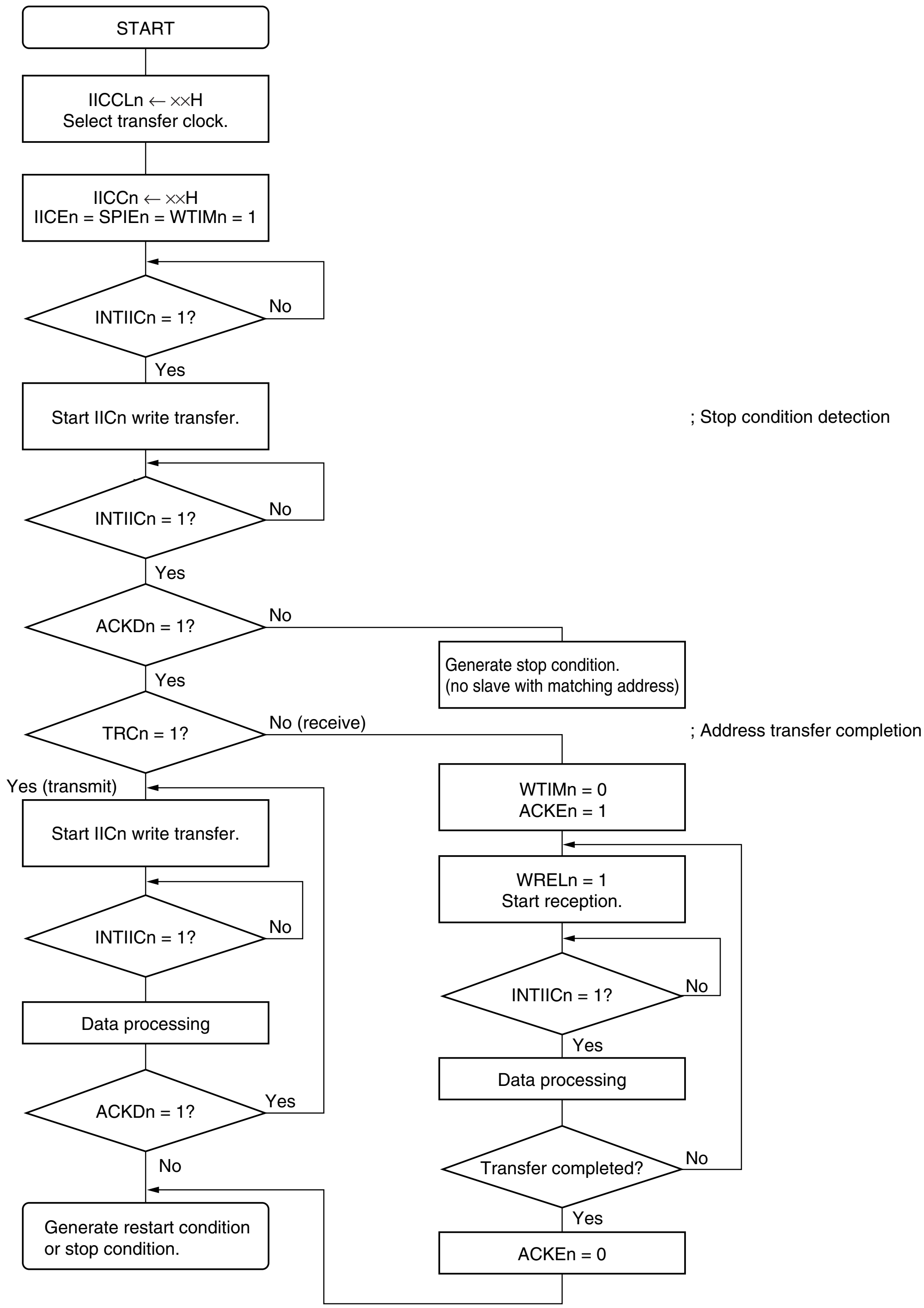
Remark n = 0, 1

10.3.14 Communication operations

(1) Master operations

The following is a flowchart of the master operations.

Figure 10-19. Master Operation Flowchart

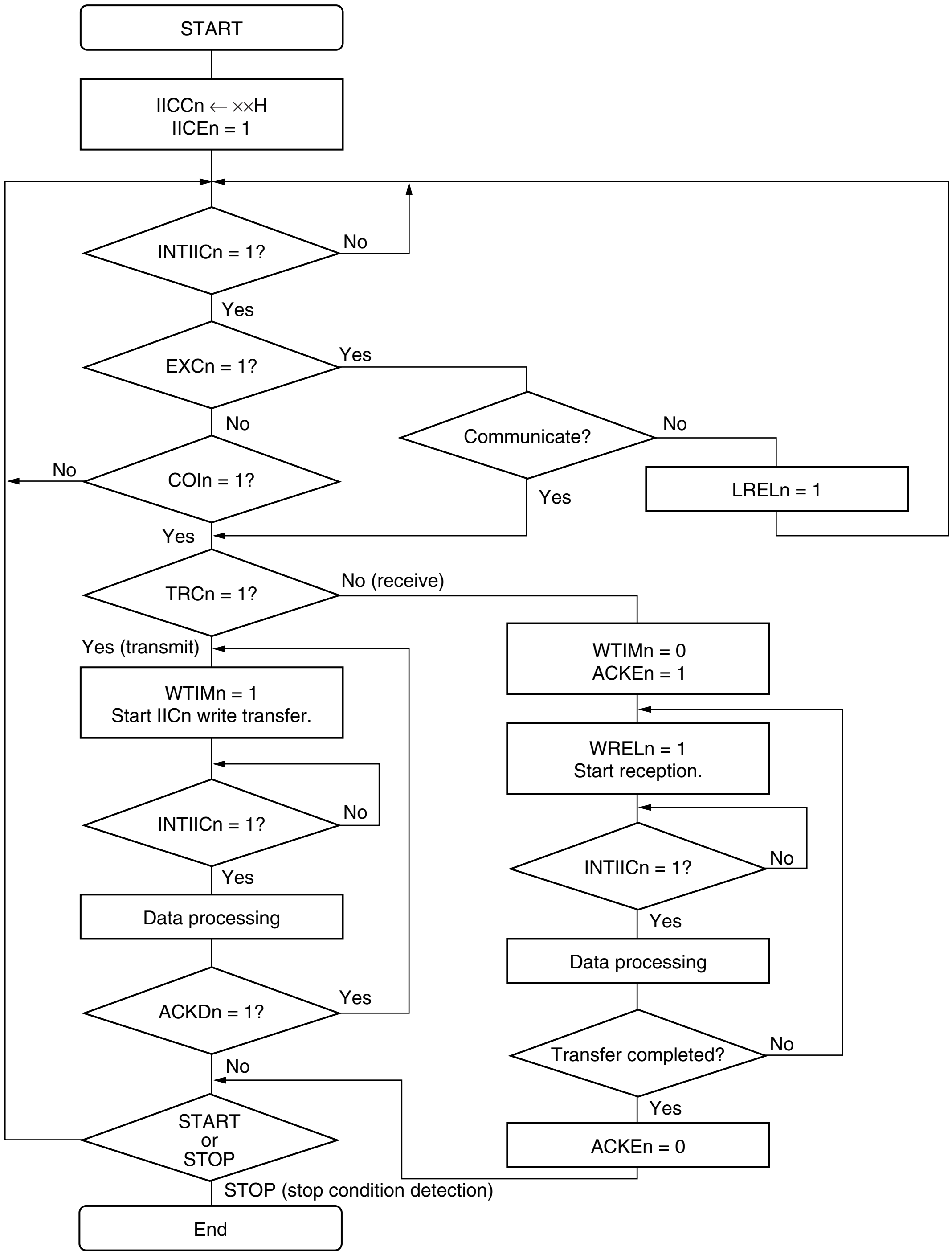


Remark n = 0, 1

(2) Slave operation

An example of slave operation is shown below.

Figure 10-20. Slave Operation Flowchart



Remark n = 0, 1

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 TRCn bit (bit 3 of IIC status register n (IICSn)) that specifies the data transfer direction and then starts serial communication with the slave device.

The shift operation of IIC bus shift register n (IICn) is synchronized with the falling edge of the serial clock (SCLn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAn pin.

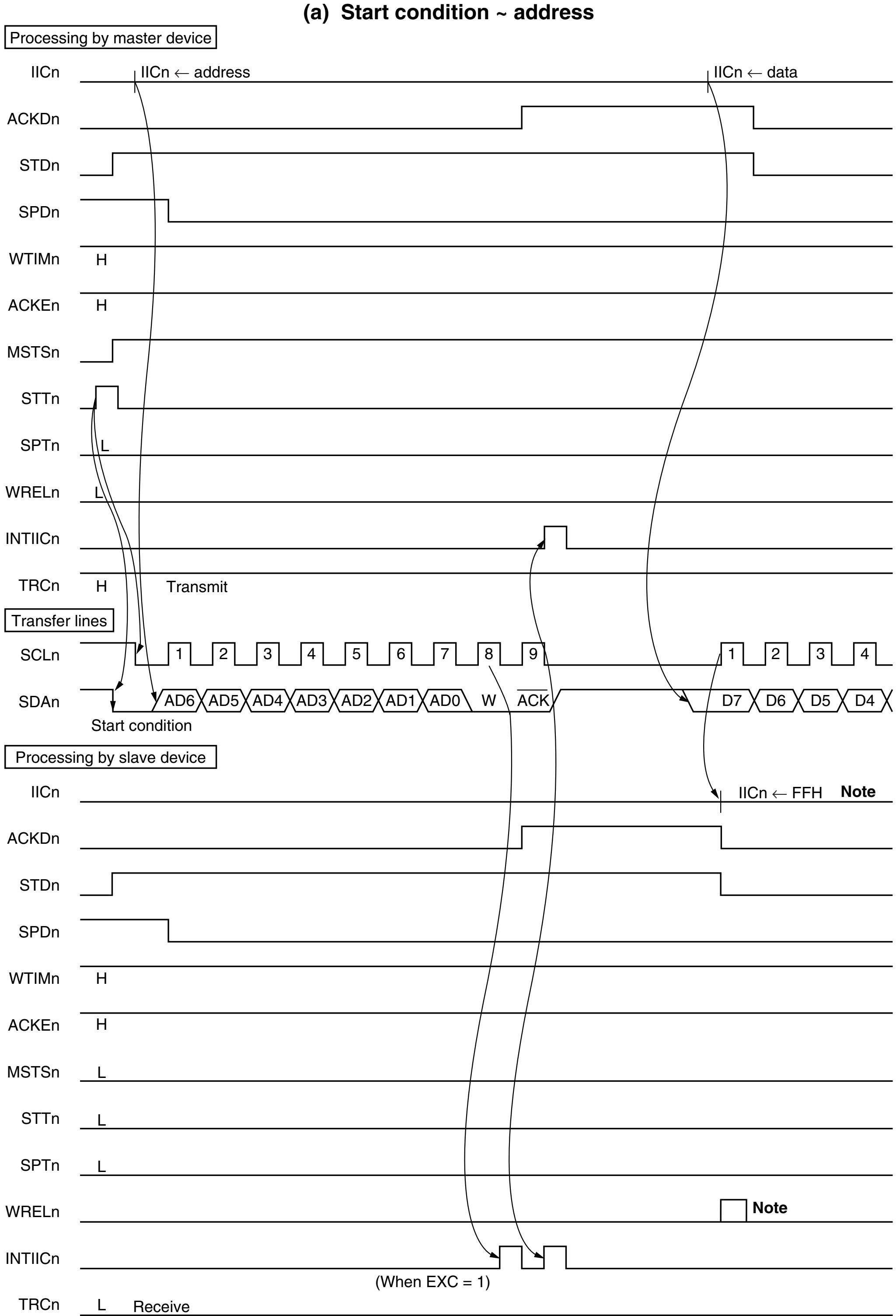
Data input via the SDAn pin is captured by IICn at the rising edge of SCLn.

The data communication timing is shown below.

Remark n = 0, 1

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

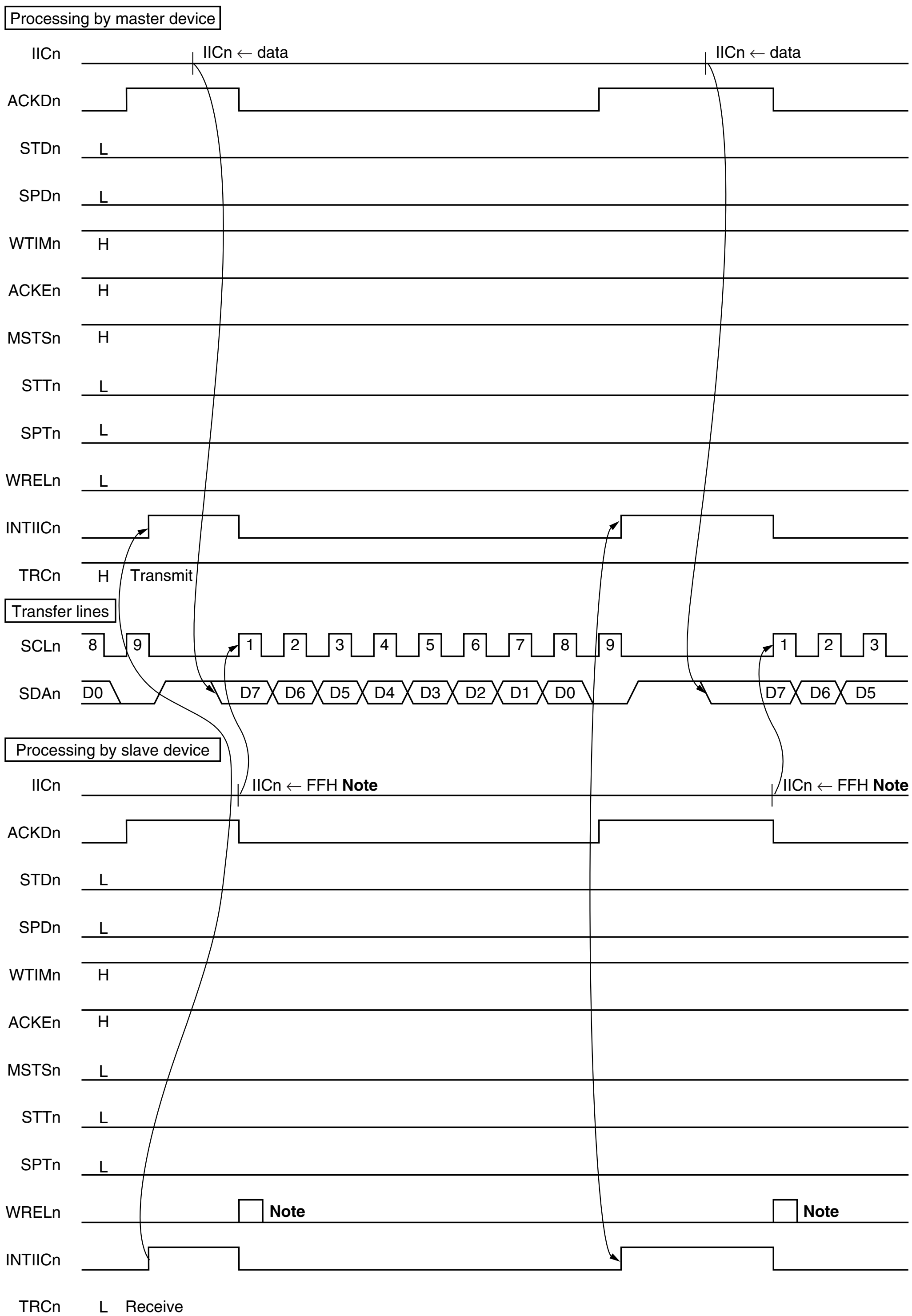


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

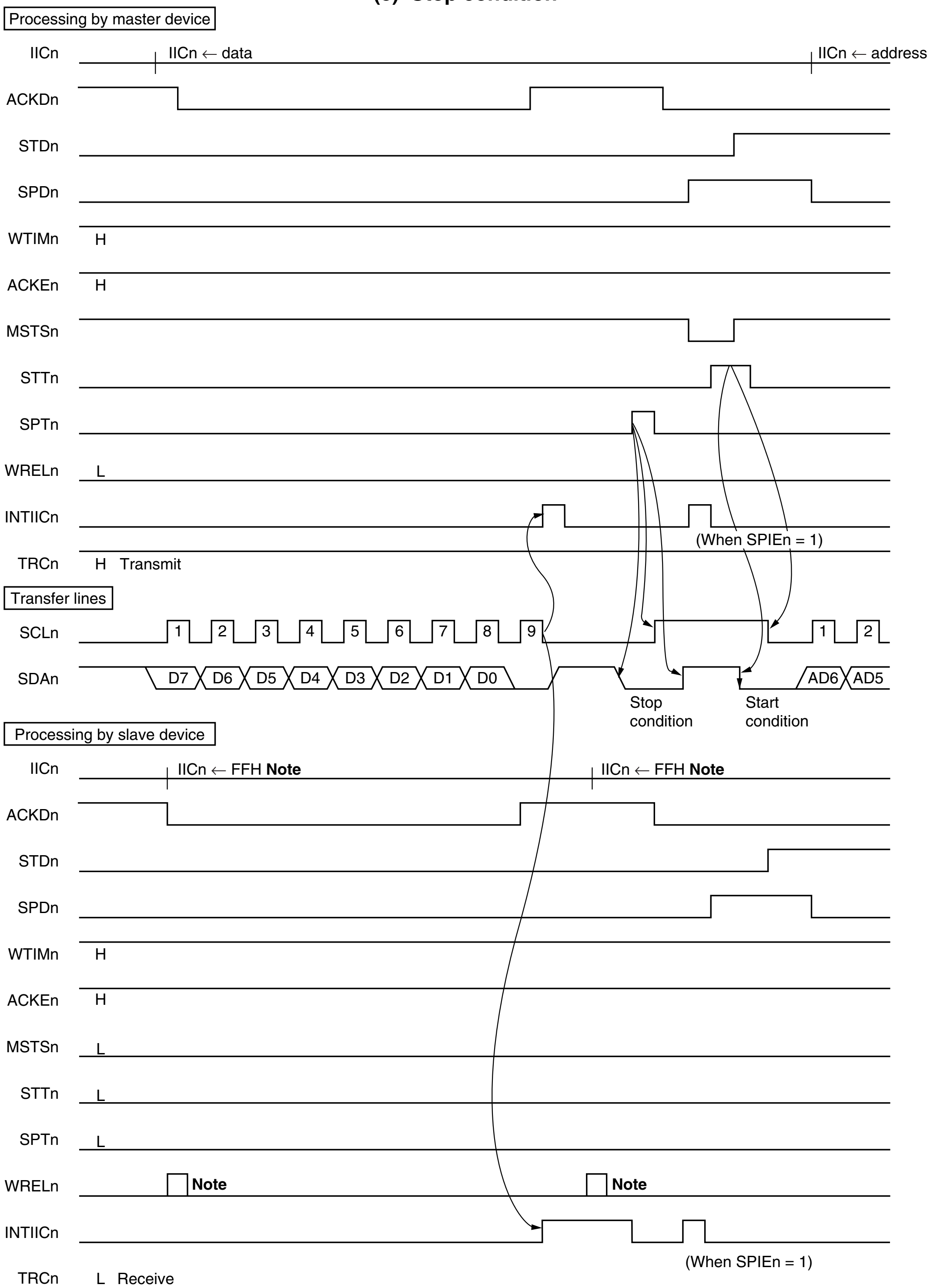


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

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

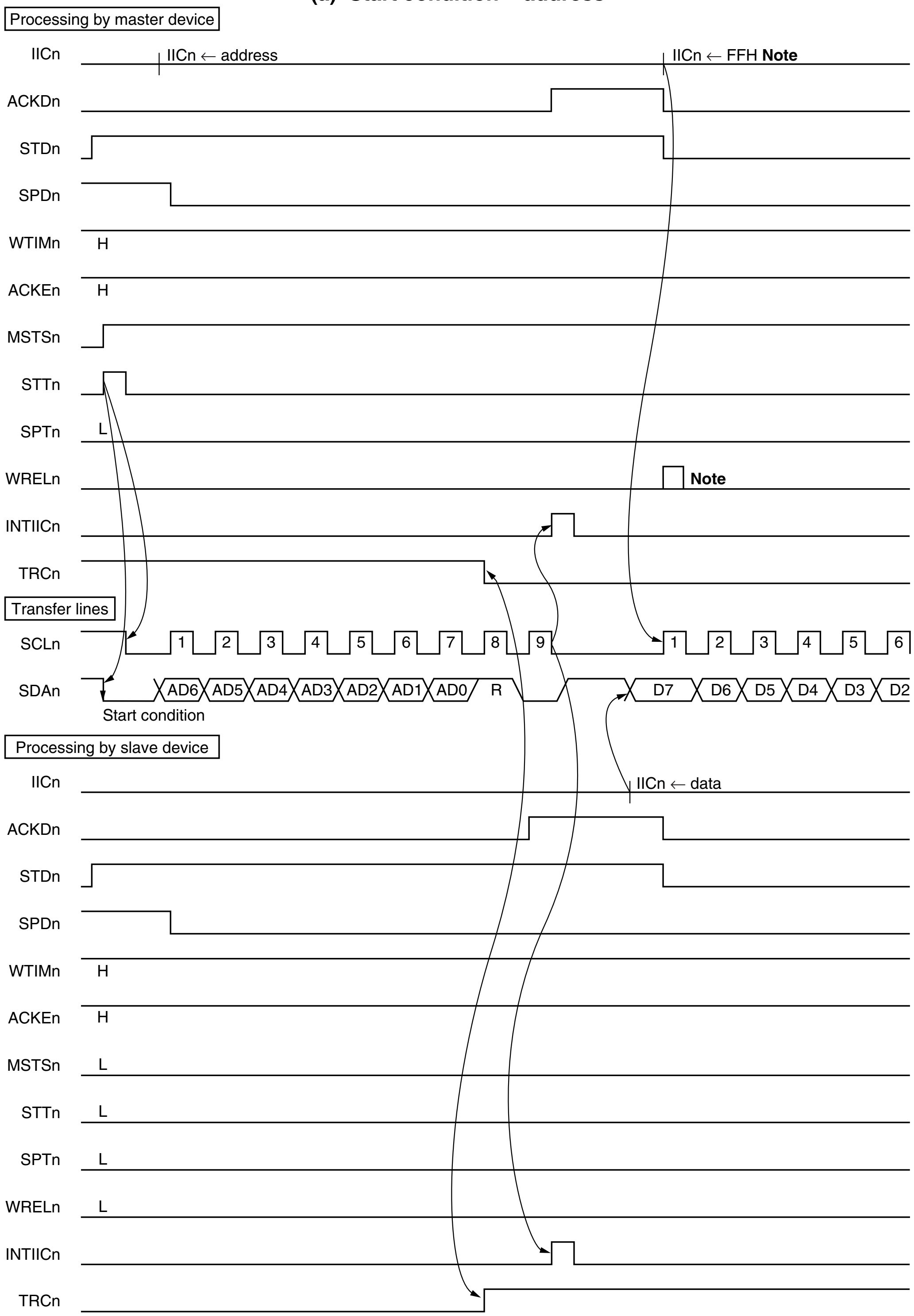


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address

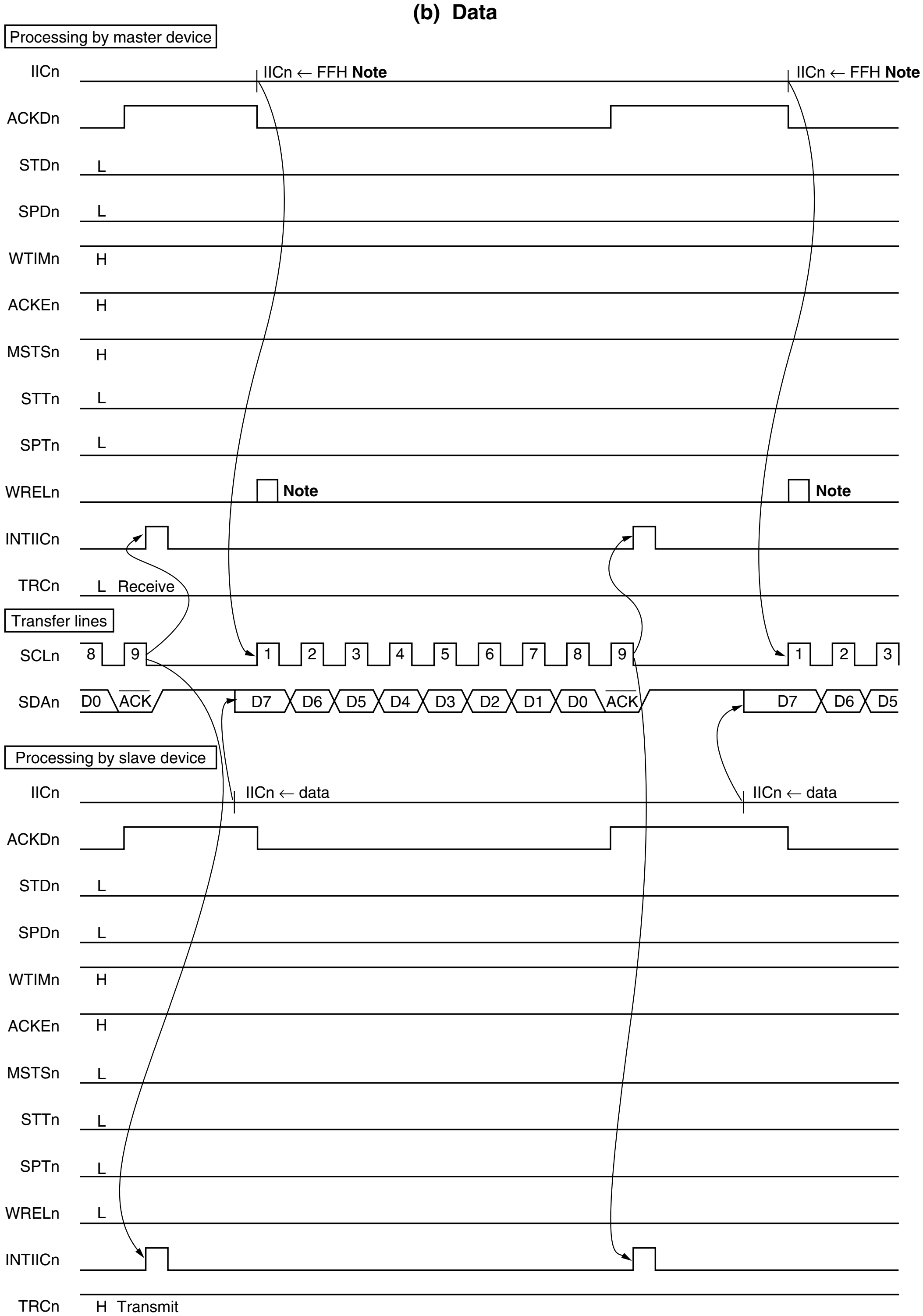


Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-22. Example of Slave to Master Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

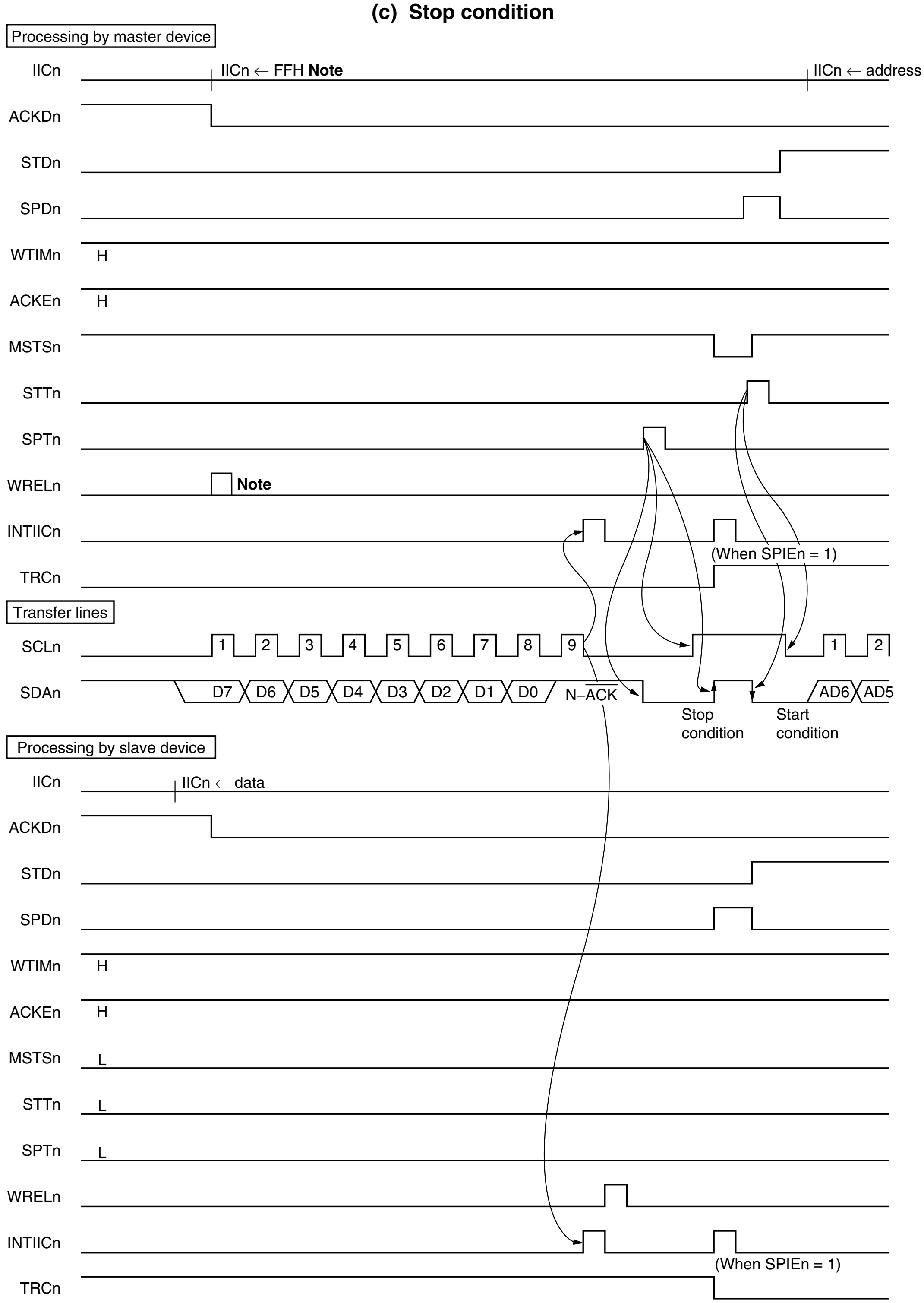


Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-22. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition



Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

★ 10.4 I^{2}C Bus (B and H Versions)

To use the I^{2}C bus function, set the P10/SDA0, P12/SCL0, P20/SDA1, and P22/SCL1 pins to N-ch open drain output.

The products with an on-chip I^{2}C bus are shown below.

- V850/SB1: μPD703030BY, 703031BY, 703032BY, 703033BY, 70F3030BY, 70F3032BY, 70F3033BY
- V850/SB2: μPD703034BY, 703035BY, 703036HY, 703037HY, 70F3035BY, 70F3036HY, 70F3037HY

The I^{2}C0 and I^{2}C1 have the following two modes.

- Operation stop mode
- I^{2}C (Inter IC) bus mode (multimaster 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) I^{2}C bus mode (multimaster support)

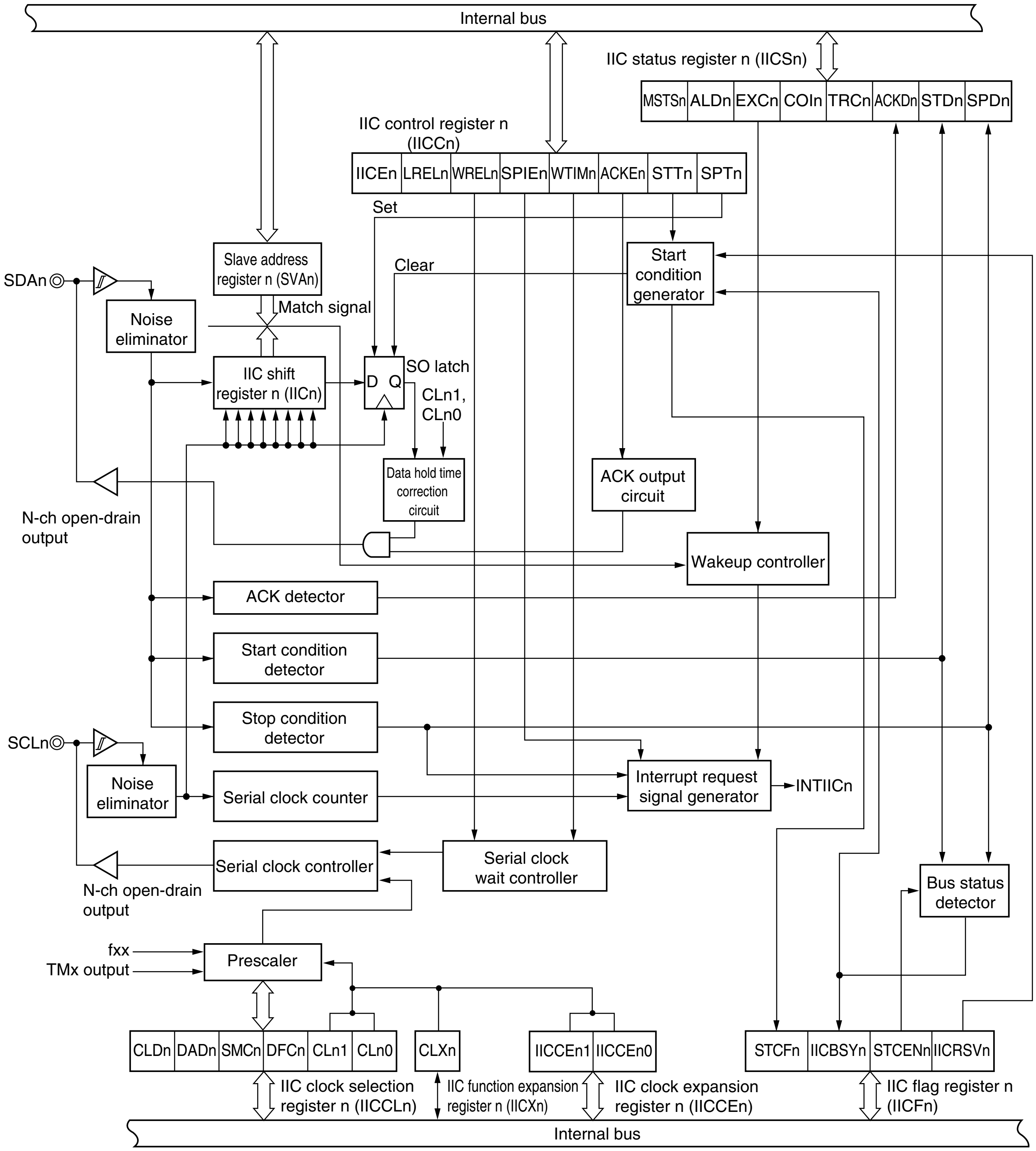
This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLn) line and a serial data bus (SDAn) line.

This mode complies with the I^{2}C bus format and the master device can output “start condition”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received data by hardware. This function can simplify the part of application program that controls the I^{2}C bus.

Since SCLn and SDAn are open drain outputs, the I^{2}Cn requires pull-up resistors for the serial clock line and the serial data bus line.

Remark n = 0, 1

Figure 10-23. Block Diagram of I²C



Remarks 1. n = 0, 1

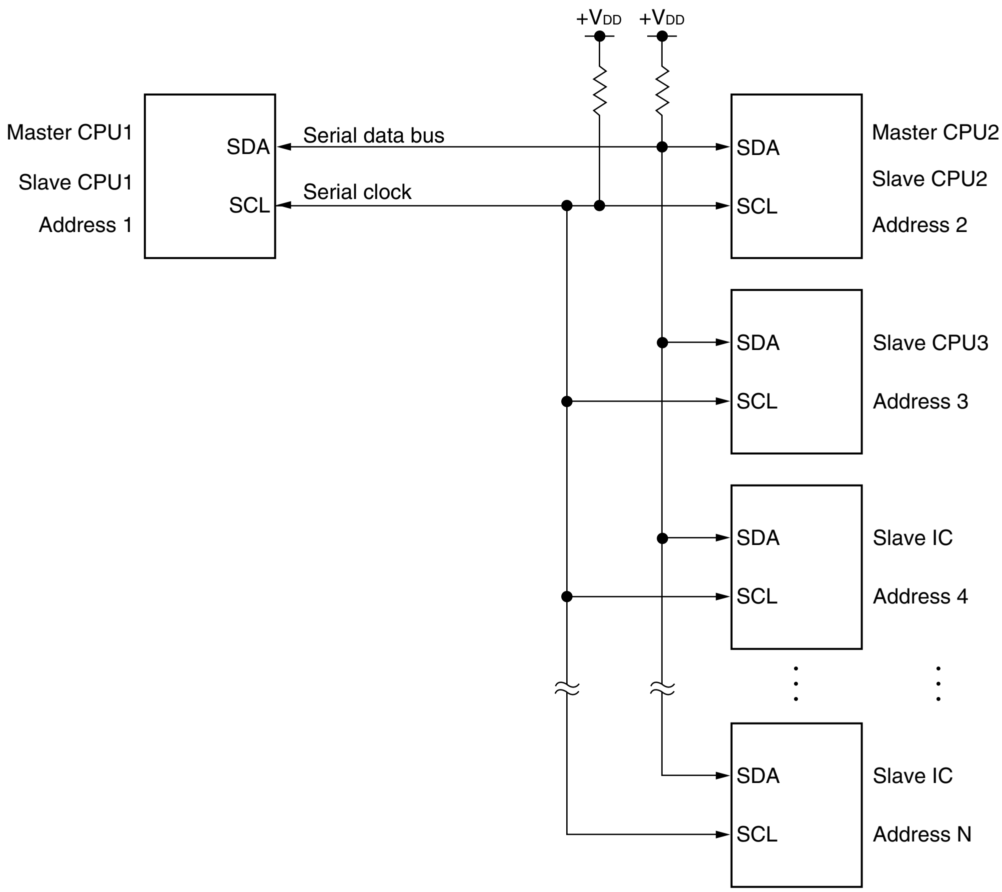
2. TMx output

n = 0: TM2 output

n = 1: TM3 output

A serial bus configuration example is shown below.

Figure 10-24. Serial Bus Configuration Example Using I²C Bus



10.4.1 Configuration

I^2Cn includes the following hardware (n = 0, 1).

Table 10-8. Configuration of I^2Cn

Item	Configuration
Registers	IIC shift registers 0 and 1 (IIC0, IIC1) Slave address registers 0 and 1 (SVA0, SVA1)
Control registers	IIC control registers 0 and 1 (IICC0, IICC1) IIC status registers 0 and 1 (IICS0, IICS1) IIC flag registers 0 and 1 (IICF0, IICF1) IIC clock selection registers 0 and 1 (IICCL0, IICCL1) IIC clock expansion registers 0 and 1 (IICCE0, IICCE1) IICC function expansion registers 0 and 1 (IICX0, IICX1)

(1) IIC shift registers 0 and 1 (IIC0, IIC1)

IICn is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data.
IICn can be used for both transmission and reception (n = 0, 1).
Write and read operations to IICn are used to control the actual transmit and receive operations.
IICn is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears IIC0 and IIC1 to 00H.

(2) Slave address registers 0 and 1 (SVA0, SVA1)

SVAn sets local addresses when in slave mode.
SVAn is set by an 8-bit memory manipulation instruction (n = 0, 1).
$\overline{\text{RESET}}$ input clears SVA0 and SVA1 to 00H.

(3) SO latch

The SO latch is used to retain the SDAn pin's output level (n = 0, 1).

(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 n (SVAn) or when an extension code is received (n = 0, 1).

(5) Clock selector

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICn).

An I^{2}C interrupt is generated following either of two triggers.

- Eighth or ninth clock of the serial clock (set by WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by SPIEn bit)

Remarks 1. n = 0, 1

2. WTIMn bit: Bit 3 of IIC control register n (IICCn)

SPIEn bit: Bit 4 of IIC control register n (IICCn)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCLn pin from a sampling clock (n = 0, 1).

(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.

(12) Start condition generator

This circuit generates a start condition when the STTn bit is set.

However, in the communication reservation disabled status (IICRSVn = 1), when the bus is not released (IICBSYn = 1), start condition requests are ignored and the STCFn bit is set.

Remark IICRSVn bit: Bit 0 of IIC flag register n (IICFn)

IICBSYn bit: Bit 6 of IIC flag register n (IICFn)

STCFn bit: Bit 7 of IIC flag register n (IICFn)

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCENn bit.

Remark STCENn bit: Bit 1 of IIC flag register n (IICFn)

10.4.2 I^{2}C control register

I^{2}C0 and I^{2}C1 are controlled by the following registers.

- IIC control registers 0, 1 (IICC0, IICC1)
- IIC status registers 0, 1 (IICS0, IICS1)
- IIC flag registers 0, 1 (IICF0, IICF1)
- IIC clock selection registers 0, 1 (IICCL0, IICCL1)
- IIC clock expansion registers 0, 1 (IICCE0, IICCE1)
- IIC function expansion registers 0, 1 (IICX0, IICX1)

The following registers are also used.

- IIC shift registers 0, 1 (IIC0, IIC1)
- Slave address registers 0, 1 (SVA0, SVA1)

(1) IIC control registers 0, 1 (IICC0, IICC1)

IICCn is used to enable/disable I^{2}C operations, set wait timing, and set other I^{2}C operations.
IICCn can be set by an 8-bit or 1-bit memory manipulation instruction (n = 0, 1).
$\overline{\text{RESET}}$ input clears IICCn to 00H.

Caution In I^{2}C0, I^{2}C1 bus mode, set the port 1 mode register (PM1), port 2 mode register (PM2), port 1 function register (PF1), and port 2 function register (PF2) as follows. In addition, set each output latch to 0.

Pin	Port Mode Register	Port Function Register
P10/SI0/SDA0	PM10 of PM1 register = 0	PF10 of PF1 register = 1
P12/$\overline{\text{SCK0}}$/SCL0	PM12 of PM1 register = 0	PF12 of PF1 register = 1
P20/SI2/SDA1	PM20 of PM2 register = 0	PF20 of PF2 register = 1
P22/$\overline{\text{SCK2}}$/SCL1	PM22 of PM2 register = 0	PF22 of PF2 register = 1

(1/4)

After reset: 00H R/W Address: FFFFF340H, FFFFF350H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

(n = 0, 1)

IICEn	I^{2}Cn operation enable/disable specification	
0	Stops operation. Presets IIC status register n (IICSn). Stops internal operation.	
1	Enables operation.	
Condition for clearing (IICEn = 0)		Condition for setting (IICEn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input		• Set by instruction

LRELn	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 SCLn and SDAn lines are set to high impedance. The following flags are cleared. • STDn • ACKDn • TRCn • COIn • EXCn • MSTSn • STTn • SPTn	
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 (LRELn = 0)[Note]		Condition for setting (LRELn = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input		• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark STDn: Bit 1 of IIC state register n (IICSn)
ACKDn: Bit 2 of IIC state register n (IICSn)
TRCn: Bit 3 of IIC state register n (IICSn)
COIn: Bit 4 of IIC state register n (IICSn)
EXCn: Bit 5 of IIC state register n (IICSn)
MSTSn: Bit 7 of IIC state register n (IICSn)

(2/4)

WRELn	Wait cancellation control	
0	Does not cancel wait	
1	Cancels wait. This setting is automatically cleared after wait is canceled.	
Condition for clearing (WRELn = 0)[Note]		Condition for setting (WRELn = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input		• Set by instruction

SPIEn	Enable/disable generation of interrupt request when stop condition is detected	
0	Disable	
1	Enable	
Condition for clearing (SPIEn = 0)[Note]		Condition for setting (SPIEn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input		• Set by instruction

WTIMn	Control of wait and interrupt request generation	
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.	
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.	
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 (WTIMn = 0)[Note]		Condition for setting (WTIMn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input		• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark n = 0, 1

(3/4)

<table>
<tr><th>ACKEn</th><th colspan="2">Acknowledge control</th></tr>
<tr><td>0</td><td colspan="2">Disable acknowledge.</td></tr>
<tr><td>1</td><td colspan="2">Enable acknowledge. 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 EXCn = 1.</td></tr>
<tr><td colspan="2">Condition for clearing (ACKEn = 0)[Note]</td><td>Condition for setting (ACKEn = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction
• When $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

<table>
<tr><th>STTn</th><th colspan="2">Start condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Does not generate a start condition.</td></tr>
<tr><td>1</td><td colspan="2">When bus is released (in STOP mode):
Generates a start condition (for starting as master). The SDAn 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, SCLn is changed to low level.
When bus is not used:
• When communication reservation function is enabled (IICRSVn = 0)
Functions as the start condition reservation flag. When set, automatically generates a start condition after the bus is released.
• When communication reservation function is disabled (IICRSVn = 1)
The STCFn bit is set. No start condition is generated.
In the wait state (when master device)
Generates a restart condition after releasing the wait.</td></tr>
<tr><td colspan="3">Cautions concerning set timing
• For master reception: Cannot be set during transfer. Can be set only when ACKEn has been set to 0 and slave has been notified of final reception.
• For master transmission: A start condition cannot be generated normally during the ACKn period. Set during the wait period.
• For slave: The communication reserved status is entered even if the communication reservation function is disabled (IICRSVn = 1).
• Cannot be set at the same time as SPTn</td></tr>
<tr><td colspan="2">Condition for clearing (STTn = 0)</td><td>Condition for setting (STTn = 1)</td></tr>
<tr><td colspan="2">• Cleared by loss in arbitration
• Cleared after start condition is generated by master device
• When LRELn = 1
• When IICEn = 0
• Cleared when $\overline{\text{RESET}}$ is input</td><td>• Set by instruction</td></tr>
</table>

Note This flag's signal is invalid when IICEn = 0.

Remarks 1. Bit 1 (STTn) is 0 if it is read immediately after data setting.

2. IICRSVn: Bit 0 of IIC flag register n (IICFn)

STCFn: Bit 7 of IIC flag register n (IICFn)

3. n = 0, 1

(4/4)

<table>
<tr><th>SPTn</th><th colspan="2">Stop condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device’s transfer).
After the SDAn line goes to low level, either set the SCLn line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDAn line is changed from low level to high level and a stop condition is generated.</td></tr>
<tr><td colspan="3">Cautions concerning setting timing
• For master reception: Cannot be set during transfer.
Can be set only when ACKEn has been set to 0 and during the wait period after slave has been notified of final reception.
• For master transmission: A stop condition cannot be generated normally during the ACKn period. Set during the wait period.
• Cannot be set at the same time as STTn.
• SPTn can be set only when in master mode[Note]
• When WTIMn has been set to 0, if SPTn 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, WTIMn should be changed from 0 to 1 during the wait period following output of eight clocks, and SPTn should be set during the wait period that follows output of the ninth clock.</td></tr>
<tr><td colspan="2">Condition for clearing (SPTn = 0)</td><td>Condition for setting (SPTn = 1)</td></tr>
<tr><td colspan="2">• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• When LRELn = 1
• When IICEn = 0
• Cleared when RESET is input</td><td>• Set by instruction</td></tr>
</table>

Note Set SPTn only in master mode. However, when the IICRSVn bit of IIC flag register n (IICFn) is 0, SPTn 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 **10.4.13 Cautions**.

Caution When bit 3 (TRCn) of IIC status register n (IICSn) is set to 1, WRELn is set during the ninth clock and wait is canceled, after which TRCn is cleared and the SDAn line is set to high impedance.

Remarks **1.** Bit 0 (SPTn) is 0 if it is read immediately after data setting.

2. n = 0, 1

(2) IIC status registers 0, 1 (IICS0, IICS1)

IICSn indicates the status of the I^2Cn bus.

IICSn can be set by an 8-bit or 1-bit memory manipulation instruction. IICSn is a read-only register (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICSn to 00H.

(1/3)

After reset: 00H R Address: FFFFF342H, FFFFF352H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

(n = 0, 1)

MSTSn	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is generated

ALDn	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”. MSTSn is cleared.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after IICSn is read[Note] • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ 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 IICSn.

Remark LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

(2/3)

EXCn	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXCn = 0)	Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COIn = 0)	Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVAn) (set at the rising edge of the eighth clock).

TRCn	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAn line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDAn line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRCn = 0)	Condition for setting (TRCn = 1)
• When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • Cleared by WRELn = 1[Note] • When ALDn 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 TRCn is cleared and SDAn line become high impedance when bit 5 (WRELn) of IIC control register n (IICCn) is set and wait state is released at ninth clock with bit 3 (TRCn) of IIC status register n (IICSn) = 1.

Remarks 1. WRELn: Bit 5 of IIC control register n (IICCn)
LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

2. n = 0, 1

(3/3)

ACKDn	Detection of $\overline{\text{ACK}}$
0	$\overline{\text{ACK}}$ was not detected.
1	$\overline{\text{ACK}}$ was detected.
Condition for clearing (ACKDn = 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 LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• After the SDAn line is set to low level at the rising edge of the SCLn's ninth clock

STDn	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 (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	When a start condition is detected

SPDn	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 (SPDn = 0)	Condition for setting (SPDn = 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 IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	When a stop condition is detected

Remarks 1. LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

2. n = 0, 1

(3) IIC flag registers 0, 1 (IICF0, IICF1)

IICFn is used to set the I^2Cn operation mode and to indicate the I^2C bus status.

IICFn can be set by an 8-bit or 1-bit memory manipulation instruction.

The IICRSVn bit is used to enable/disable the communication reservation function (see **10.4.12 Communication reservation**).

The STCENn bit is used to set the initial value of the IICBSYn bit (see **10.4.13 Cautions**).

The IICRSVn and STCENn bits can be written only when the I^2Cn operation is disabled (IICEn bit of IIC control register n (IICCn) = 0). After operation is enabled, the IICFn register can be read (n = 0).

$\overline{\text{RESET}}$ input clears IICFn to 00H.

When IICFn = 00H, these registers operate in the same way as the A versions.

(1/2)

After reset: 00H R/W[Note] Address: FFFFF33CH, FFFFF33EH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

(n = 0, 1)

STCFn	STTn bit clear
0	Generate start condition
1	Clear STTn bit

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Clearing by setting STTn = 1 • $\overline{\text{RESET}}$ input	• Clearing of STTn when communication reservation is disabled (IICRSVn = 1).

IICBSYn	I^2Cn bus status
0	Bus release status
1	Bus communication status

Condition for clearing (IICBSYn = 0)	Setting conditions (IICBSYn = 1)
• Detection of stop condition • $\overline{\text{RESET}}$ input	• Detection of start condition • Setting of IICEn when STCENn = 0

Note Bits 6 and 7 are read-only bits.

Remark STTn: Bit 1 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

(2/2)

STCENn	Initial start enable trigger
0	After operation is enabled (IICEn = 1), cannot generate a start condition until a stop condition is detected.
1	After operation is enabled (IICEn = 1), can generates a start condition without detecting a stop condition.

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• Detection of start condition • $\overline{\text{RESET}}$ input	• Setting by instruction

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Clearing by instruction • $\overline{\text{RESET}}$ input	• Setting by instruction

Cautions 1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).

2. As the bus release status (IICBSYn = 0) is recognized immediately after I^{2}C operation is enabled regardless of the actual bus status when STCENn = 1, when generating the first start condition (STTn = 1), it is necessary to verify that the bus has been released in order to prevent such communications from being destroyed.

3. Write to the IICRSVn bit only when the operation is stopped (IICEn = 0).

Remark STTn: Bit 1 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

(4) IIC clock selection registers 0, 1 (IICCL0, IICCL1)

IICCLn is used to set the transfer clock for the I^2Cn bus.

IICCLn can be set by an 8-bit or 1-bit memory manipulation instruction. Bits SMCn, CLn1 and CLn0 are set using CLXn bit of IIC function expansion register n (IICXn) in combination with bits IICCEn1 and IICCEn0 of IIC clock expansion register n (IICCEn) (n = 0, 1) (see **10.4.2 (7) I^2Cn transfer clock setting method**).

$\overline{\text{RESET}}$ input clears IICCLn to 00H.

After reset: 00H R/W[Note] Address: FFFFF344H, FFFFF354H

	7	6	<5>	<4>	3	2	1	0
IICCLn	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0

(n = 0, 1)

CLDn	Detection of SCLn line level (valid only when IICEn = 1)
0	SCLn line was detected at low level.
1	SCLn line was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLn line is at low level • When IICEn = 0 • When $\overline{\text{RESET}}$ is input	• When the SCLn line is at high level

DADn	Detection of SDAn line level (valid only when IICEn = 1)
0	SDAn line was detected at low level.
1	SDAn line was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAn line is at low level • When IICEn = 0 • When $\overline{\text{RESET}}$ is input	• When the SDAn line is at high level

SMCn	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFCn switching (on/off).	

Note Bits 4 and 5 are read only bits.

Caution Be sure to set bits 7 and 6 to 0.

Remark IICEn: Bit 7 of IIC control register n (IICCn)

(5) IIC function expansion registers 0, 1 (IICX0, IICX1)

These registers set the function expansion of I^{2}Cn (valid only in high-speed mode).

IICXn is set by an 8-bit or 1-bit memory manipulation instruction. Set the CLXn bit in combination with the SMCn, DFCn, CLn1, and the CLn0 bits of IIC clock selection register n (IICCLn) and the IICCEn1 and IICCEn0 bits of IIC clock expansion register n (IICCEn) (see **10.4.2 (7) I^{2}Cn transfer clock setting method**) (n = 0, 1).

$\overline{\text{RESET}}$ input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF34AH, FFFFF35AH

	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(n = 0, 1)

(6) IIC clock expansion registers 0, 1 (IICCE0, IICCE1)

These registers set the transfer clock expansion of I^{2}Cn.

IICCEn is set by an 8-bit memory manipulation instruction. Set the IICCEn1 and IICCEn0 bits in combination with the SMCn, CLn1, and CLn0 bits of IIC clock selection register n (IICCLn) and the CLXn bit of IIC function expansion register n (IICXn) (see **10.4.2 (7) I^{2}Cn transfer clock setting method**) (n = 0, 1).

$\overline{\text{RESET}}$ input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF34CH, FFFFF35CH

	7	6	5	4	3	2	1	0
IICCEn	0	0	0	0	0	0	IICCEn1	IICCEn0

(n = 0, 1)

(7) I^{2}Cn transfer clock setting method

The I^{2}Cn transfer clock frequency (f_{SCL}) is calculated using the following expression (n = 0, 1).

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 24, 48, 36, 54, 44, 86, 172, 132, 198 (see **Table 10-9 Selection Clock Setting**.)

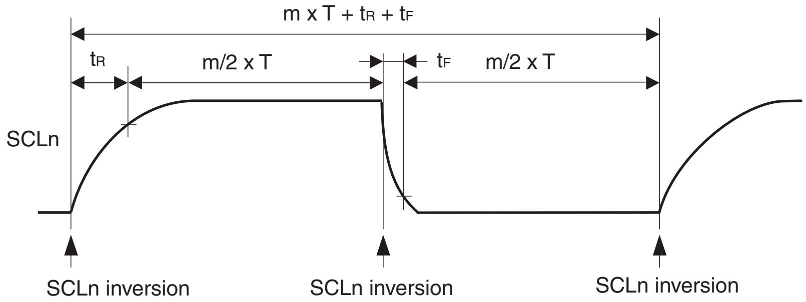
T: $1/f_{XX}$

t_R: SCLn rise time

t_F: SCLn fall time

For example, the I^{2}Cn transfer clock frequency (f_{SCL}) when f_{XX} = 20 MHz, m = 198, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(198 \times 50 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 98.5 \text{ kHz}$$



The selection clock is set using a combination of the SMCn, CLn1, and CLn0 bits of IIC clock selection register n (IICCLn), the CLXn bit of IIC function expansion register n (IICXn), and IICCEn1 and the IICCEn0 bits of IIC clock expansion register n (IICCEn) (n = 0, 1).

Table 10-9. Selection Clock Setting

IICCEn		IICXn	IICCLn			Selection Clock (f_{xx}/m)		Settable Main Clock Frequency (f_{xx}) Range	Operation Mode
Bit 1	Bit 0	Bit 0	Bit 3	Bit 1	Bit 0				
IICCEn1	IICCEn0	CLXn	SMCn	CLn1	CLn0				
x	x	1	1	0	x	f_{xx}/12		4.0 MHz to 4.19 MHz	High-speed mode (SMCn = 1)
x	x	0	1	0	x	f_{xx}/24		4.0 MHz to 8.38 MHz	
x	x	0	1	1	0	f_{xx}/48		8.0 MHz to 16.67 MHz	
0	1	0	1	1	1	f_{xx}/36		12.0 MHz to 13.4 MHz	
1	0	0	1	1	1	f_{xx}/54		16.0 MHz to 20.0 MHz[Note]	
0	0	0	1	1	1	n = 0	TM2 output/18	TM2 setting	
						n = 1	TM3 output/18	TM3 setting	
x	x	0	0	0	0	f_{xx}/44		2.0 MHz to 4.19 MHz	Normal mode (SMCn = 0)
x	x	0	0	0	1	f_{xx}/86		4.19 MHz to 8.38 MHz	
x	x	0	0	1	0	f_{xx}/172		8.38 MHz to 16.67 MHz	
0	1	0	0	1	1	f_{xx}/132		12.0 MHz to 13.4 MHz	
1	0	0	0	1	1	f_{xx}/198		16.0 MHz to 20.0 MHz[Note]	
0	0	0	0	1	1	n = 0	TM2 output/66	TM2 setting	
						n = 1	TM3 output/66	TM3 setting	
Other than above						Setting prohibited			

Note This cannot be set in the μPD703034BY, 703035BY, and 70F3035BY.

Remarks 1. n = 0, 1

2. x: don't care

3. When the output of the timer is selected as the clock, it is not necessary to set the P26/TO2/TI2 and P27/TO3/TI3 pins in the timer output mode.

(8) IIC shift registers 0, 1 (IIC0, IIC1)

IICn is used for serial transmission/reception (shift operations) that is synchronized with the serial clock. It can be read from or written to in 8-bit units, but data should not be written to IICn during a data transfer (n = 0, 1).

After reset: 00H R/W Address: FFFFF348H, FFFFF358H

	7	6	5	4	3	2	1	<0>
IICn								

(n = 0, 1)

(9) Slave address registers 0, 1 (SVA0, SVA1)

SVAn 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 as 0.

After reset: 00H R/W Address: FFFFF346H, FFFFF356H

	7	6	5	4	3	2	1	0
SVAn								0

(n = 0, 1)

10.4.3 I^2C bus mode functions

(1) Pin configuration

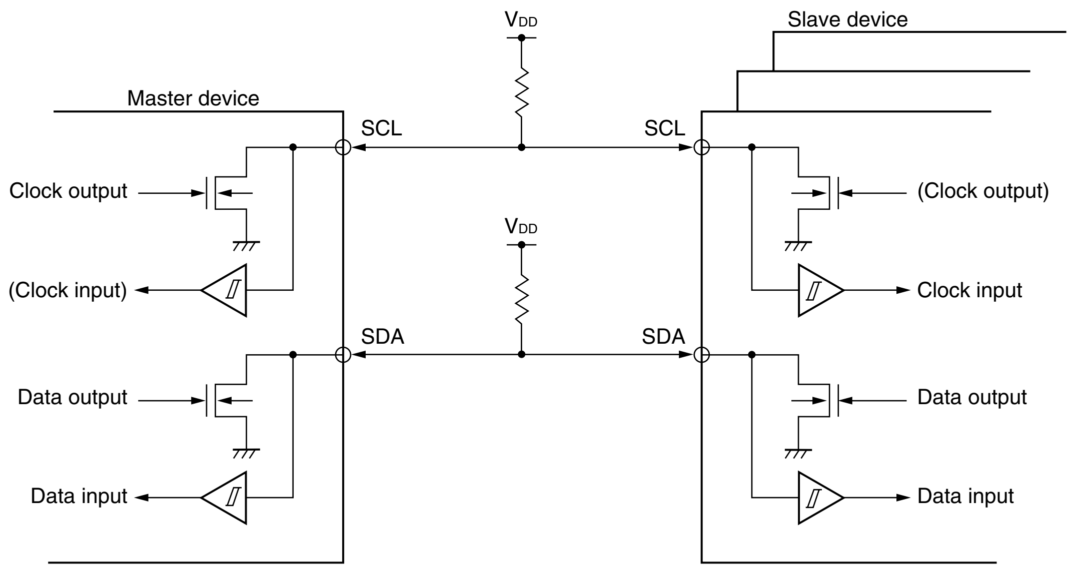
The serial clock pin (SCLn) and serial data bus pin (SDAn) are configured as follows (n = 0, 1).

SCLn .... 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.

SDAn .... 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.

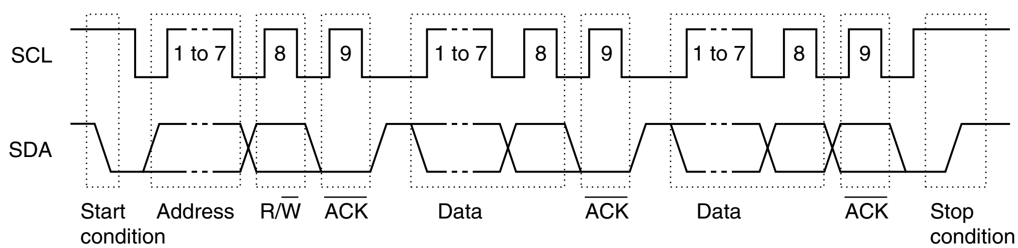
Figure 10-25. Pin Configuration Diagram



10.4.4 I²C bus definitions and control methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. The transfer timing for the "start condition", "data", and "stop condition" output via the I²C bus's serial data bus is shown below.

Figure 10-26. I²C Bus's Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

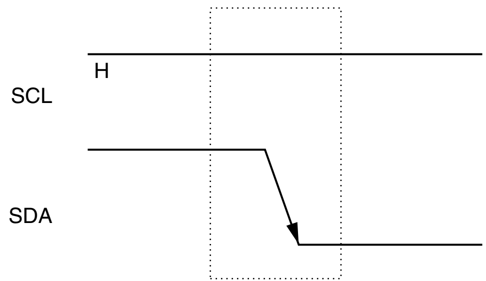
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLn) is continuously output by the master device. However, in the slave device, the SCLn's low-level period can be extended and a wait can be inserted (n = 0, 1).

(1) Start condition

A start condition is met when the SCLn pin is at high level and the SDAn pin changes from high level to low level. The start conditions for the SCLn pin and SDAn 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 (n = 0, 1).

Figure 10-27. Start Conditions



A start condition is output when IIC control register n (IICCn)'s bit 1 (STTn) is set to 1 after a stop condition has been detected (SPDn: Bit 0 = 1 in the IIC status register n (IICSn)). When a start condition is detected, IICSn's bit 1 (STDn) is set to 1 (n = 0, 1).

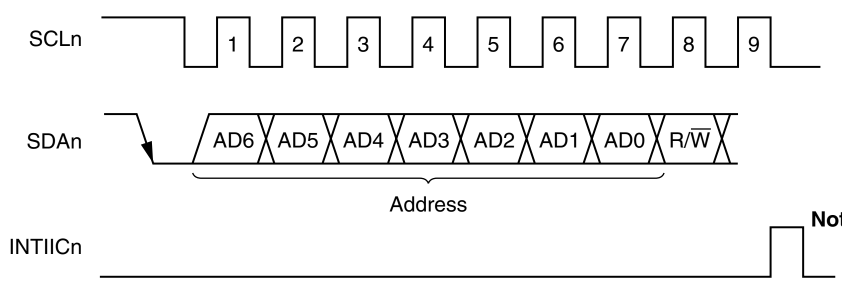
(2) Addresses

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 n (SVAn). If the address data matches the SVAn values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition (n = 0, 1).

Figure 10-28. Address



Note INTIICn is generated if a local address or extension code is received during slave device operation.

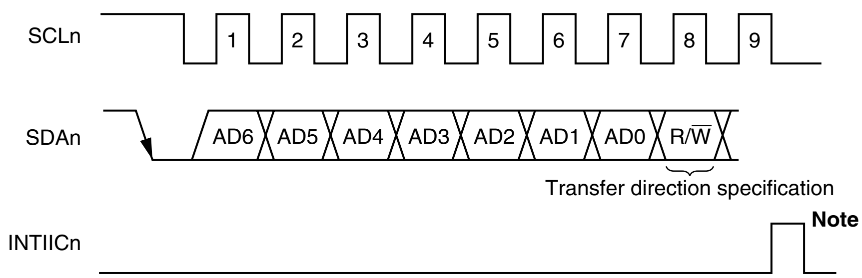
Remark n = 0, 1

The slave address and the eighth bit, which specifies the transfer direction as described in **(3) Transfer direction specification** below, are together written to the IIC shift register (IICn) and are then output. Received addresses are written to IICn (n = 0, 1).
The slave address is assigned to the higher 7 bits of IICn.

(3) Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 10-29. Transfer Direction Specification



Note INTIICn is generated if a local address or extension code is received during slave device operation.

Remark n = 0, 1

(4) Acknowledge signal ($\overline{\text{ACK}}$)

The acknowledge signal ($\overline{\text{ACK}}$) is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{\text{ACK}}$ signal for each 8 bits of data it receives. The transmitting device normally receives an $\overline{\text{ACK}}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{\text{ACK}}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{\text{ACK}}$ signal is returned after it transmits 8 bits of data. When an $\overline{\text{ACK}}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{\text{ACK}}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{\text{ACK}}$ signal may be caused by the following two factors.

(a) Reception was not performed normally.

(b) The final data was received.

When the receiving device sets the SDAn line to low level during the ninth clock, the $\overline{\text{ACK}}$ signal becomes active (normal receive response).

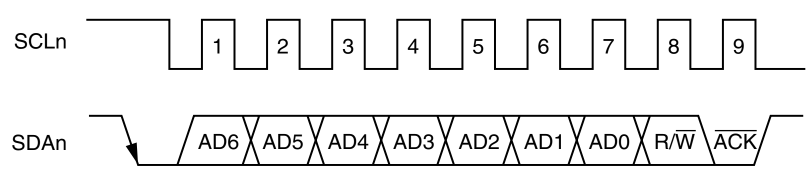
When bit 2 (ACKEn) of IIC control register n (IICCn) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled (n = 0, 1).

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRCn) of IIC status register n (IICSn) to be set. When this TRCn bit's value is 0, it indicates receive mode. Therefore, ACKEn should be set to 1 (n = 0, 1).

When the slave device is receiving (when TRCn = 0), if the slave device does not need to receive any more data after receiving several bytes, setting ACKEn to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRCn = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting ACKEn to 0 will prevent the $\overline{\text{ACK}}$ signal from being returned. This prevents the MSB data from being output via the SDAn line (i.e., stops transmission) during transmission from the slave device.

Figure 10-30. $\overline{\text{ACK}}$ Signal



Remark n = 0, 1

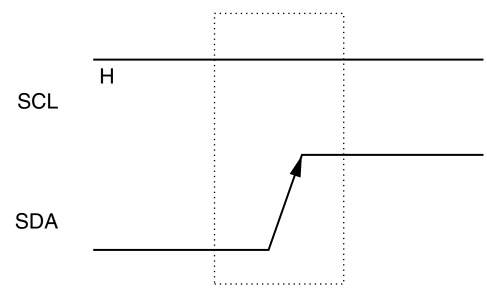
When the local address is received, an $\overline{ACK}$ signal is automatically output in synchronization with the falling edge of the SCLn's eighth clock regardless of the ACKEn value. No $\overline{ACK}$ signal is output if the received address is not a local address (n = 0, 1).
The $\overline{ACK}$ 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 SCLn's eighth clock if ACKEn 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 SCLn's eighth clock if ACKEn has already been set to 1.

(5) Stop condition

When the SCLn pin is at high level, changing the SDAn pin from low level to high level generates a stop condition (n = 0, 1).
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 10-31. Stop Condition



Remark n = 0, 1

A stop condition is generated when bit 0 (SPTn) of IIC control register n (IICCn) is set to 1. When the stop condition is detected, bit 0 (SPDn) of IIC status register n (IICSn) is set to 1 and INTIICn is generated when bit 4 (SPIEn) of IICCn is set to 1 (n = 0, 1).

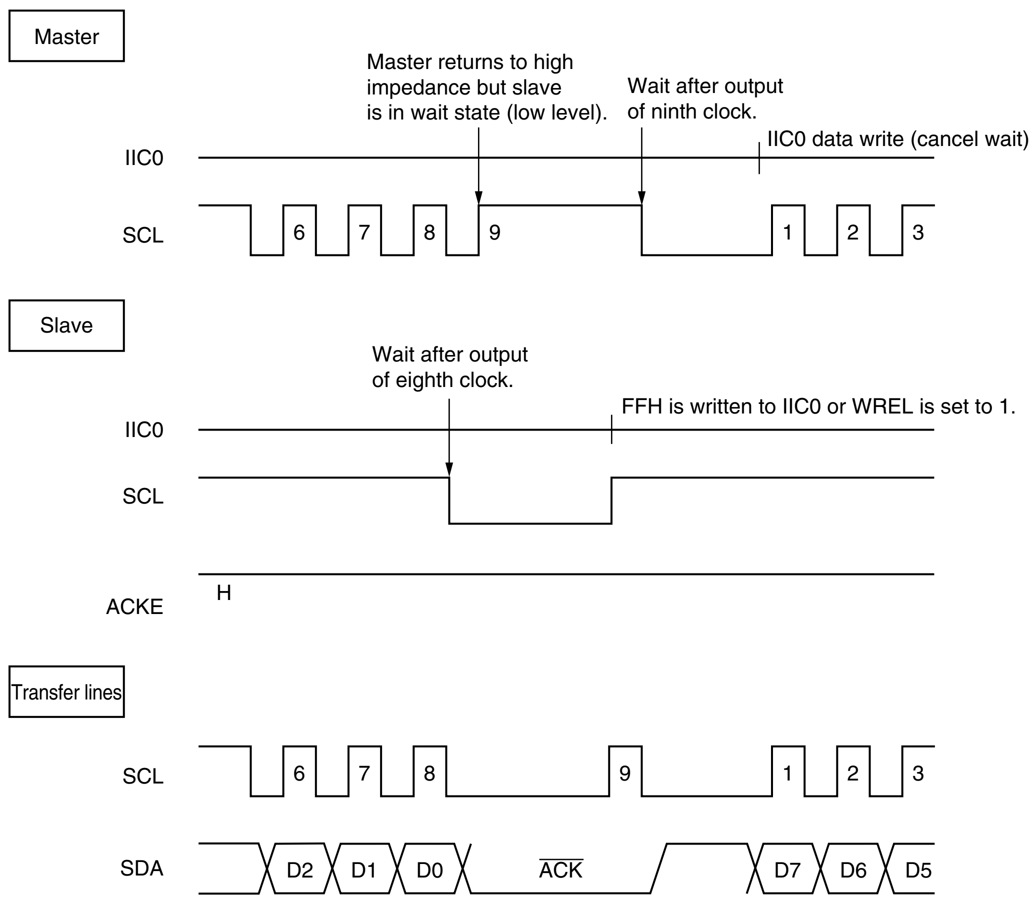
(6) Wait signal ($\overline{\text{WAIT}}$)

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 SCLn pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin (n = 0, 1).

Figure 10-32. 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 ACKEn = 1)

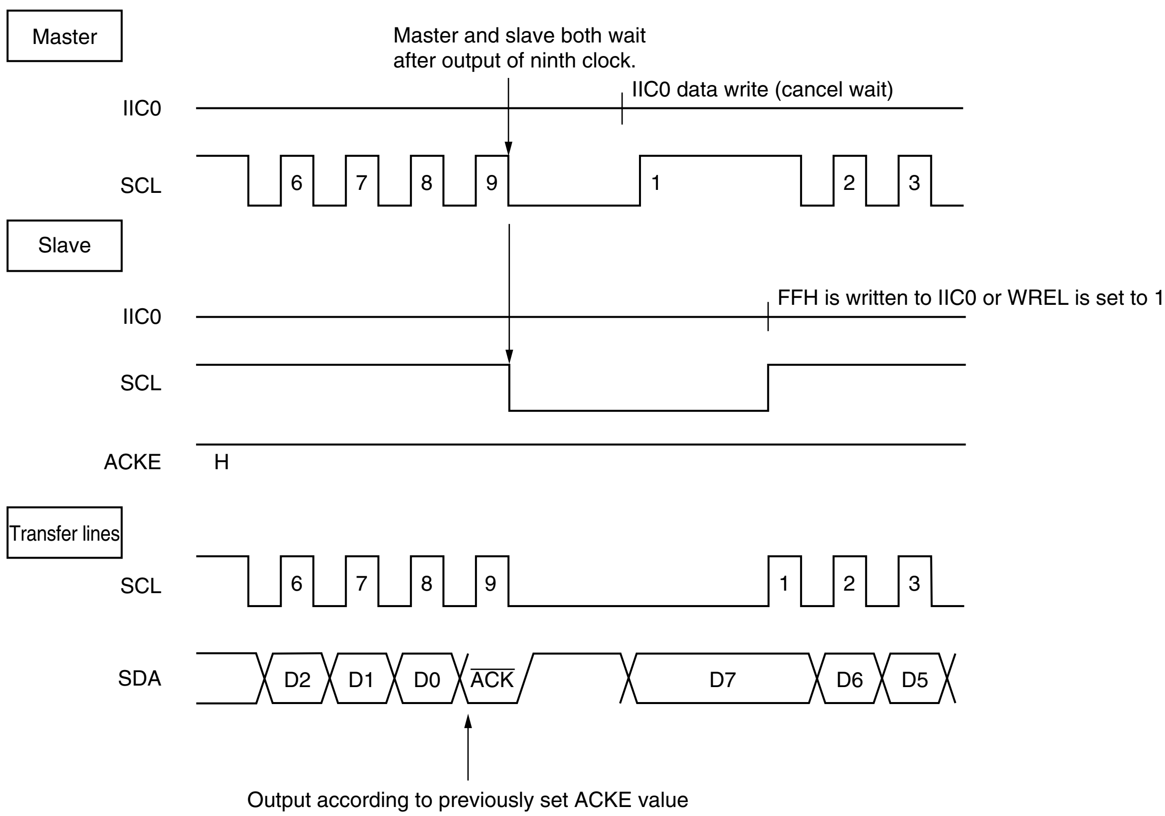


Remark n = 0, 1

Figure 10-32. Wait Signal (2/2)

(b) When master and slave devices both have a nine-clock wait

(master: transmission, slave: reception, and ACKEn = 1)



Remarks **1.** ACKEn: Bit 2 of IIC control register n (IICCn)
WRELn: Bit 5 of IIC control register n (IICCn)
2. n = 0, 1

A wait may be automatically generated depending on the setting for bit 3 (WTIMn) of IIC control register n (IICCn) (n = 0, 1).
Normally, when bit 5 (WRELn) of IICCn is set to 1 or when FFH is written to IIC shift register n (IICn), the wait status is canceled and the transmitting side writes data to IICn to cancel the wait status.
The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STTn) of IICCn to 1
- By setting bit 0 (SPTn) of IICCn to 1

10.4.5 I2C interrupt requests (INTIICn)

The following shows the value of IIC status register n (IICSn) at the INTIICn interrupt request generation timing and at the INTIICn interrupt timing (n = 0, 1).

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When WTIMn = 0

								SPTn = 1 ↓
ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	▲4 Δ5

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX000B

▲3: IICSn = 10XXX000B (WTIMn = 0)

▲4: IICSn = 10XXXX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

								SPTn = 1 ↓
ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2			▲3 Δ4

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX100B

▲3: IICSn = 10XXXX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIMn = 0

STTn = 1 ↓ SPTn = 1 ↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 ▲5 ▲6 Δ7

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX000B (WTIMn = 1)

▲3: IICSn = 10XXXX00B (WTIMn = 0)

▲4: IICSn = 10XXX110B (WTIMn = 0)

▲5: IICSn = 10XXX000B (WTIMn = 1)

▲6: IICSn = 10XXXX00B

Δ 7: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

STTn = 1 ↓ SPTn = 1 ↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXXX00B

▲3: IICSn = 10XXX110B

▲4: IICSn = 10XXXX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIMn = 0

SPTn = 1 ↓ (at AK/SP boundary)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 1010X110B

▲2: IICSn = 1010X000B

▲3: IICSn = 1010X000B (WTIMn = 1)

▲4: IICSn = 1010XX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

<2> When WTIMn = 1

SPTn = 1 ↓ (at AK/SP boundary)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 1010X110B

▲2: IICSn = 1010X100B

▲3: IICSn = 1010XX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

(2) Slave device operation (when receiving slave address data (match with SVAn))

(a) Start ~ Address ~ Data ~ Data ~ Stop

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 0001X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X100B

▲3: IICSn = 0001XX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 0001X110B

▲4: IICSn = 0001X000B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0001X110B

▲4: IICSn = 0001XX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2				▲3		▲4		Δ5

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X000B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2			▲3	▲4		▲5	Δ6

▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X110B

▲5: IICSn = 0010XX00B

Δ 6: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3			Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

▲3: IICSn = 00000X10B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1			▲2			▲3			Δ4

▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 00000X10B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(3) Slave device operation (when receiving extension code)

(a) Start ~ Code ~ Data ~ Data ~ Stop

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 0010X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010X100B

▲4: IICSn = 0010XX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1		▲2					▲3	▲4		∆5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 0001X110B

▲4: IICSn = 0001X000B

∆ 5: IICSn = 00000001B

Remark ▲: Always generated
∆: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

<2> When WTIMn = 1 (after restart, match with SVAn)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3				▲4		▲5	∆6

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0001X110B

▲5: IICSn = 0001XX00B

∆ 6: IICSn = 00000001B

Remark ▲: Always generated
∆: Generated only when SPIEn = 1
X: don’t care
n = 0, 1

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1		▲2				▲3		▲4		△5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X000B

△ 5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3			▲4	▲5		▲6	△7

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0010X010B

▲5: IICSn = 0010X110B

▲6: IICSn = 0010XX00B

△ 7: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
		▲1		▲2					▲3			Δ4

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

▲3: IICSn = 00000X10B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
		▲1	▲2		▲3				▲4			Δ5

▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 00000X10B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

Δ1

Δ 1: IICSn = 00000001B

Remark Δ: Generated only when SPIEn = 1
n = 0, 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0101X110B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0001X000B

▲3: IICSn = 0001X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 0101X110B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0001X100B

▲3: IICSn = 0001XX00B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(b) When arbitration loss occurs during transmission of extension code

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
			▲1		▲2		▲3		Δ4

▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0010X000B

▲3: IICSn = 0010X000B

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3		▲4	Δ5

▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0010X110B

▲3: IICSn = 0010X100B

▲4: IICSn = 0010XX00B

Δ 5: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

(a) When arbitration loss occurs during transmission of slave address data

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 (after AK following RW) Δ2 (after SP)

▲1: IICSn = 01000110B (Example: when ALDn is read during interrupt servicing)

Δ 2: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
n = 0, 1

(b) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 (after RW) Δ2 (after SP)

▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

IICCn's LRELn is set to 1 by software

Δ 2: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(c) When arbitration loss occurs during data transfer

<1> When WTIMn = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 Δ3

▲1: IICSn = 10001110B

▲2: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
n = 0, 1

<2> When WTIMn = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 Δ3

▲1: IICSn = 10001110B

▲2: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
n = 0, 1

(d) When loss occurs due to restart condition during data transfer

<1> Not extension code (Example: mismatches with SVAn)

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1					▲2			Δ3

▲1: IICSn = 1000X110B

▲2: IICSn = 01000110B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
Dn = D6 to D0
n = 0, 1

<2> Extension code

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1				▲2				Δ3

▲1: IICSn = 1000X110B

▲2: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

IICCn's LRELn is set to 1 by software

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
Dn = D6 to D0
n = 0, 1

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	RW	AK	D7 to Dn	SP

▲1 Δ2

▲1: IICSn = 1000X110B

Δ 2: IICSn = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
Dn = D6 to D0
n = 0, 1

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

When WTIMn = 1

STTn = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

▲3: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

When WTIMn = 1

STTn = 1 ↓ (at SP)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2	Δ3

▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

Δ 3: IICSn = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

When WTIMn = 1

SPTn = 1 ↓ (after second AK)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3			Δ4

▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

▲3: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

Δ 4: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: don't care
n = 0, 1

10.4.6 Interrupt request (INTIICn) generation timing and wait control

The setting of bit 3 (WTIMn) in IIC control register n (IICCn) determines the timing by which INTIICn is generated and the corresponding wait control, as shown below (n = 0, 1).

Table 10-10. INTIICn Generation Timing and Wait Control

WTIMn	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 INTIICn signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register n (SVAn).
At this point, $\overline{\text{ACK}}$ is output regardless of the value set to IICCn's bit 2 (ACKEn). For a slave device that has received an extension code, INTIICn occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of slave address register n (SVAn), neither INTIICn nor a wait occurs.

Remarks 1. The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

2. n = 0, 1

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined regardless of the WTIMn bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By setting bit 5 (WRELn) of IIC control register n (IICCn) to 1
- By writing to IIC shift register n (IICn)
- By start condition setting (bit 1 (STTn) of IIC control register n (IICCn) = 1)
- By step condition setting (bit 0 (SPTn) of IIC control register n (IICCn) = 1)

When an 8-clock wait has been selected (WTIMn = 0), the output level of $\overline{\text{ACK}}$ must be determined prior to wait cancellation.

Remark n = 0, 1

(5) Stop condition detection

INTIICn is generated when a stop condition is detected.

Remark n = 0, 1

10.4.7 Address match detection method

When 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 (INTIICn) occurs when a local address has been set to slave address register n (SVAn) and when the address set to SVAn matches the slave address sent by the master device, or when an extension code has been received (n = 0, 1).

10.4.8 Error detection

In I^2C bus mode, the status of the serial data bus (SDAn) during data transmission is captured by IIC shift register n (IICn) of the transmitting device, so the IICn data prior to transmission can be compared with the transmitted IICn data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match (n = 0, 1).

10.4.9 Extension code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXCn) is set for extension code reception and an interrupt request (INTIICn) is issued at the falling edge of the eighth clock (n = 0, 1).
The local address stored in slave address register n (SVAn) is not affected.

(2) If 11110xx0 is set to SVAn by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that INTIICn occurs at the falling edge of the eighth clock (n = 0, 1).

- Higher 4 bits of data match: EXCn = 1 [Note]
- 7 bits of data match: COIn = 1 [Note]

Note EXCn: Bit 5 of IIC status register n (IICSn)
COIn: Bit 4 of IIC status register n (IICSn)

(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 (LRELn) of IIC control register n (IICCn) to 1 and the CPU will enter the next communication wait state.

Table 10-11. 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 different bus format
1111 0xx	X	10-bit slave address specification

10.4.10 Arbitration

When several master devices simultaneously output a start condition (when STTn is set to 1 before STDn 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 (n = 0, 1).

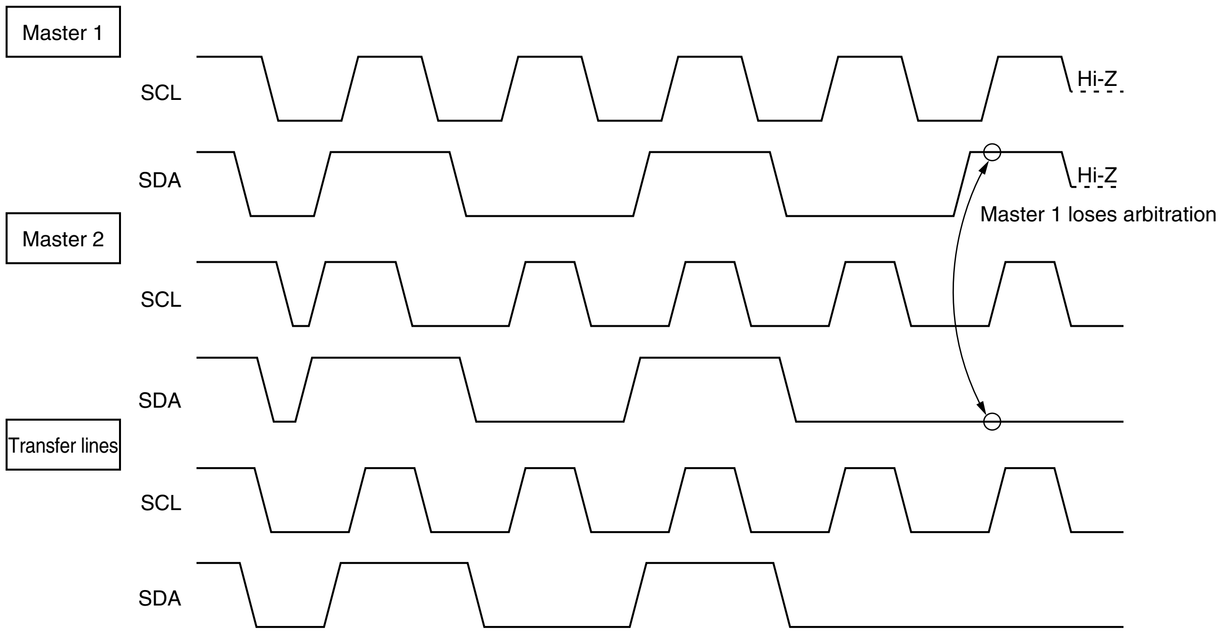
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in IIC status register n (IICSn) is set via the timing by which the arbitration loss occurred, and the SCLn and SDAn lines are both set for high impedance, which releases the bus (n = 0, 1).

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 ALDn = 1 setting that has been made by software (n = 0, 1).

For details of interrupt request timing, see **10.4.5 I^{2}C interrupt requests (INTIICn)**.

Note STDn: Bit 1 of IIC status register n (IICSn)
STTn: Bit 1 of IIC control register n (IICCn)

Figure 10-33. Arbitration Timing Example



Remark n = 0, 1

Table 10-12. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ signal transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIEn = 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 SPIEn = 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 SCLn is at low level while attempting to output a restart condition	

Notes 1. When WTIMn (bit 3 of the IIC control register n (IICCn)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock (n = 0, 1).

2. When there is a possibility that arbitration will occur, set SPIEn = 1 for master device operation (n = 0, 1).

Remark SPIEn: Bit 5 of IIC control register n (IICCn)

10.4.11 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request (INTIICn) 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 (SPIEn) of IIC control register n (IICCn) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled (n = 0, 1).

10.4.12 Communication reservation

(1) When communication reservation function is enabled (IICRSVn of IICFn register = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when LRELn of IIC control register n (IICCn) was set to “1”) (n = 0, 1).

If bit 1 (STTn) of IICCn is set while the bus is not used, a start condition is automatically generated and 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 n (IICn) causes the master’s address transfer to start. At this point, IICCn’s bit 4 (SPIEn) should be set (n = 0, 1).
When STTn has been set, the operation mode (as start condition or as communication reservation) is determined according to the bus status (n = 0, 1).

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 STTn, set STTn, wait for the wait period, then check the MSTSn (bit 7 of IIC status register n (IICSn)) (n = 0, 1).
Wait periods, which should be set via software, are listed in Table 10-13. These wait periods can be set via the settings for bits 3, 1, and 0 (SMCn, CLn1, and CLn0) in IIC clock selection register n (IICCLn) (n = 0, 1).

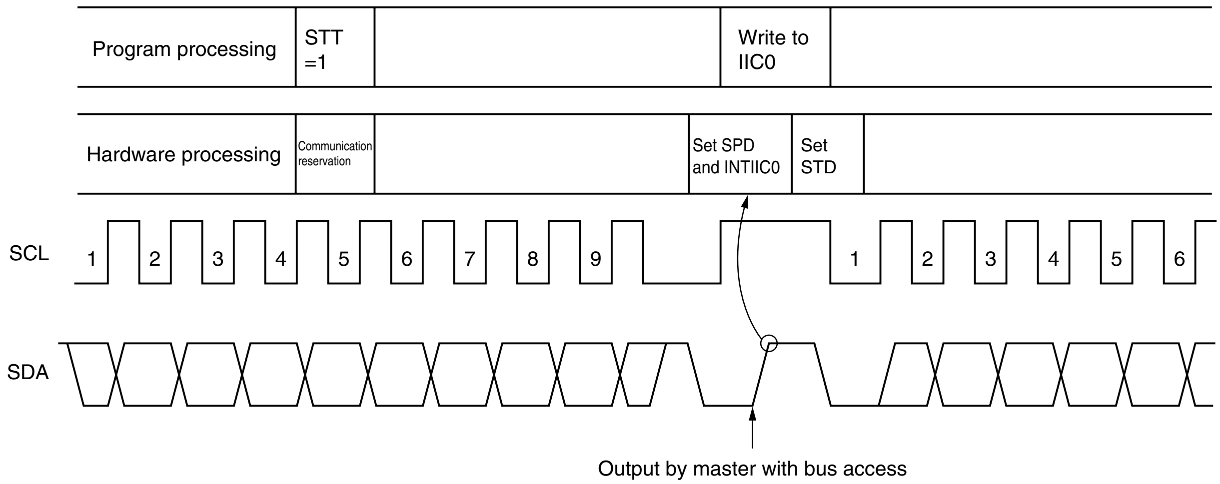
Table 10-13. Wait Periods

<table>
<tr><th>SMCn</th><th>CLn1</th><th>CLn0</th><th>Wait Period</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>26 clocks</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>46 clocks</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>92 clocks</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>37 clocks</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td rowspan="2">16 clocks</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>32 clocks</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>13 clocks</td></tr>
</table>

Remark n = 0, 1

The communication reservation timing is shown below.

Figure 10-34. Communication Reservation Timing

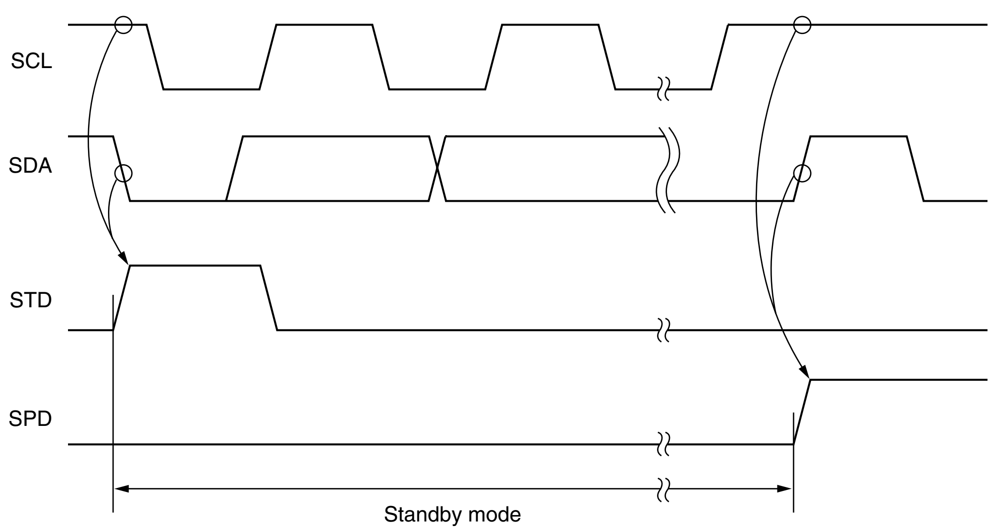


IICn: IIC shift register n
STTn: Bit 1 of IIC control register n (IICCn)
STDn: Bit 1 of IIC status register n (IICSn)
SPDn: Bit 0 of IIC status register n (IICSn)

Remark n = 0, 1

Communication reservations are accepted via the following timing. After bit 1 (STDn) of IIC status register n (IICSn) is set to "1", a communication reservation can be made by setting bit 1 (STTn) of IIC control register n (IICCn) to "1" before a stop condition is detected (n = 0, 1).

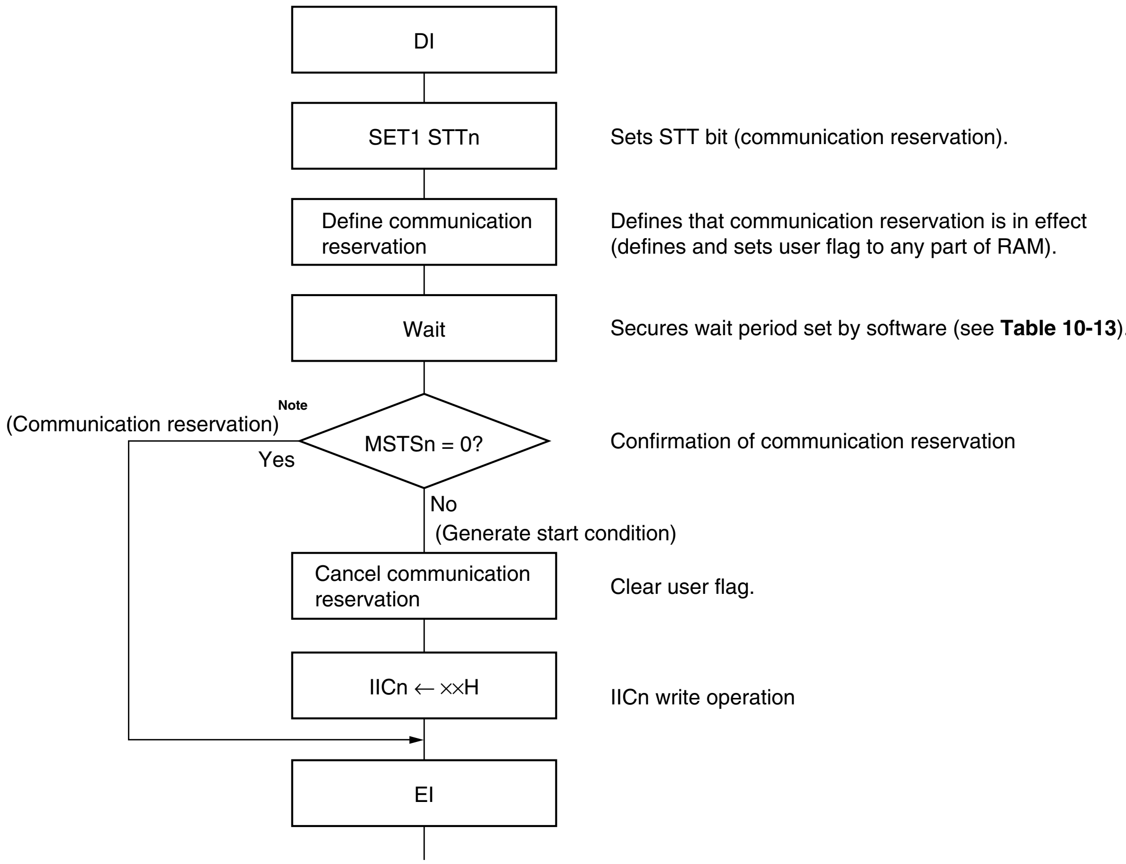
Figure 10-35. Timing for Accepting Communication Reservations



Remark n = 0, 1

The communication reservation flowchart is illustrated below.

Figure 10-36. Communication Reservation Flowchart



Note The communication reservation operation executes a write to IIC shift register n (IICn) when a stop condition interrupt request occurs.

Remark n = 0, 1

(2) When communication reservation function is disabled (IICRSVn of IICFn register = 1)

If the STTn bit of the IICn register is set when the bus is not participating in the current communication while bus communication is in progress, this request is rejected and a start condition is not generated. There are two modes under which the bus is not participating in communication.

- 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 LRELn of IIC control register n (IICCn) was set to “1”) (n = 0, 1).

To detect whether a start condition was generated or the request was rejected, check the STCFn flag of the IICFn register. Wait for the period shown in Table 10-14 using software since it takes the period to set the STCFn flag from when STTn = 1 is set.

Table 10-14. Wait Time

IICCEn1	IICCEn0	CLn1	CLn0	Wait Time
X	X	0	0	3 clocks
X	X	0	1	3 clocks
X	X	1	0	6 clocks
0	0	1	1	$3 \times N$
0	1	1	1	6 clocks
1	0	1	1	9 clocks

Remarks 1. N: TM2, TM3 output

X: Don’t care

2. n = 0, 1

Caution Do not set STTn = 1 if the slave status is entered on an address match or extension code reception (at the timing shown in Figure 10-37). If it is set, the communication reservation status is entered.

Figure 10-37. Timing at Which STTn = 1 Cannot Be Set

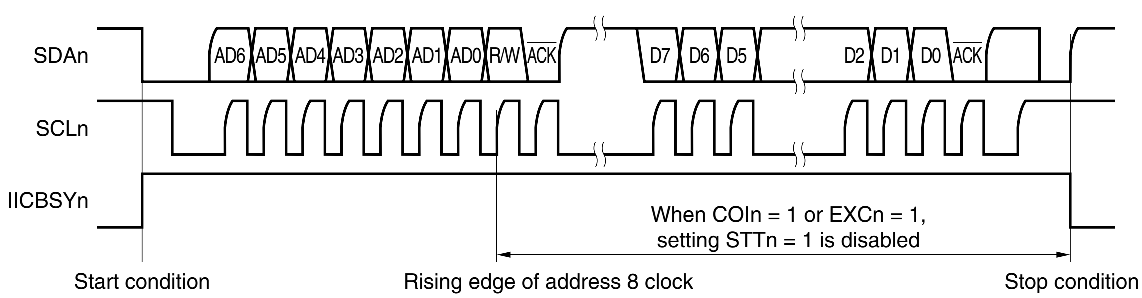
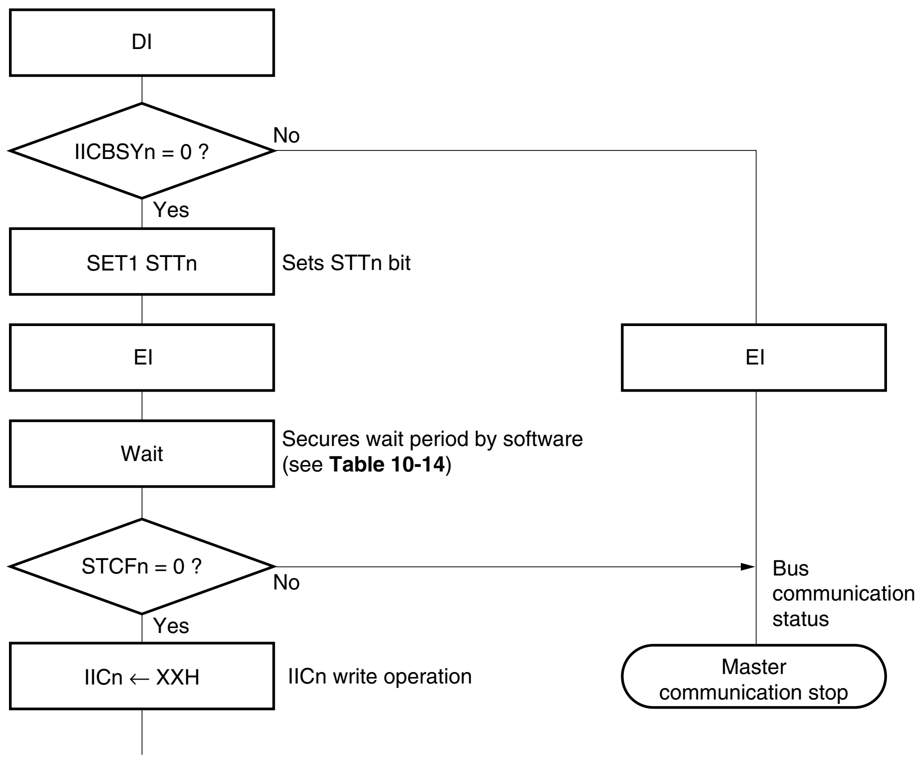


Figure 10-38. Master Communication Start or Stop Flowchart



Remark n = 0, 1

10.4.13 Cautions

(a) When STCENn of IICFn register = 0

Immediately after the I^2C operation is enabled, the communication status (IICBSYn of IICFn register = 1) is recognized regardless of the actual bus status. To perform master communication in the status in which the stop condition is not detected, first generate a stop condition to release the bus and then perform master communication.

Use the following sequence for generating a stop condition.

<1> Set IIC clock selection register n (IICCLn)
<2> Set the IICEn bit of IIC control register n (IICCn)
<3> Set the SPTn bit of the IICCn register

(b) When STCENn of IICFn register = 1

Immediately after the I^2C operation is enabled, the bus released status (IICBSYn = 0) is recognized regardless of the actual bus status. To issue the first start condition (STTn of IICCn register = 1), check that the bus is released in order to prevent disturbing other communications.

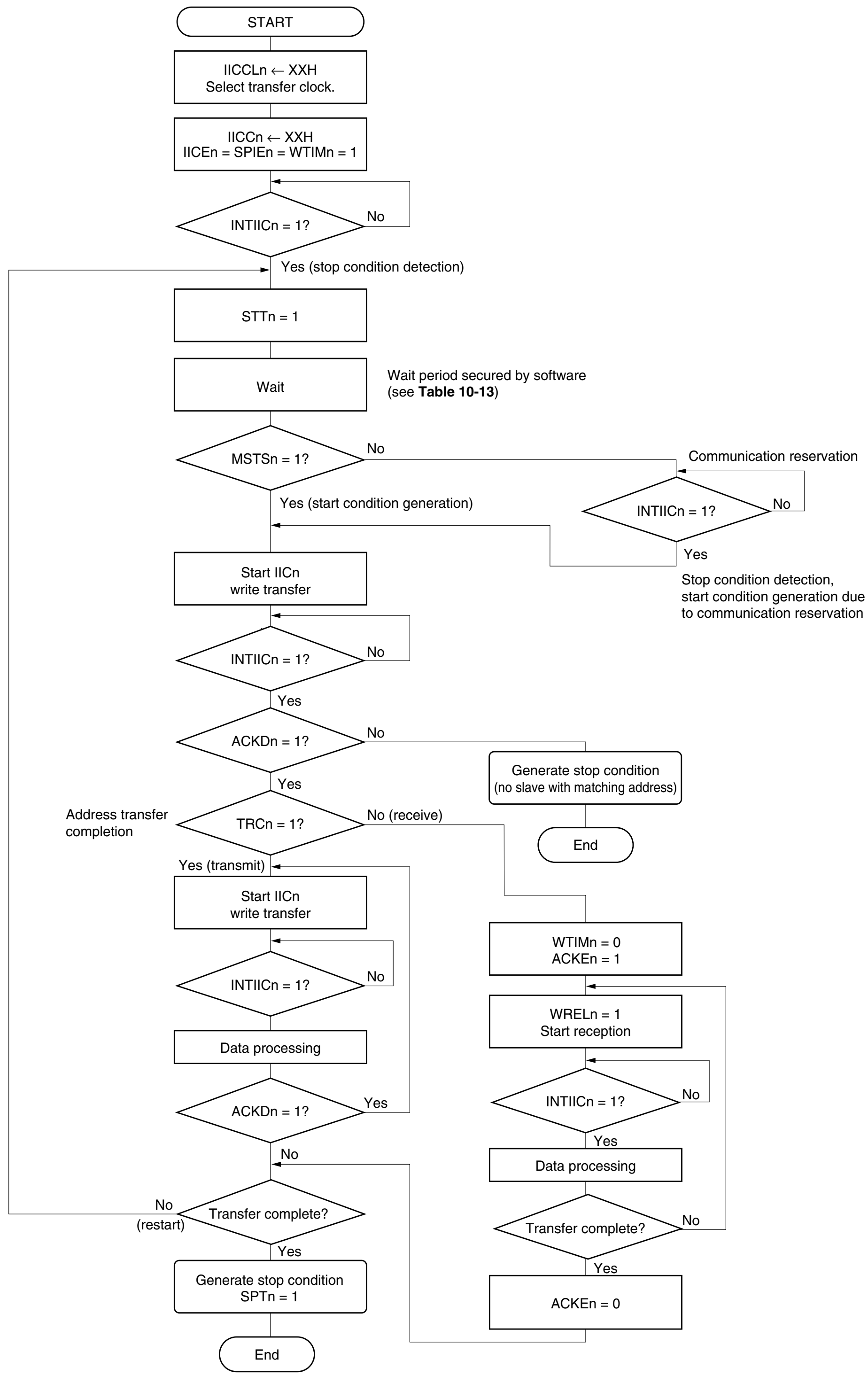
Remark n = 0, 1

10.4.14 Communication operations

(1) Master operations

The following shows an example of the master communication flowchart when the communication reservation function is enabled (IICRSVn = 0) and when communication is started after a stop condition is detected (STCENn = 0).

Figure 10-39. Master Operation Flowchart (1)

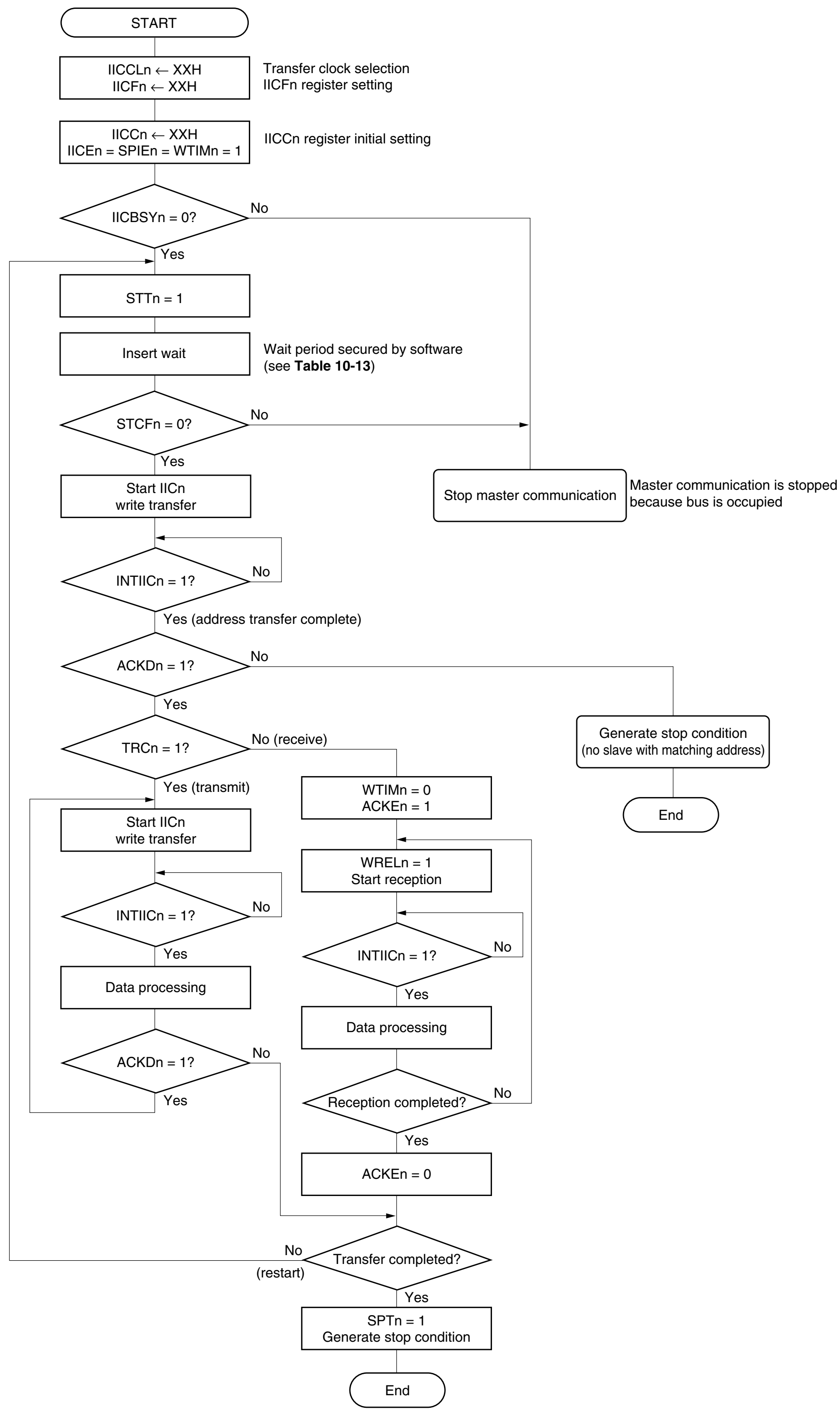


Remark n = 0, 1

(2) Master operation

The following shows an example of the master communication flowchart when the communication reservation function is disabled (IICRSVn = 1) and when communication is started without detecting a stop condition (STCENn = 1).

Figure 10-40. Master Operation Flowchart (2)

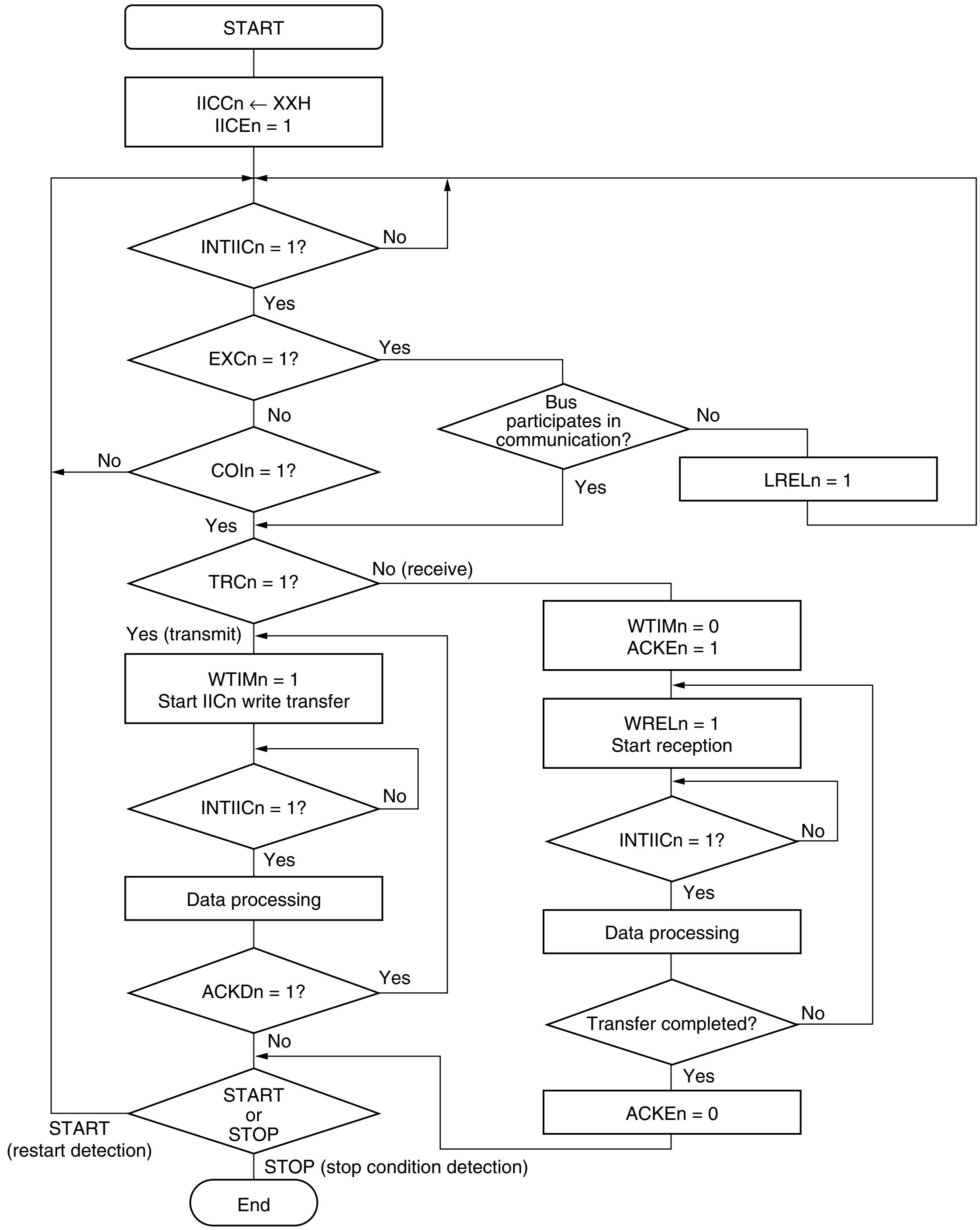


Remark n = 0, 1

(3) Slave operation

The following shows an example of the slave communication flowchart.

Figure 10-41. Slave Operation Flowchart



Remark n = 0, 1

10.4.15 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 TRCn bit (bit 3 of IIC status register n (IICSn)) that specifies the data transfer direction and then starts serial communication with the slave device.

IIC bus shift register n (IICn)'s shift operation is synchronized with the falling edge of the serial clock (SCLn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAn pin.

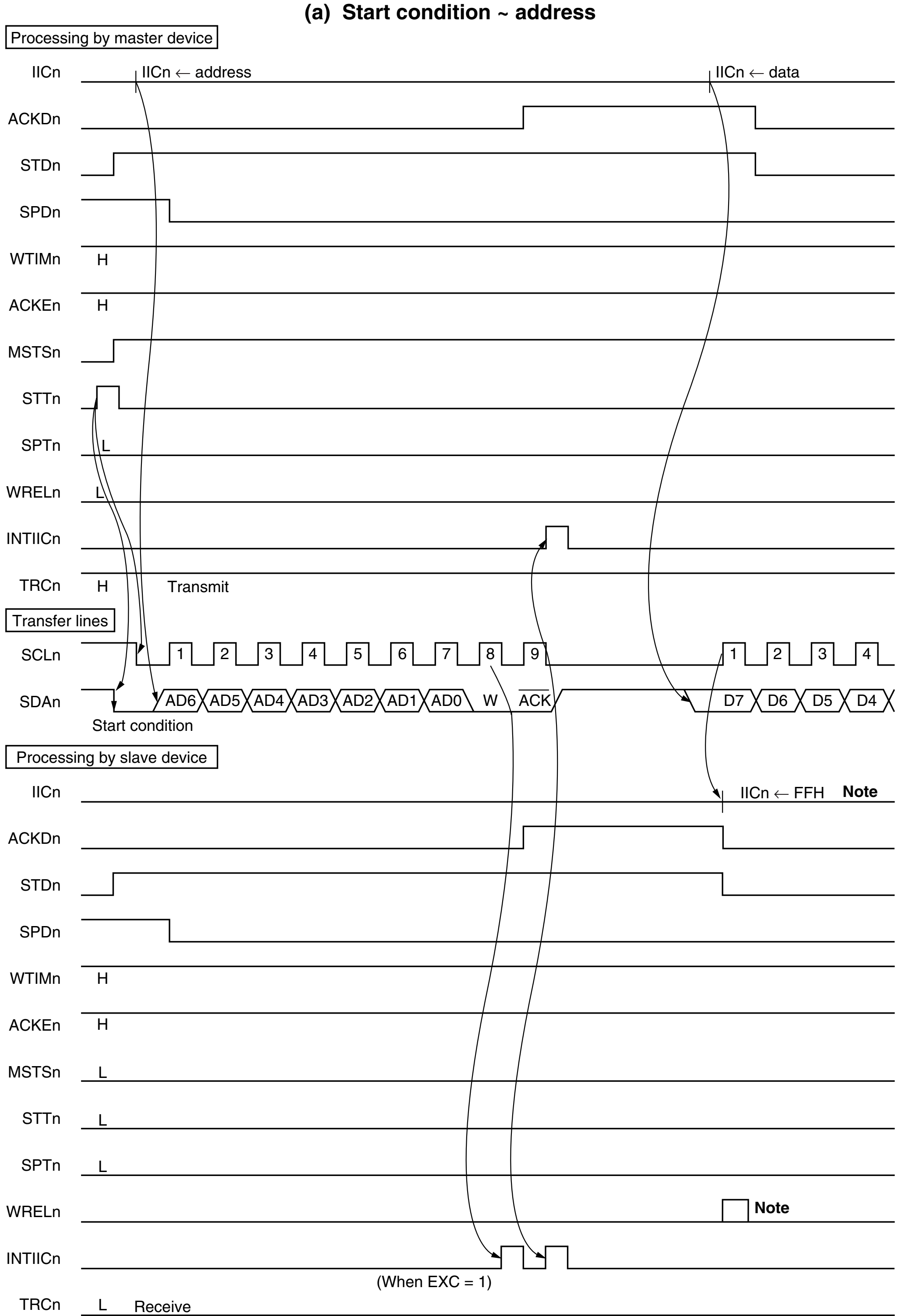
Data input via the SDAn pin is captured by IICn at the rising edge of SCLn.

The data communication timing is shown below.

Remark n = 0, 1

Figure 10-42. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address

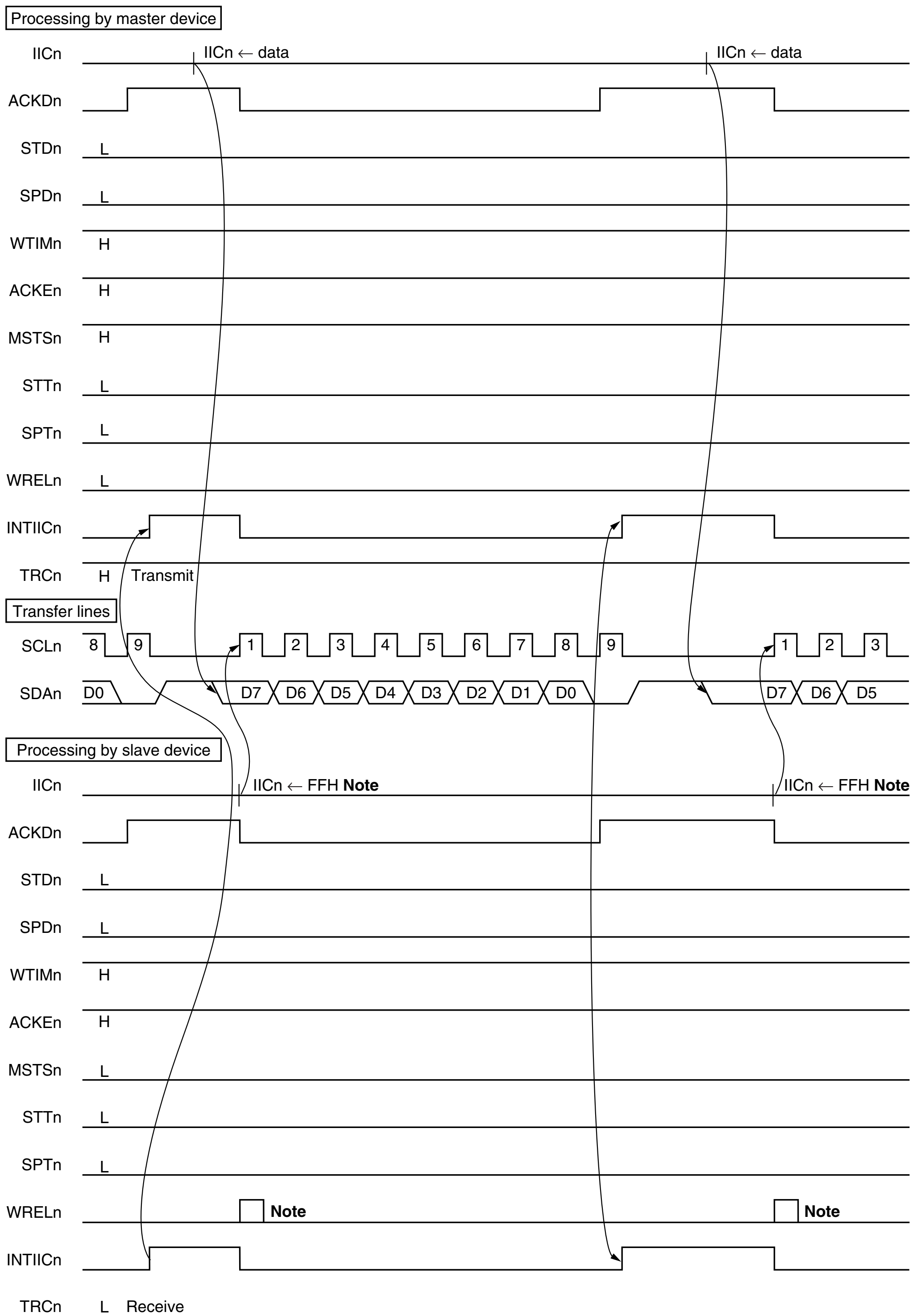


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-42. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

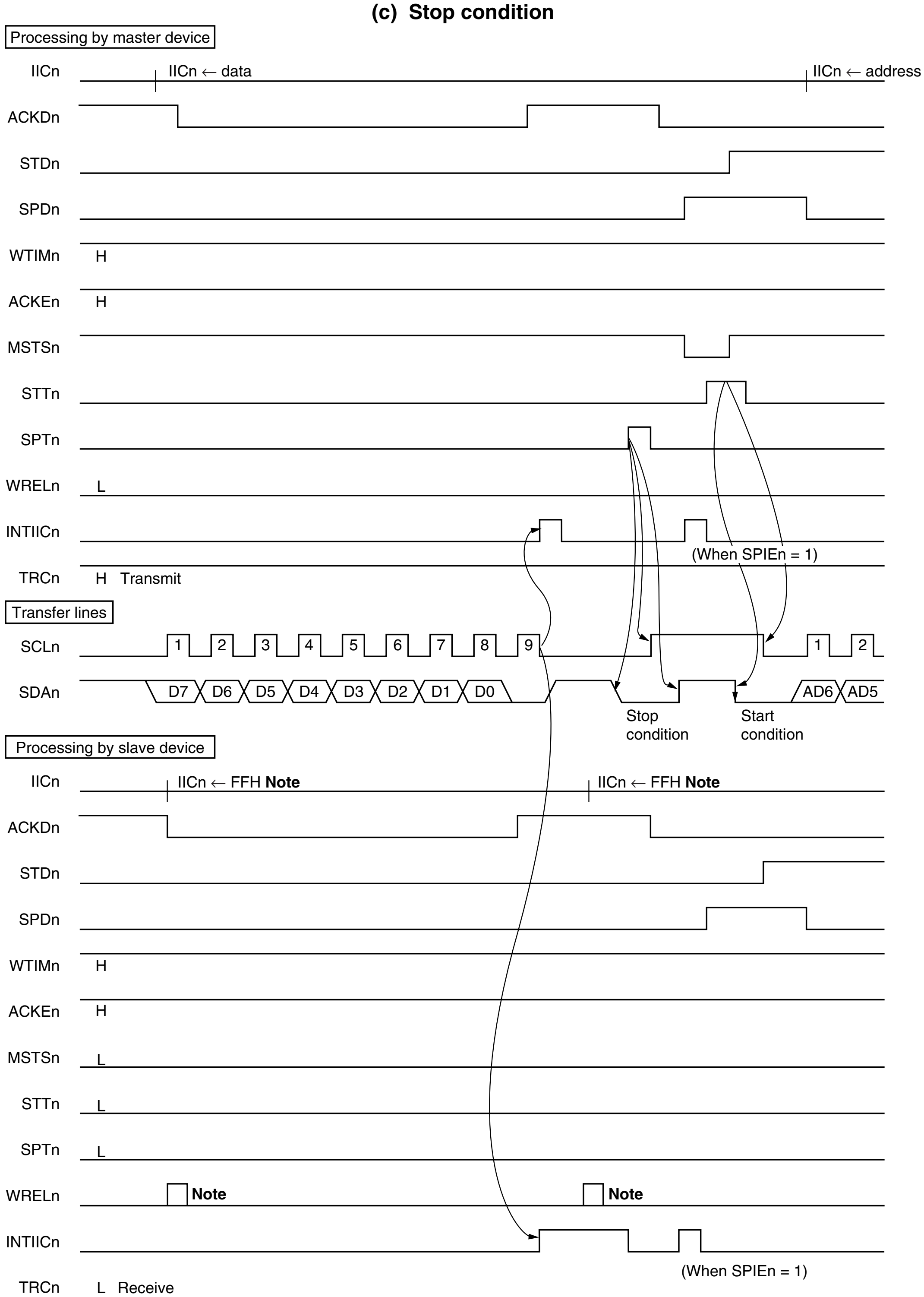


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-42. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition

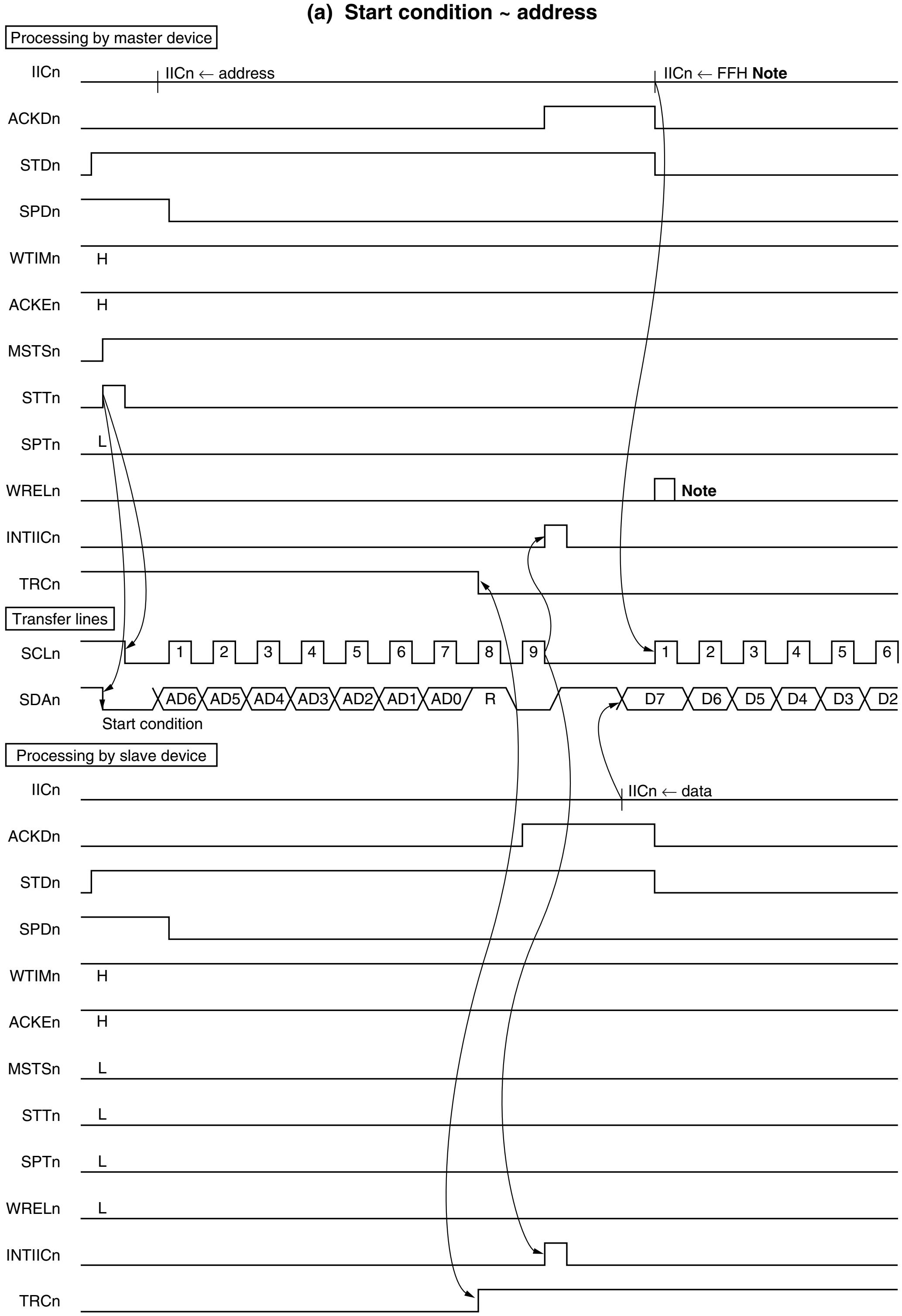


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-43. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address

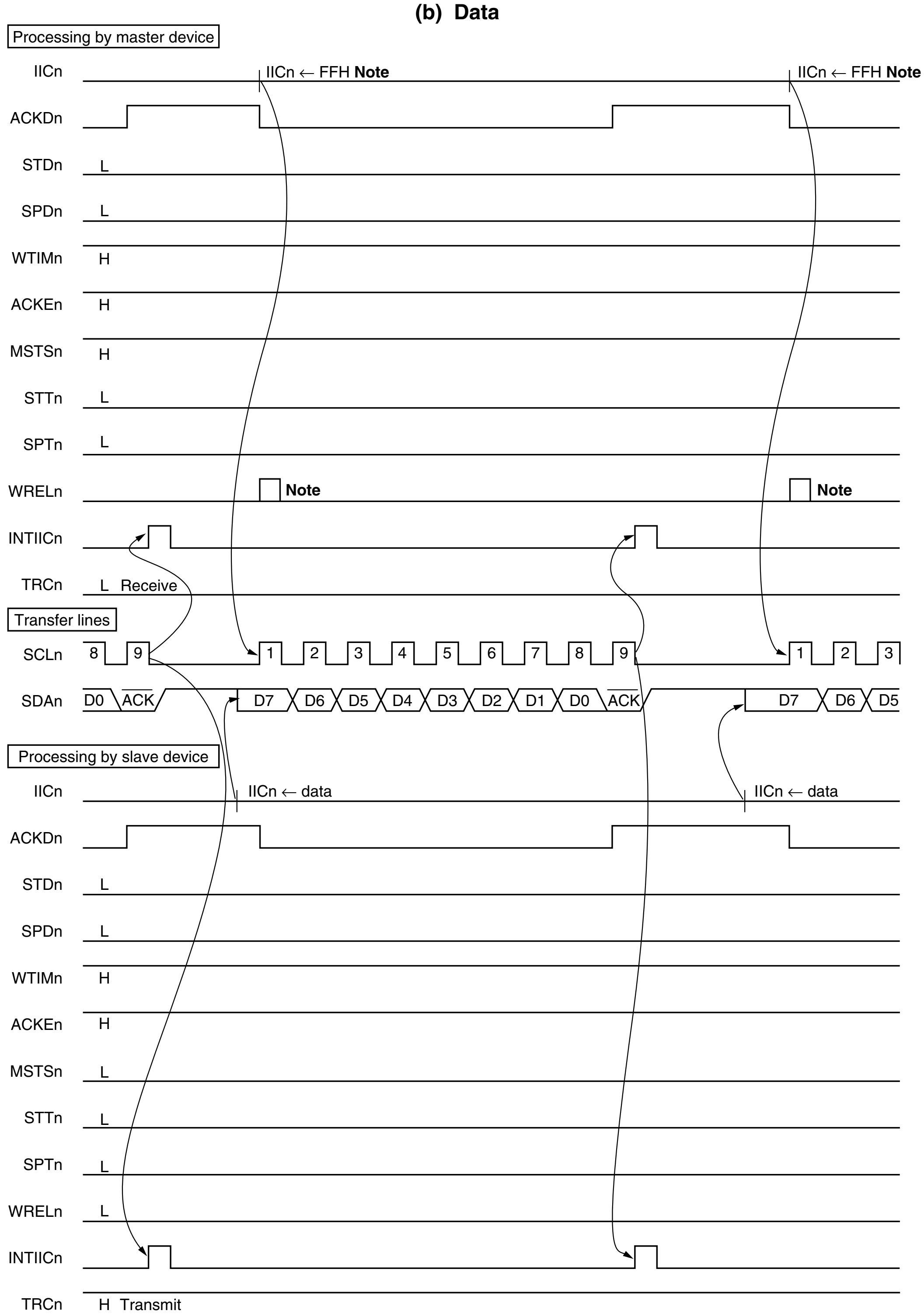


Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-43. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

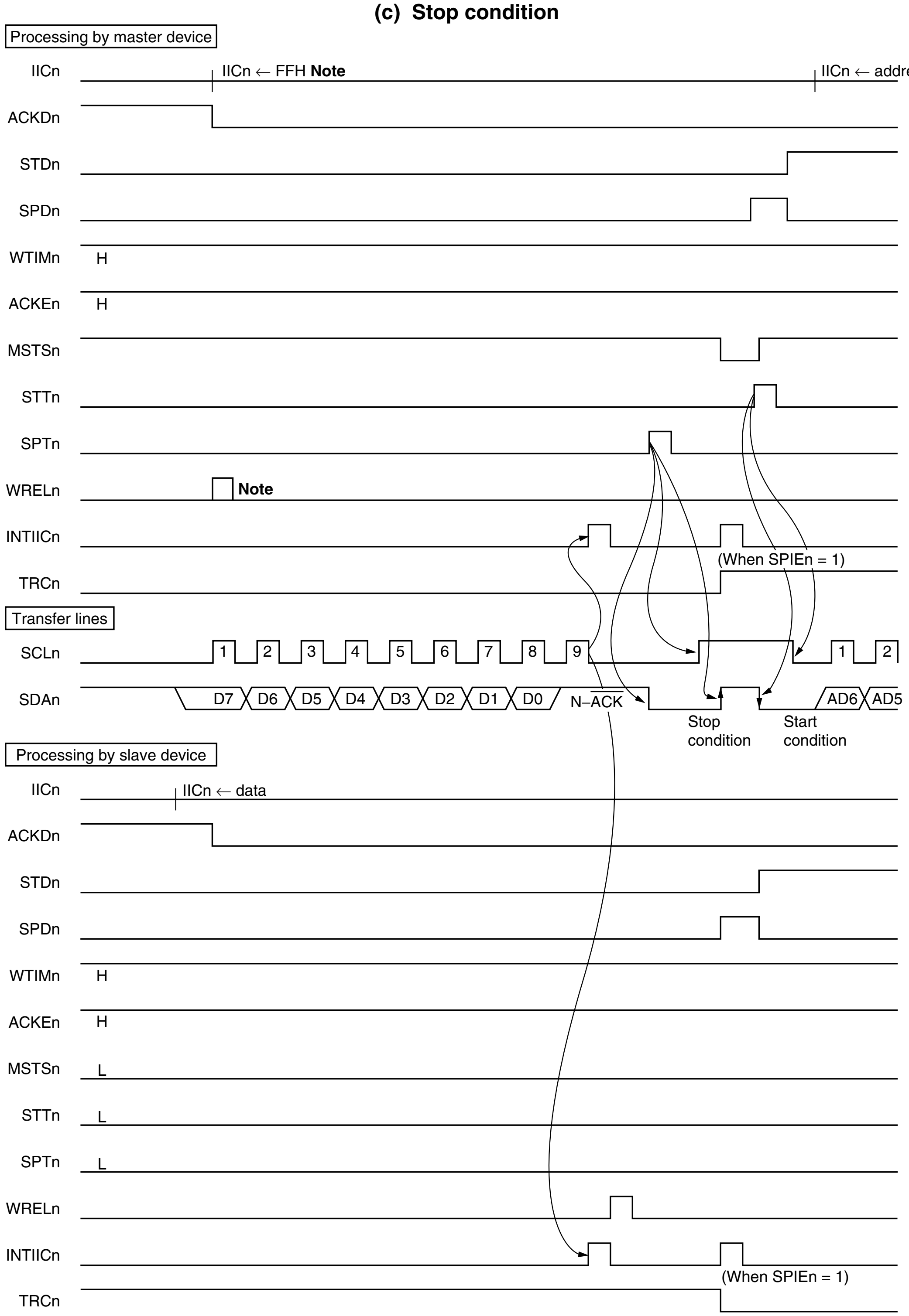


Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-43. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition



Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

10.5 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 mode

This mode enables full-duplex operation which transmits and receives one byte of data 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).

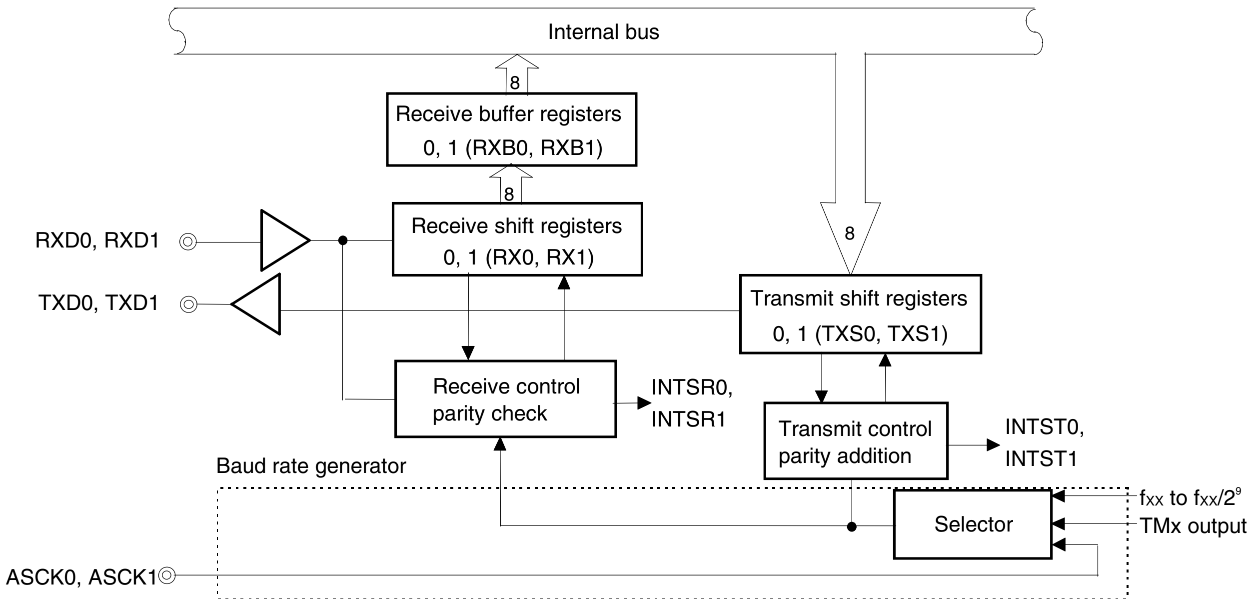
10.5.1 Configuration

The UARTn includes the following hardware.

Table 10-15. 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 00, 01 (BRGMC00, BRGMC01) Baud rate generator mode control registers 10, 11 (BRGMC10, BRGMC11)

Figure 10-44. Block Diagram of UARTn



Remark TMx output is as follows:
When UART0: TM2
When UART1: TM3

(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, 1(RXB0, RXB1).

RX0, 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.

10.5.2 UARTn control registers

UARTn uses the following registers for control function (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 n0, n1 (BRGMCn0, BRGMCn1)

(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.
RESET input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF300H, FFFFF310H

	<7>	<6>	5	4	3	2	1	0
ASIMn (n = 0, 1)	TXEn	RXEn	PS1n	PS0n	UCLn	SLn	ISRMn	0

TXEn	RXEn	Operation mode	RXDn/Pxx pin function	TXDn/Pxx pin function
0	0	Operation stop	Port function	Port function
0	1	UARTn mode (receive only)	Serial function	Port function
1	0	UARTn mode (transmit only)	Port function	Serial function
1	1	UARTn 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 asynchronous serial interface 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 clears these registers to 00H.

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 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 clears 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 n0, n1 (BRGMCn0, BRGMCn1)

These registers set the UARTn source clock.

BRGMCn0 and BRGMCn1 are set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

After reset: 00H R/W Address: FFFFF30EH, FFFFF31EH

	7	6	5	4	3	2	1	0
BRGMCn0 (n = 0, 1)	0	0	0	0	0	TPSn2	TPSn1	TPSn0

After reset: 00H R/W Address: FFFFF320H, FFFFF322H

	7	6	5	4	3	2	1	0
BRGMCn1 (n = 0, 1)	0	0	0	0	0	0	0	TPSn3

TPSn3	TPSn2	TPSn1	TPSn0	8-bit counter source clock selection	m
0	0	0	0	External clock (ASCKn)	–
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	at n = 0: TM3 output at n = 1: TM2 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 BRGMCn0, n1 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 BRGMCn0, n1 during communication processing.

★ **2. Be sure to set bits 7 to 3 of the BRGMCn0 to 0.**

Remarks 1. f_{SCK}: Source clock of 8-bit counter

2. When the output of the timer is selected as the clock, it is not necessary to set the P26/TO2/TI2 and P27/TO3/TI3 pins in the timer output mode.

10.5.3 Operations

UARTn has the following two operation modes.

- Operation stop mode
- Asynchronous serial interface mode

(1) Operation stop mode

In this mode serial transfers are not performed, allowing a reduction in power consumption.
When in operation stop mode, pins can be used as normal ports.

(a) Register settings

Operation stop mode settings are made via bits TxEn and RXEn of asynchronous serial interface mode register n (ASIMn).

Figure 10-45. ASIMn Setting (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 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

The asynchronous serial interface mode settings are made via ASIMn, BRGCn, BRGMCn0, and BRGMCn1 (n = 0, 1).

Figure 10-46. ASIMn Setting (Asynchronous Serial Interface 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	UARTn mode (receive only)	Serial function	Port function
1	0	UARTn mode (transmit only)	Port function	Serial function
1	1	UARTn 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.**

★ **3. Set RXEn to 1 while a high level is being input to the RXDn pin. If RXEn is set to 1 while the RXDn pin is low level, reception is started unexpectedly.**

Figure 10-47. ASISn Setting (Asynchronous Serial Interface 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 RXBn are read, further overrun errors will occur when receiving data.

Figure 10-48. BRGCn Setting (Asynchronous Serial Interface 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 clears the BRGCn register to 00H. Before starting operation, select a setting other than "Setting prohibited". Selecting "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 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 10-49. BRGMCn0 and BRGMCn1 Settings (Asynchronous Serial Interface Mode)

After reset: 00H R/W Address: FFFFF30EH, FFFFF31EH

	7	6	5	4	3	2	1	0
BRGMCn0 (n = 0, 1)	0	0	0	0	0	TPSn2	TPSn1	TPSn0

After reset: 00H R/W Address: FFFFF320H, FFFFF322H

	7	6	5	4	3	2	1	0
BRGMCn1 (n = 0, 1)	0	0	0	0	0	0	0	TPSn3

TPSn3	TPSn2	TPSn1	TPSn0	8-bit counter source clock selection	m
0	0	0	0	External clock (ASCKn)	–
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	at n = 0: TM3 output at n = 1: TM2 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 BRGMCn0, n1 during communication processing, the output of the baud rate generator is disturbed and communication will not be performed normally. Therefore, do not write to BRGMCn0 and BRGMCn1 during communication processing.

★ **2. Be sure to set bits 7 to 3 of the BRGMCn0 register to 0.**

Remarks 1. f_{XX}: Main clock oscillation frequency

2. When the output of the timer is selected as the clock, it is not necessary to set the P26/TO2/TI2 and P27/TO3/TI3 pins in the timer output mode.

(b) Baud rate

The baud rate transmit/receive clock that is generated is obtained 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 TPSn3 to TPSn0 ($0 \leq m \leq 9$)
  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 10-16 shows the relationship between the main clock and the baud rate, and Figure 10-50 shows an example of the baud rate error tolerance.

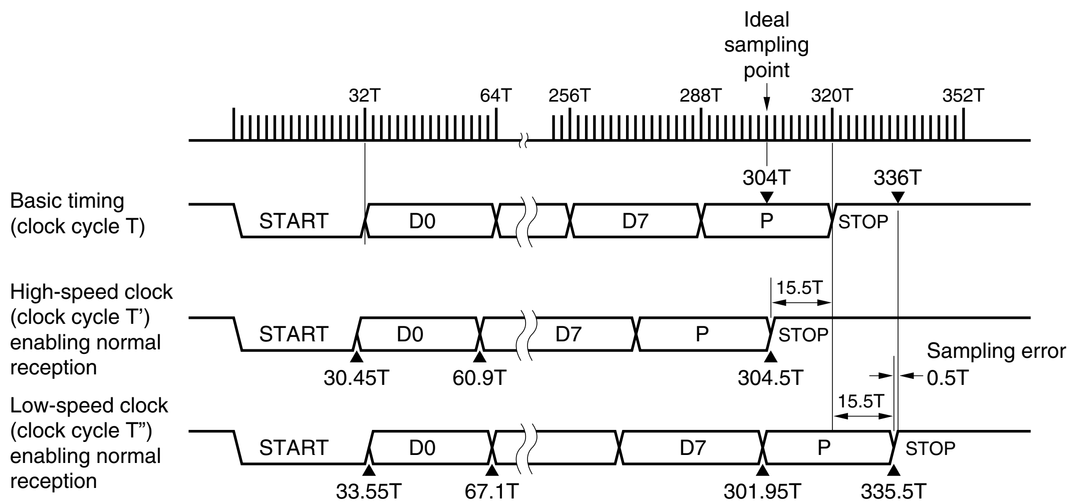
Table 10-16. Relationship Between Main Clock and Baud Rate

Baud Rate (bps)	f_{XX} = 8 MHz			f_{XX} = 12.58 MHz			f_{XX} = 16 MHz[Note 1]			f_{XX} = 20 MHz[Note 2]		
	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)
32	244	9	0.06	–	–	–	–	–	–	–	–	–
64	244	8	0.06	192	9	–0.02	244	9	0.06	–	–	–
128	244	7	0.06	192	8	–0.02	244	8	0.06	152	9	0.39
300	208	6	0.16	164	7	–0.12	208	7	0.16	130	8	0.16
600	208	5	0.16	164	6	–0.12	208	6	0.16	130	7	0.16
1200	208	4	0.16	164	5	–0.12	208	5	0.16	130	6	0.16
2400	208	3	0.16	164	4	–0.12	208	4	0.16	130	5	0.16
4800	208	2	0.16	164	3	–0.12	208	3	0.16	130	4	0.16
9600	208	1	0.16	164	2	–0.12	208	2	0.16	130	3	0.16
19200	208	0	0.16	164	1	–0.12	208	1	0.16	130	2	0.16
38400	104	0	0.16	164	0	–0.12	208	0	0.16	130	1	0.16
76800	52	0	0.16	82	0	–0.12	104	0	0.16	130	0	0.16
150000	27	0	–1.24	42	0	–0.16	53	0	0.63	67	0	–0.50
300000	13	0	2.56	21	0	–0.16	27	0	–1.24	33	0	1.01
312500	13	0	–1.54	20	0	0.64	26	0	–1.54	8	2	0.00

Notes 1. Only in the V850/SB1 and the H versions of the V850/SB2
2. Only in the V850/SB1

Remark f_{XX}: Main clock oscillation frequency

Figure 10-50. Error Tolerance (When k = 16), Including Sampling Errors



Remark T: 8-bit counter's source clock cycle

$$\text{Baud rate error tolerance (when k = 16)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

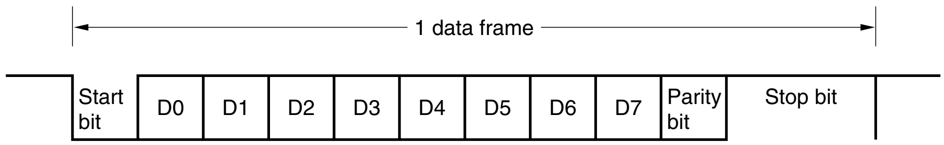
(3) Communication operations

(a) Data format

As shown in Figure 10-43, 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 10-51. 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).

(b) 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.

(i) 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.

(ii) 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.

(iii) 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".

(iv) 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.

(c) Transmission

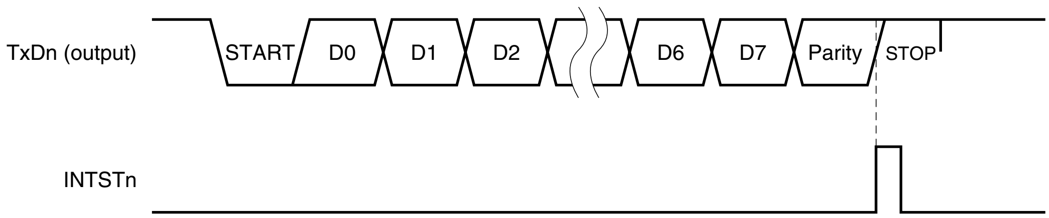
The transmit operation is started when transmit data is written to transmit shift register n (TXSn). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXSn, thereby emptying TXSn, after which a transmit completion interrupt (INTSTn) is issued.

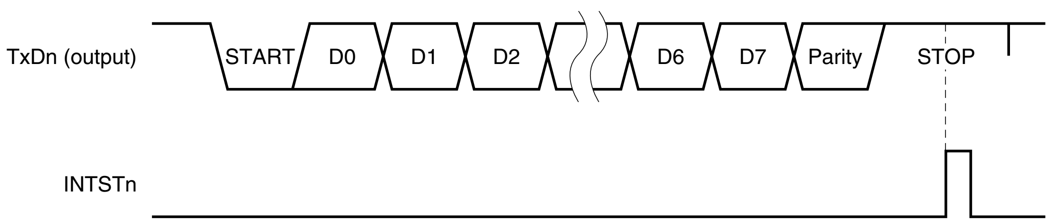
The timing of the transmit completion interrupt is shown below.

Figure 10-52. Timing of Asynchronous Serial Interface Transmit Completion Interrupt

(a) Stop bit length: 1



(b) Stop bit length: 2



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

(d) 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 BRGCn is used when sampling the RXDn pin.

When the RXDn pin goes low, the 8-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 8-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.

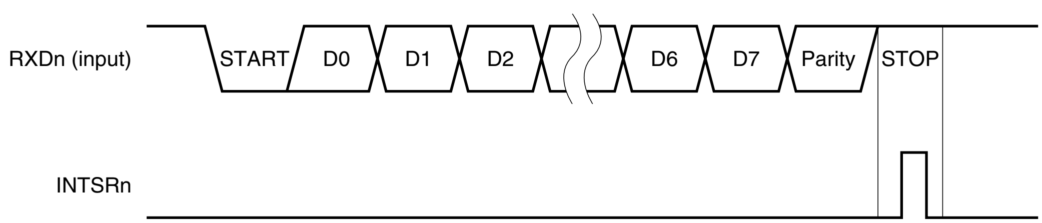
Even if an error has occurred, the receive data in which the error occurred is still transferred to RXBn.

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).

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 10-53. 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.

Remark n = 0, 1

(e) 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).

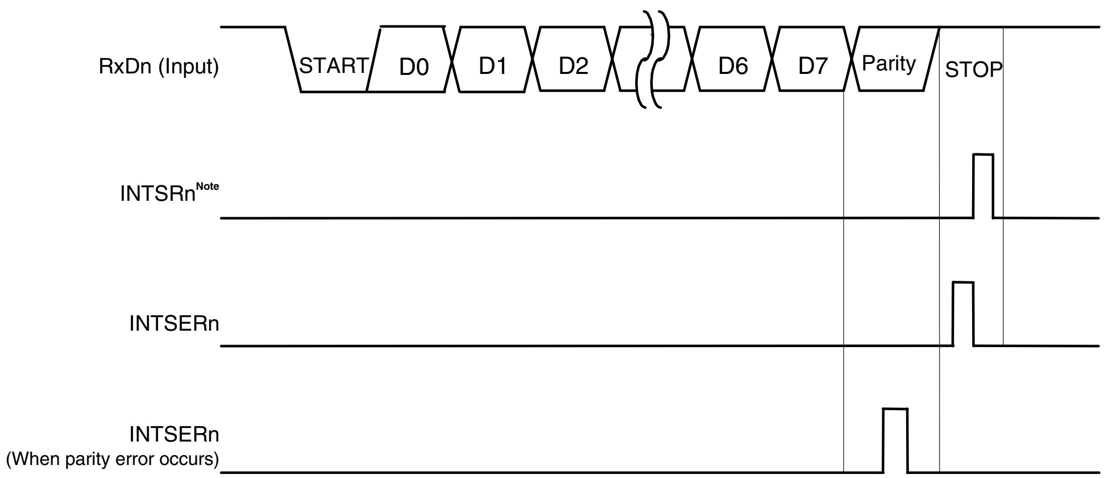
By reading the contents of ASISn during receive error interrupt servicing (INTSERn), it is possible to ascertain which error has occurred during reception.

The contents of ASISn are reset (0) by reading receive buffer register n (RXBn) or receiving subsequent data (if there is an error in the subsequent data, the error flag is set).

Table 10-17. 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 10-54. 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 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

10.5.4 Standby function

(1) Operation in HALT mode

Serial transfer operations are performed normally.

(2) Operation in STOP and IDLE modes

(a) When internal clock is selected as serial clock

The operations of asynchronous serial interface mode register n (ASIMn), 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 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

Serial transfer operations are performed normally.

10.6 3-Wire Variable-Length Serial I/O (CSI4)

CSI4 has the following two operation modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed.

(2) 3-wire variable-length serial I/O mode (MSB/LSB first switchable)

This mode transfers variable data of 8 to 16 bits via three lines: serial clock ($\overline{SCK4}$), serial output (SO4), and serial input (SI4).

Since the data can be transmitted and received simultaneously in the 3-wire variable-length serial I/O mode, the processing time of data transfer is shortened.

MSB and LSB can be switched for the first bit of data to be transferred in serial.

The 3-wire variable-length serial I/O mode is useful when connecting to a peripheral I/O device that includes a clocked serial interface, a display controller, etc.

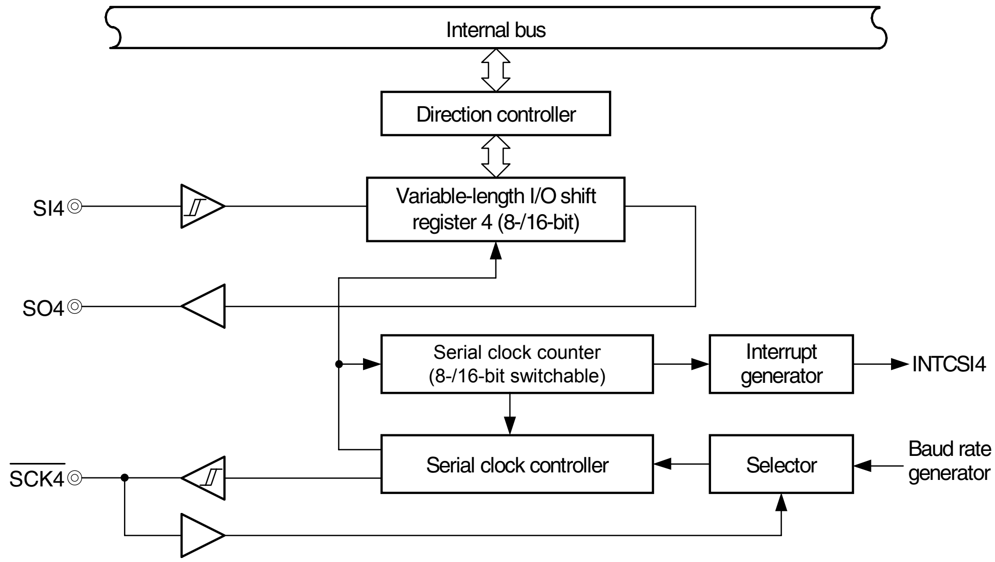
10.6.1 Configuration

CSI4 includes the following hardware.

Table 10-18. Configuration of CSI4

Item	Configuration
Register	Variable-length serial I/O shift register 4 (SIO4)
Control registers	Variable-length serial control register 4 (CSIM4) Variable-length serial setting register 4 (CSIB4) Baud rate generator source clock selection register 4 (BRGCN4) Baud rate generator output clock selection register 4 (BRGCK4)

Figure 10-55. Block Diagram of CSI4



(1) Variable-length serial I/O shift register 4 (SIO4)

SIO4 is a 16-bit variable register that performs parallel-serial conversion and transmit/receive (shift operations) synchronized with the serial clock.

SIO4 is set by a 16-bit memory manipulation instruction.

The serial operation starts when data is written to or read from SIO4, while the bit 7 (CSIE4) of variable-length serial control register 4 (CSIM4) is 1.

When transmitting, data written to SIO4 is output via the serial output (SO4).

When receiving, data is read from the serial input (SI4) and written to SIO4.

RESET input clears SIO4 to 0000H.

Caution Do not execute SIO4 access except for the access that becomes the transfer start trigger during transfer operations (read is disabled when MODE4 = 0 and write is disabled when MODE4 = 1).

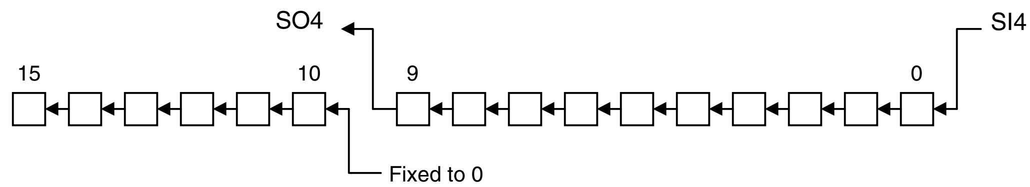
After reset: 0000H R/W Address: FFFFF2E0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIO4																

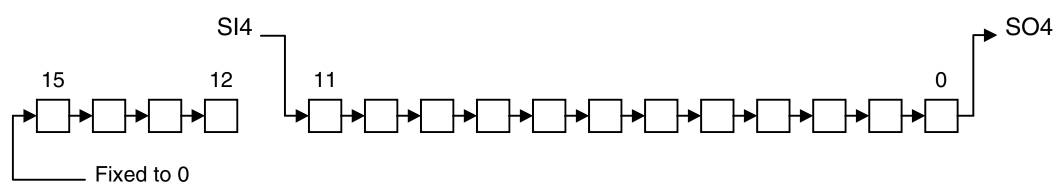
When the transfer bit length is set to other than 16 bits and data is set to the shift register, data should be aligned from the lowest bit of the shift register, regardless of whether MSB or LSB is set for the first transfer bit. Any data can be set to the unused higher bits; however, in this case the received data after a serial transfer operation becomes 0.

Figure 10-56. When Transfer Bit Length Other Than 16 Bits Is Set

(a) When transfer bit length is 10 bits and MSB first



(b) When transfer bit length is 12 bits and LSB first



10.6.2 CSI4 control registers

CSI4 uses the following registers for control functions.

- Variable-length serial control register 4 (CSIM4)
- Variable-length serial setting register 4 (CSIB4)
- Baud rate generator source clock selection register 4 (BRGCN4)
- Baud rate generator output clock selection register 4 (BRGCK4)

(1) Variable-length serial control register 4 (CSIM4)

This register is used to enable or disable serial interface channel 4's serial clock, operation modes, and specific operations.

CSIM4 can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIM4 to 00H.

After reset: 00H R/W Address: FFFFF2E2H

	<7>	6	5	4	3	2	1	0
CSIM4	CSIE4	0	0	0	0	MODE4	0	SCL4

CSIE4	SIO4 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function[Note 1]
1	Operation enabled	Count operation enabled	Serial function + port function[Note 2]

MODE4	Transfer operation mode flag		
	Operation mode	Transfer start trigger	SO4 output
0	Transmit/receive mode	SIO4 write	Normal output
1	Receive-only mode	SIO4 read	Port function

SCL4	Clock selection
0	External clock input ($\overline{\text{SCK4}}$)
1	BRG (Baud rate generator)

Notes 1. When CSIE4 = 0 (SIO4 operation disabled status), the port function is available for the SI4, SO4, and $\overline{\text{SCK4}}$ pins.

2. When CSIE4 = 1 (SIO4 operation enable status), the port function is available only for the SI4 pin when using the transmit function only and to SO4 pin when using the receive function.

(2) Variable-length serial setting register 4 (CSIB4)

CSIB4 is used to set the operation format of serial interface channel 4.

The bit length of a variable register is set by setting bits 3 to 0 (BSEL3 to BSEL0) of variable-length serial setting register 4. Data is transferred MSB first while bit 4 (DIR) is 1, and is transferred LSB first while DIR is 0.

CSIB4 can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIB4 to 00H.

After reset: 00H R/W Address: FFFFF2E4H

	7	<6>	<5>	<4>	3	2	1	0
CSIB4	0	CMODE	DMODE	DIR	BSEL3	BSEL2	BSEL1	BSEL0

CMODE	DMODE	$\overline{\text{SCK4}}$ active level	SI4 interrupt timing	SO4 output timing
0	0	Low level	Rising edge of $\overline{\text{SCK4}}$	Falling edge of $\overline{\text{SCK4}}$
0	1	Low level	Falling edge of $\overline{\text{SCK4}}$	Rising edge of $\overline{\text{SCK4}}$
1	0	High level	Falling edge of $\overline{\text{SCK4}}$	Rising edge of $\overline{\text{SCK4}}$
1	1	High level	Rising edge of $\overline{\text{SCK4}}$	Falling edge of $\overline{\text{SCK4}}$

DIR	Serial transfer direction
0	LSB first
1	MSB first

BSEL3	BSEL2	BSEL1	BSEL0	Bit length of serial register
0	0	0	0	16 bits
1	0	0	0	8 bits
1	0	0	1	9 bits
1	0	1	0	10 bits
1	0	1	1	11 bits
1	1	0	0	12 bits
1	1	0	1	13 bits
1	1	1	0	14 bits
1	1	1	1	15 bits
Other than above				Setting prohibited

(3) Baud rate generator source clock selection register 4 (BRGCN4)

BRGCN4 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears BRGCN4 to 00H.

After reset: 00H R/W Address: FFFFF2E6H

	7	6	5	4	3	2	1	0
BRGCN4	0	0	0	0	0	BRGN2	BRGN1	BRGN0

BRGN2	BRGN1	BRGN0	Source clock (f_{SCK})	n
0	0	0	f_{XX}	0
0	0	1	$f_{XX}/2$	1
0	1	0	$f_{XX}/4$	2
0	1	1	$f_{XX}/8$	3
1	0	0	$f_{XX}/16$	4
1	0	1	$f_{XX}/32$	5
1	1	0	$f_{XX}/64$	6
1	1	1	$f_{XX}/128$	7

(4) Baud rate generator output clock selection register 4 (BRGCK4)

BRGCK4 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGCK4 to 7FH.

After reset: 7FH R/W Address: FFFFF2E8H

	7	6	5	4	3	2	1	0
BRGCK4	0	BRGK6	BRGK5	BRGK4	BRGK3	BRGK2	BRGK1	BRGK0

BRGK6	BRGK5	BRGK4	BRGK3	BRGK2	BRGK1	BRGK0	Baud rate output clock	k
0	0	0	0	0	0	0	Setting prohibited	0
0	0	0	0	0	0	1	$f_{SCK}/2$	1
0	0	0	0	0	1	0	$f_{SCK}/4$	2
0	1	0	3	0	1	1	$f_{SCK}/6$	3
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{SCK}/252$	126
1	1	1	1	1	1	1	$f_{SCK}/254$	127

The baud rate transmit/receive clock that is generated is obtained 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 1 ≤ k ≤ 127>

$$[\text{Baud rate}] = \frac{f_{xx}}{2^n \times k \times 2} \ [\text{Hz}]$$

f_{XX}: Main clock oscillation frequency

n: Value set by BRGN2 to BRGN0 ($0 \leq n \leq 7$)

k: Value set by BRGK7 to BRGK0 ($1 \leq k \leq 127$)

Caution Do not use the baud rate transmit/receive clock of CSI4 with the transfer rate higher than the CPU operation clock. If used with the transfer rate higher than the CPU operation clock, transfer cannot be performed correctly.

10.6.3 Operations

CSI4 has the following two operation modes.

- Operation stop mode
- 3-wire variable-length serial I/O mode

(1) Operation stop mode

In this mode serial transfers are not performed and therefore power consumption can be reduced.

When in operation stop mode, SI4, SO4, and $\overline{\text{SCK4}}$ can be used as normal I/O ports.

(a) Register settings

Operation stop mode is set via CSIE4 bit of variable-length serial control register 4 (CSIM4). While CSIE4 = 0 (SIO4 operation stop state), the pins connected to SI4, SO4, or $\overline{\text{SCK4}}$ function as port pins.

Figure 10-57. CSIM4 Setting (Operation Stop Mode)

After reset: 00H R/W Address: FFFFF2E2H

	7	6	5	4	3	2	1	0
CSIM4	CSIE4	0	0	0	0	MODE4	0	SCL4

CSIE4	SIO4 operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation disabled	Clear	Port function

(2) 3-wire variable-length serial I/O mode

The 3-wire variable-length 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 ($\overline{SCK4}$), serial output line (SO4), and serial input line (SI4).

(a) Register settings

The 3-wire variable-length serial I/O mode is set via variable-length serial control register 4 (CSIM4).

Figure 10-58. CSIM4 Setting (3-Wire Variable-Length Serial I/O Mode)

After reset: 00H R/W Address: FFFFF2E2H

	7	6	5	4	3	2	1	0
CSIM4	CSIE4	0	0	0	0	MODE4	0	SCL4

CSIE4	SIO4 operation enable/disable specification		
	Shift register operation	Serial counter	Port
1	Operation enabled	Count operation enabled	Serial function + port function

MODE4	Transfer operation mode flag		
	Operation mode	Transfer start trigger	SO4 output
0	Transmit/receive mode	Write to SIO4	Normal output
1	Receive-only mode	Read from SIO4	Port function

SCL4	Clock selection
0	External clock input ($\overline{SCK4}$)
1	BRG (Baud rate generator)

The bit length of a variable-length register is set by setting bits 3 to 0 (BSEL3 to BSEL0) of CSIB4. Data is transferred MSB first while bit 4 (DIR) is 1, and is transferred LSB first while DIR is 0.

Figure 10-59. CSIB4 Setting (3-Wire Variable-Length Serial I/O Mode)

After reset: 00H R/W Address: FFFFF2E4H

	7	<6>	<5>	<4>	3	2	1	0
CSIB4	0	CMODE	DMODE	DIR	BSEL3	BSEL2	BSEL1	BSEL0

CMODE	DMODE	$\overline{SCK4}$ active level	SI4 interrupt timing	SO4 output timing
0	0	Low level	Rising edge of $\overline{SCK4}$	Falling edge of $\overline{SCK4}$
0	1	Low level	Falling edge of $\overline{SCK4}$	Rising edge of $\overline{SCK4}$
1	0	High level	Falling edge of $\overline{SCK4}$	Rising edge of $\overline{SCK4}$
1	1	High level	Rising edge of $\overline{SCK4}$	Falling edge of $\overline{SCK4}$

DIR	Serial transfer direction
0	LSB first
1	MSB first

BSEL3	BSEL2	BSEL1	BSEL0	Bit length of serial register
0	0	0	0	16 bits
1	0	0	0	8 bits
1	0	0	1	9 bits
1	0	1	0	10 bits
1	0	1	1	11 bits
1	1	0	0	12 bits
1	1	0	1	13 bits
1	1	1	0	14 bits
1	1	1	1	15 bits
Other than above				Setting prohibited

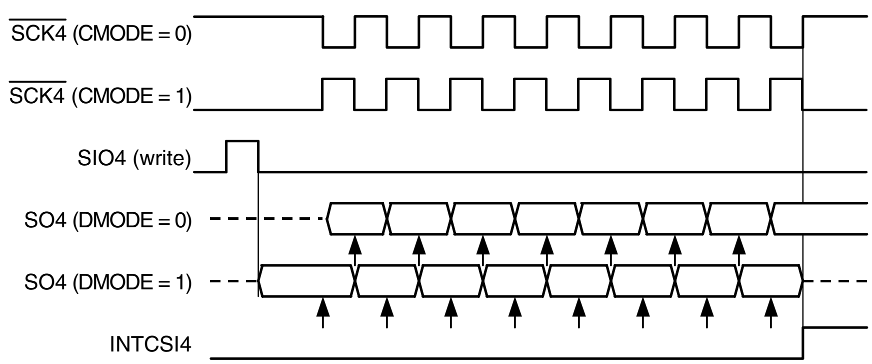
(b) Communication Operations

In the 3-wire variable-length serial I/O mode, data is transmitted and received in 8 to 16-bit units, and is specified by setting bits 3 to 0 (BSEL3 to BSEL0) of variable-length serial setting register 4 (CSIB4). Each bit of data is transmitted or received in synchronization with the serial clock.

After transfer of all bits is completed, SIC4 stops operation automatically and the interrupt request flag (INTCSI4) is set.

Bits 6 and 5 (CMODE and DMODE) of variable-length serial setting register 4 (CSIB4) can change the attribute of the serial clock ($\overline{\text{SCK4}}$) and the phases of serial data (SI4 and SO4).

Figure 10-60. Timing of 3-Wire Variable-Length Serial I/O Mode



Remark An arrow shows the SI4 data fetch timing.

When CMODE = 0, the serial clock ($\overline{\text{SCK4}}$) stops at the high level during the operation stop, and outputs the low level during a data transfer operation. When CMODE = 1, on the other hand, $\overline{\text{SCK4}}$ stops at the low level during the operation stop and outputs the high level during a data transfer operation.

The phases of the SO4 output timing and the S14 fetch timing can be shifted half a clock by setting DMODE.

However, the interrupt signal (INTCSI4) is generated at the final edge of the serial clock ($\overline{\text{SCK4}}$), regardless the setting of CSIB4.

(c) Transfer start

A serial transfer becomes possible when the following two conditions have been satisfied.

- The SIO4 operation control bit (CSIE4) = 1
- After a serial transfer, the internal serial clock is stopped.

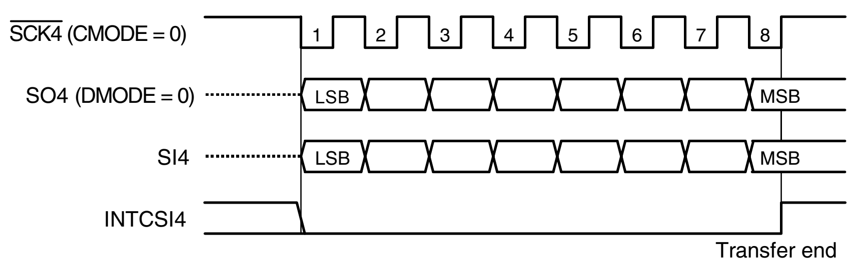
Serial transfer starts when the following operation is performed after the above two conditions have been satisfied.

- Transmit/transmit and receive mode (MODE4 = 0)
  Transfer starts when writing to SIO4.
- Receive-only mode
  Transfer starts when reading from SIO4.

Caution After data has been written to SIO4, transfer will not start even if the CSIE4 bit value is set to 1.

Completion of the final-bit transfer automatically stops the serial transfer operation and sets the interrupt request flag (INTCSI4).

Figure 10-61. Timing of 3-Wire Variable-Length Serial I/O Mode (When CSIB4 = 08H)



Remark CSIB4 = 08H (CMODE = 0, DMODE = 0, DIR = 0, BSEL3 to BSEL0 = 1000)

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).

The V850/SB1 and V850/SB2 support the low power consumption mode by low-speed conversion.

(1) Hardware start

Conversion is started by trigger input (ADTRG) (rising edge, falling edge, or both rising and falling edges can be specified).

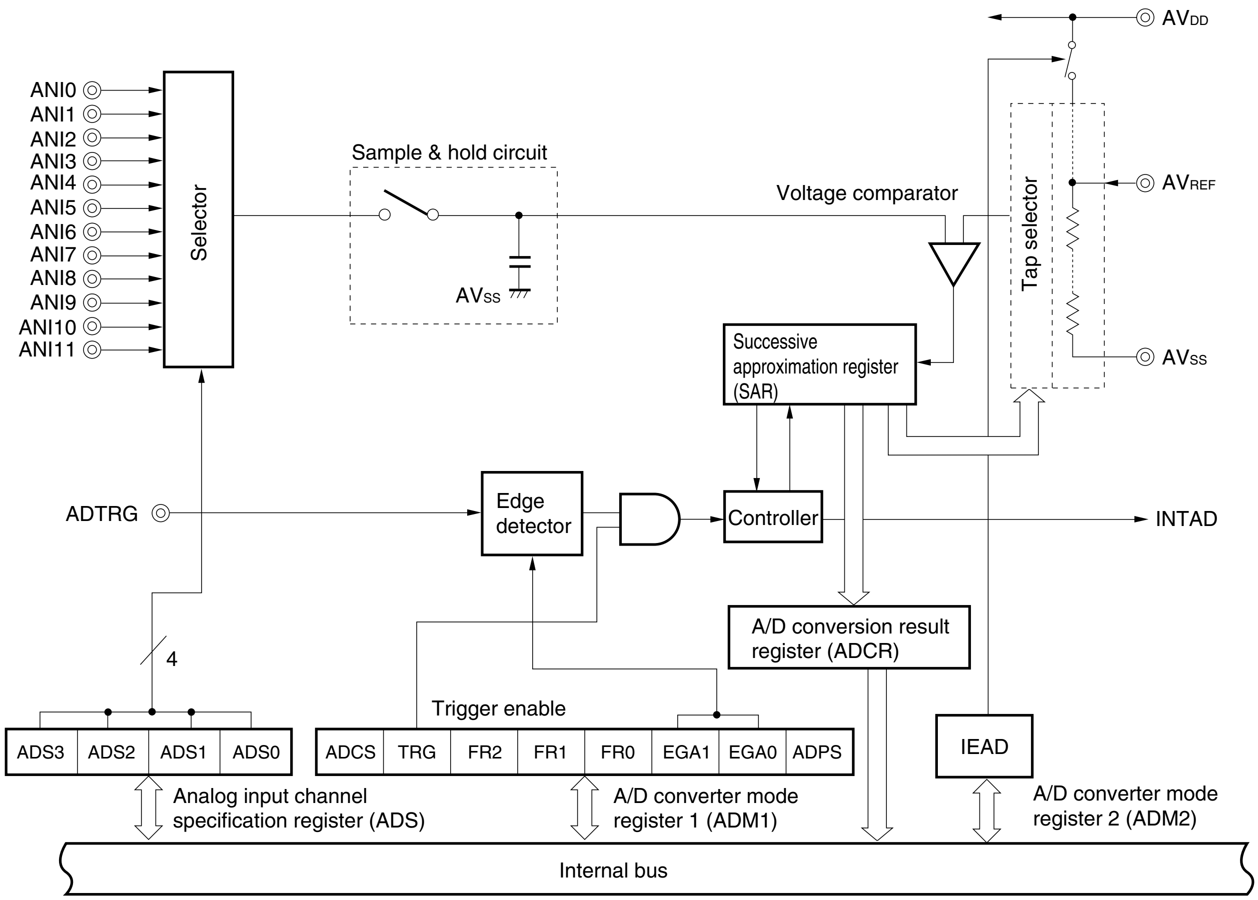
(2) Software start

Conversion is started by setting A/D converter mode register 1 (ADM1).

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 a hardware start, conversion stops once it has been completed, and an interrupt request (INTAD) is generated. If conversion has been started by means of a software start, conversion is performed repeatedly. Each time conversion is completed, INTAD is generated.

The block diagram is shown below.

Figure 11-1. Block Diagram of A/D Converter



11.2 Configuration

The A/D converter includes 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 1 (ADM1) A/D converter mode register 2 (ADM2) 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.

★ **Caution A write operation to A/D converter mode register 1 (ADM1) and the analog input channel specification register (ADS) may cause the ADCR contents to be undefined. Therefore, read the A/D conversion result during A/D conversion (ADCS = 1). Correct conversion results may not be read if the timing is other than the above.**

(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 the 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 to 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 to 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 1 (ADM1)
- Analog input channel specification register (ADS)
- A/D converter mode register 2 (ADM2)

(1) A/D converter mode register 1 (ADM1)

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 clears ADM1 to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF3C0H

	<7>	<6>	5	4	3	2	1	<0>
ADM1	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		
				Conversion time[Note 1] + stabilization time[Note 2]	fxx	
					20 MHz[Note 3]	12.58 MHz
0	0	0	0	168/fxx	8.4 μs	Setting prohibited
0	0	0	1	120/fxx	6.0 μs	9.5 μs
0	0	1	0	84/fxx	Setting prohibited	6.7 μs
0	0	1	1	60/fxx	Setting prohibited	Setting prohibited
0	1	0	0	48/fxx	Setting prohibited	Setting prohibited
0	1	0	1	36/fxx	Setting prohibited	Setting prohibited
0	1	1	0	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	12/fxx	Setting prohibited	Setting prohibited
1	0	0	0	168/fxx + 64/fxx	8.4 + 4.2 μs	Setting prohibited
1	0	0	1	120/fxx + 60/fxx	6.0 + 3.0 μs	9.5 + 4.8 μs
1	0	1	0	84/fxx + 42/fxx	Setting prohibited	6.7 + 3.3 μs
1	0	1	1	60/fxx + 30/fxx	Setting prohibited	Setting prohibited
1	1	0	0	48/fxx + 24/fxx	Setting prohibited	Setting prohibited
1	1	0	1	36/fxx + 18/fxx	Setting prohibited	Setting prohibited
1	1	1	0	Setting prohibited	Setting prohibited	Setting prohibited
1	1	1	1	12/fxx + 6/fxx	Setting prohibited	Setting prohibited

EGA1	EGA0	Valid 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	Comparator control when A/D conversion is stopped (ADCS = 0)
0	Comparator on
1	Comparator off

Notes 1. Conversion time (actual A/D conversion time).
Always set the time to 5 μs ≤ Conversion time ≤ 10 μs.

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.

3. Only in the V850/SB1.

Cautions 1. The A/D converter cannot be used when the operation frequency is 2.4 to 3.6 MHz.

2. Cut the current consumption by setting ADPS to 1 when ADCS = 0.

(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 clears 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 7 to 4 to 0.**

(3) A/D converter mode register 2 (ADM2)

ADM2 specifies connection/disconnection of AV_{DD} and AV_{REF}.
ADM2 is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears ADM2 to 00H.

After reset: 00H R/W Address: FFFFF3C8H

	7	6	5	4	3	2	1	<0>
ADM2	0	0	0	0	0	0	0	IEAD

IEAD	A/D current cut control
0	Cut between AV_{DD} and AV_{REF}
1	Connect between AV_{DD} and AV_{REF}

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.

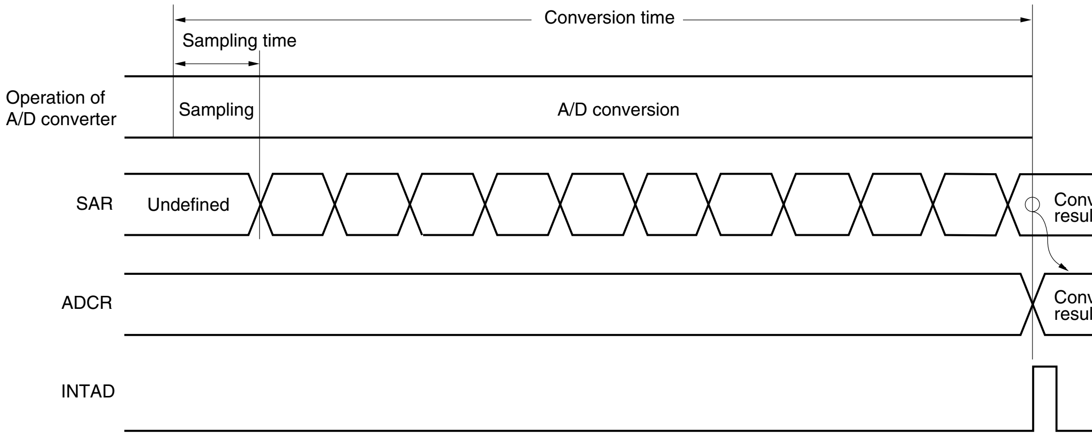
- Analog input voltage ≥ voltage tap: Bit 8 = 1
- Analog input voltage ≤ voltage tap: Bit 8 = 0

<7> The above steps are repeated until the 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 A/D converter mode register 1 (ADM1) is reset to 0 by software.

If ADM1 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 ( )

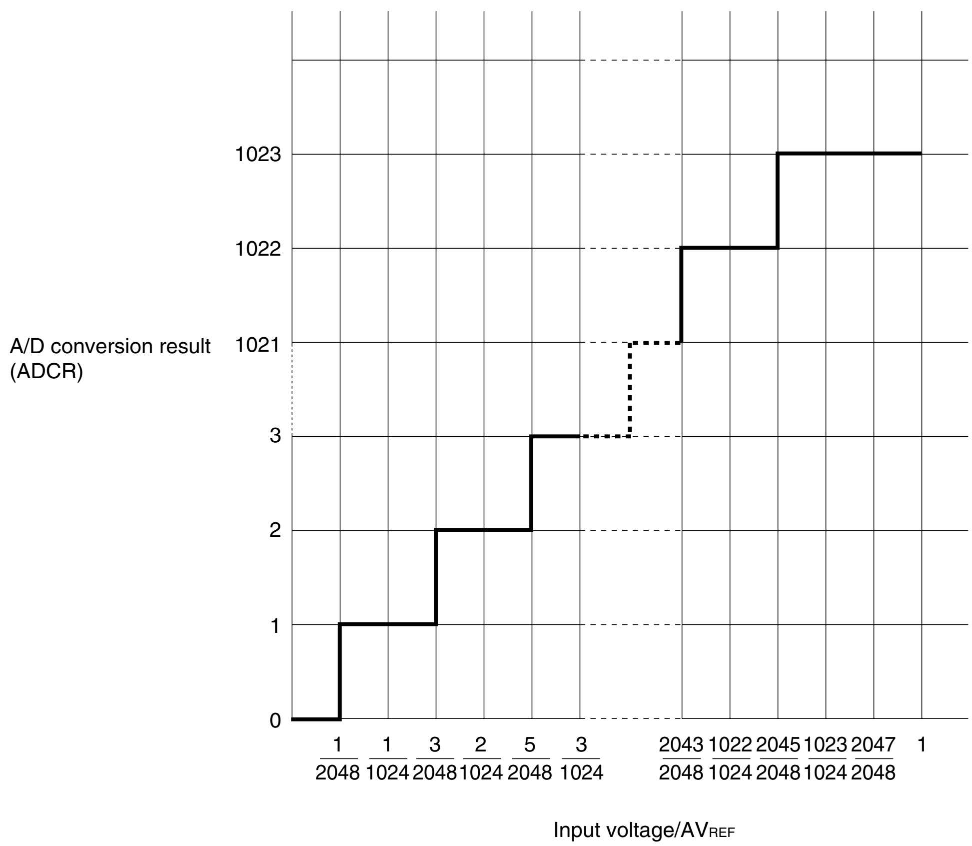
V_{IN}: Analog input voltage

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 1 (ADM1)

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

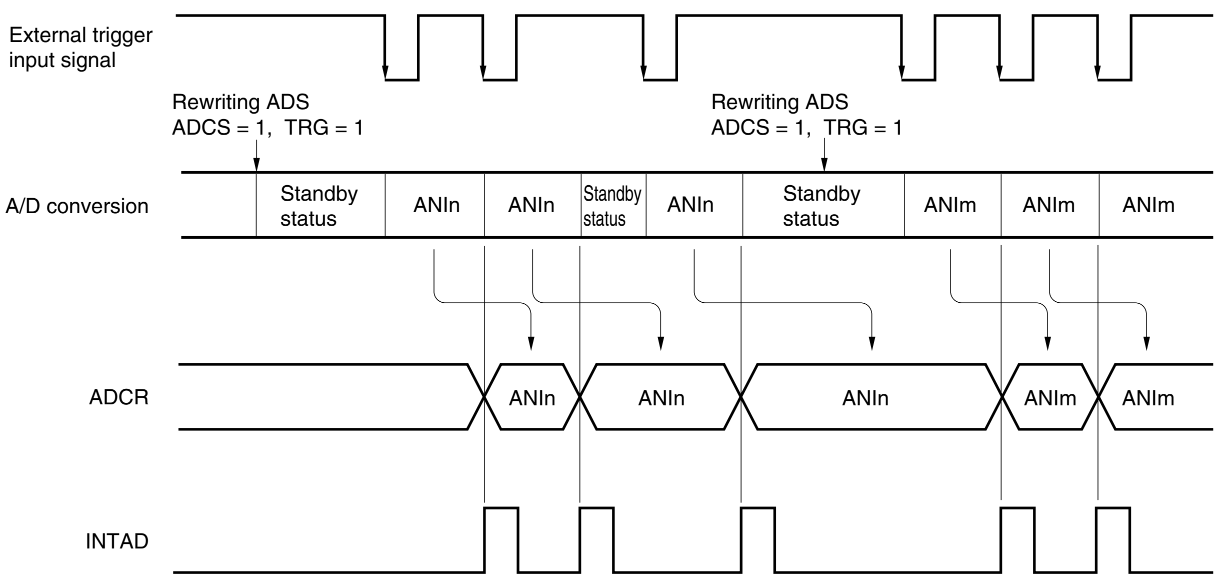
A/D conversion is on standby if bit 6 (TRG) and bit 7 (ADCS) of A/D converter mode register 1 (ADM1) are set to 1. When an external trigger signal is input, the A/D converter starts converting the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) into a digital signal.

When the 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, 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 ADM1 during A/D conversion, the conversion is immediately stopped.

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

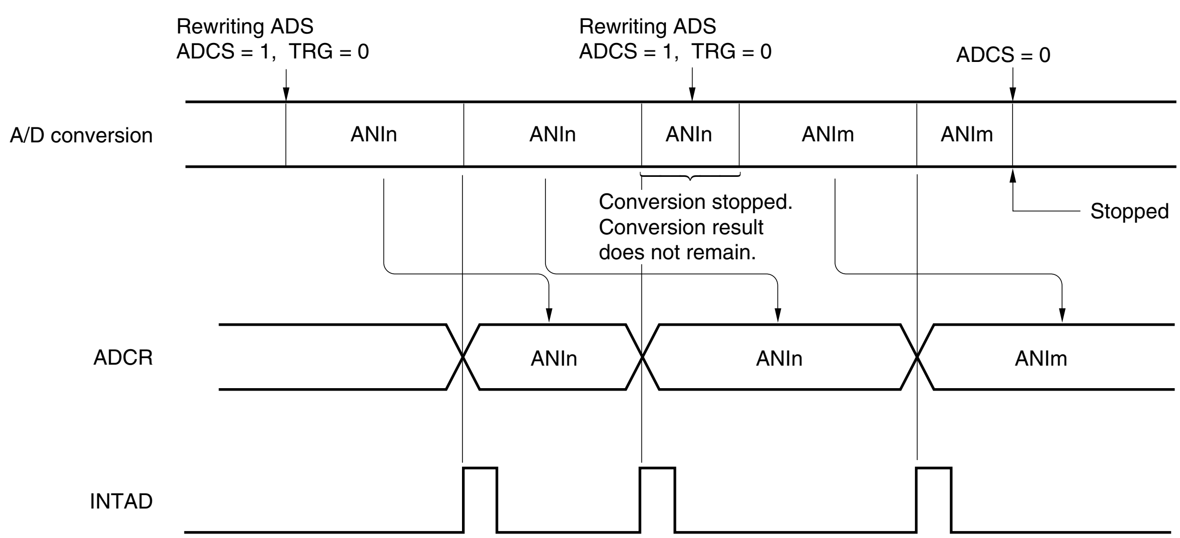
If bit 6 (TRG) of A/D converter mode register 1 (ADM1) is set to 0 and bit 7 (ADCS) is set to 1, the A/D converter starts converting the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) into a digital signal.

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 ADM1 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 Low Power Consumption Mode

The V850/SB1 and V850/SB2 feature a function that can cut or connect the current between AV_{DD} and AV_{REF}. Switching can be performed by setting A/D converter mode register 2 (ADM2).

When AV_{DD} = AV_{REF}, and when the system does not require high precision, current consumption can be reduced by connecting AV_{DD} and AV_{REF} in the normal mode or disconnecting them in standby mode after opening the AV_{REF} pin.

The conversion precision of the reference voltage is reduced since the reference voltage is supplied from AV_{DD} via a switch.

When the A/D converter is not used, cut the tap selector that reduces the current when A/D conversion is stopped (ADCS = 0), and the supply voltage (AV_{DD}), in order to reduce the current consumption.

- Set the ADPS bit of A/D converter mode register 1 (ADM1) to 1.
- Clear the IEAD bit of A/D converter mode register 2 (ADM2) to 0.

When the ADPS bit is cleared to 0 (comparator on) again, a stabilization time (5 μs max.) is required until the A/D converter is started. Therefore, use software to ensure that a wait time of 5 μs elapses.

11.6 Cautions

(1) Current consumption in standby mode

The A/D converter stops operation in the IDLE/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 A/D converter mode register 1 (ADM1) to 0).

To reduce the current consumption in the IDLE/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 to 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.

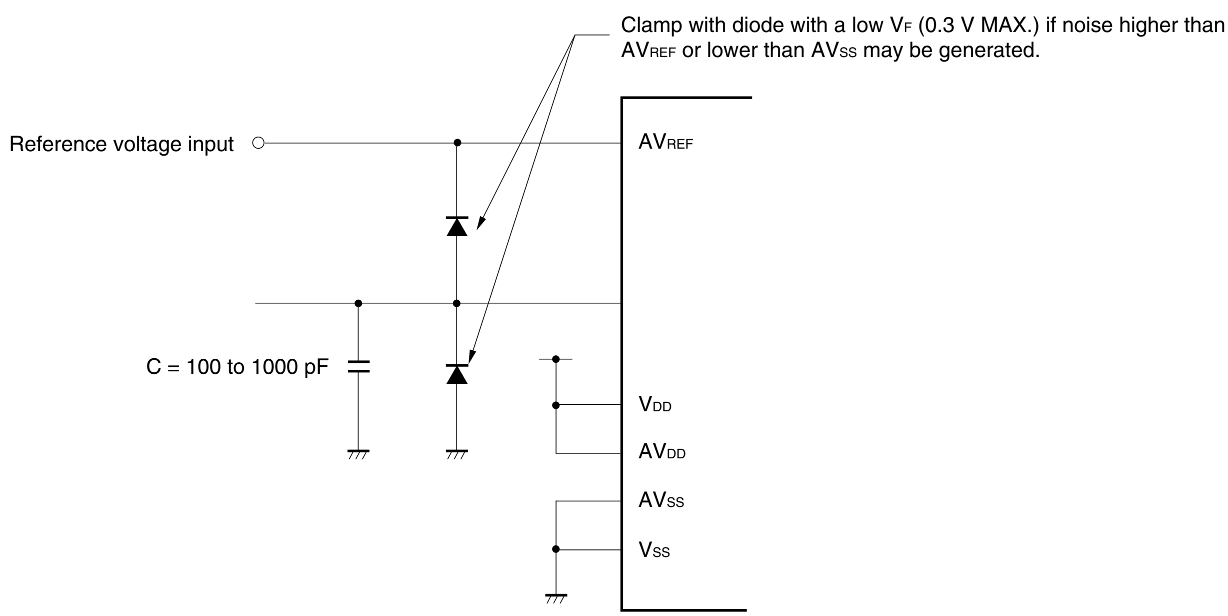
<3> Conflict between writing of ADCR and writing A/D converter mode register 1 (ADM1) or analog input channel specification register (ADS)

When ADM1 or ADS write is performed immediately after ADCR write following the end of A/D conversion,
★ the conversion result cannot be guaranteed since an undefined value is stored in the ADCR register.

(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.

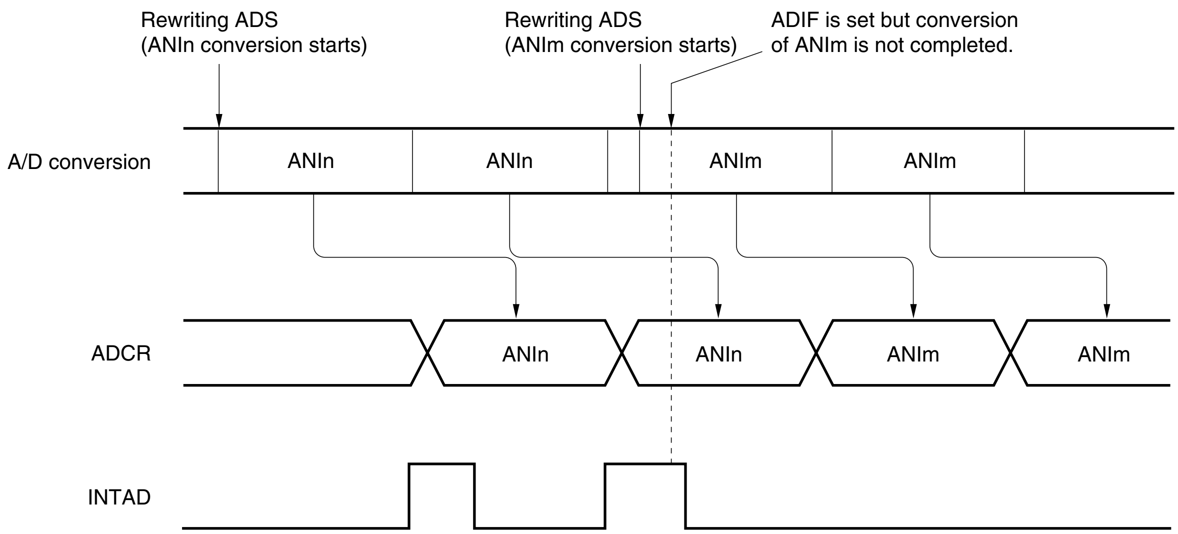
(7) Interrupt request flag (ADIF)

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 Generation Timing



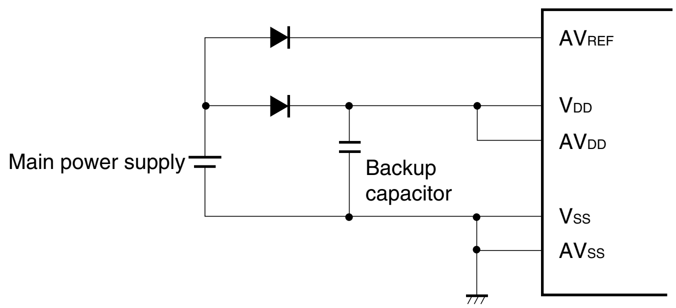
Remarks 1. n = 0, 1, ..., 11

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 converter result register (ADCR)

A write operation to A/D converter mode register 1 (ADM1) and the analog input channel specification register ★ (ADS) may cause the ADCR contents to be undefined. Therefore, read the A/D conversion result during A/D conversion (ADCS = 1). Incorrect conversion results may be read out at a timing other than the above.

★ 11.7 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.

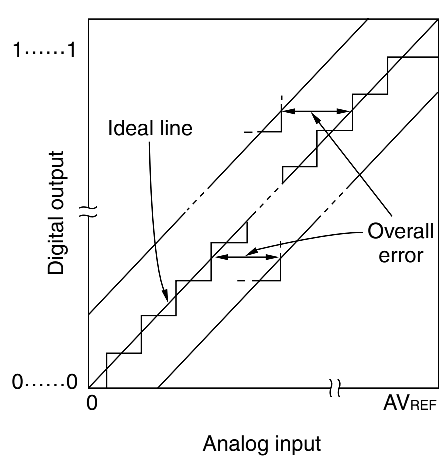
1LSB = $1/2^{10}$ = 1/1024
= 0.098%FSR

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

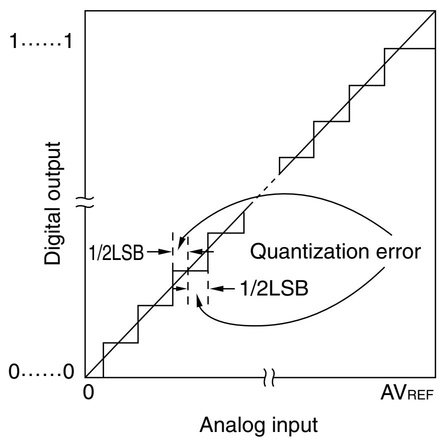


(3) Quantization error

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.

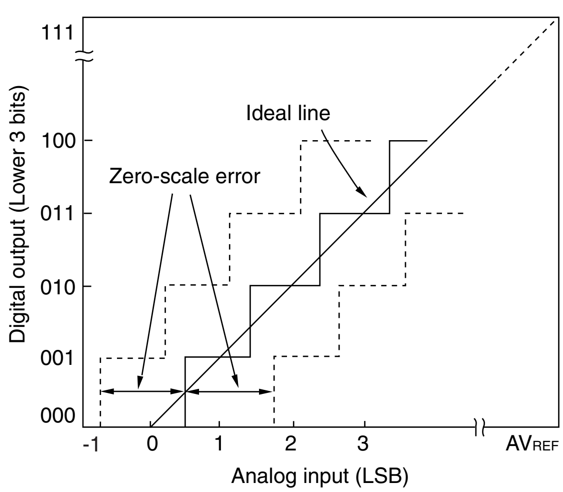
Figure 11-10. Quantization Error



(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.

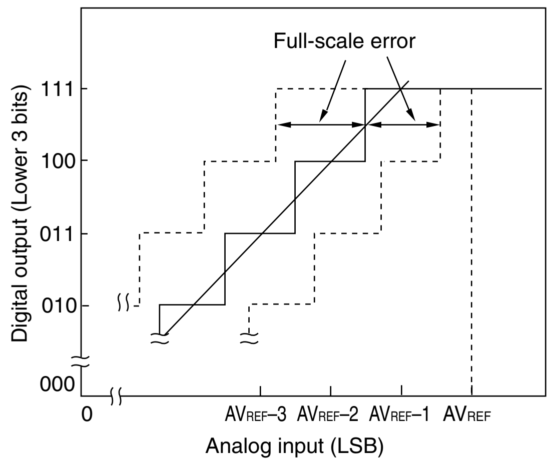
Figure 11-11. Zero-Scale Error



(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.

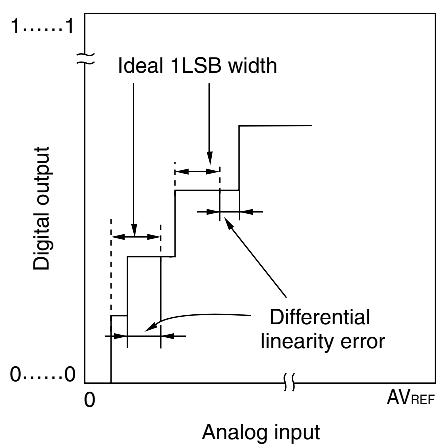
Figure 11-12. Full-Scale Error



(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.

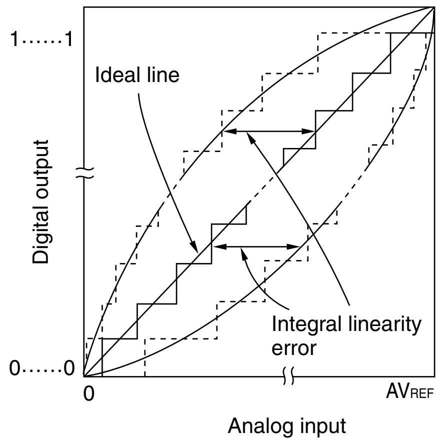
Figure 11-13. Differential Linearity Error



(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

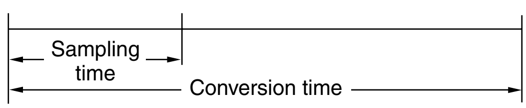
This expresses the time from when the analog input voltage was applied to the time when the digital output was obtained.

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 DMA (Direct Memory Access) controller transfers data between memory and peripheral I/Os based on DMA requests sent from on-chip peripheral hardware (such as the serial interface, timer, or A/D converter).

This product includes six independent DMA channels that can transfer data in 8-bit and 16-bit units. The maximum number of transfers is 256 (when transferring data in 8-bit units).

After a DMA transfer has occurred a specified number of times, DMA transfer completion interrupt (INTDMA0 to INTDMA5) requests are output individually from the various channels.

The priority levels of the DMA channels are fixed as follows for simultaneous generation of multiple DMA transfer requests.

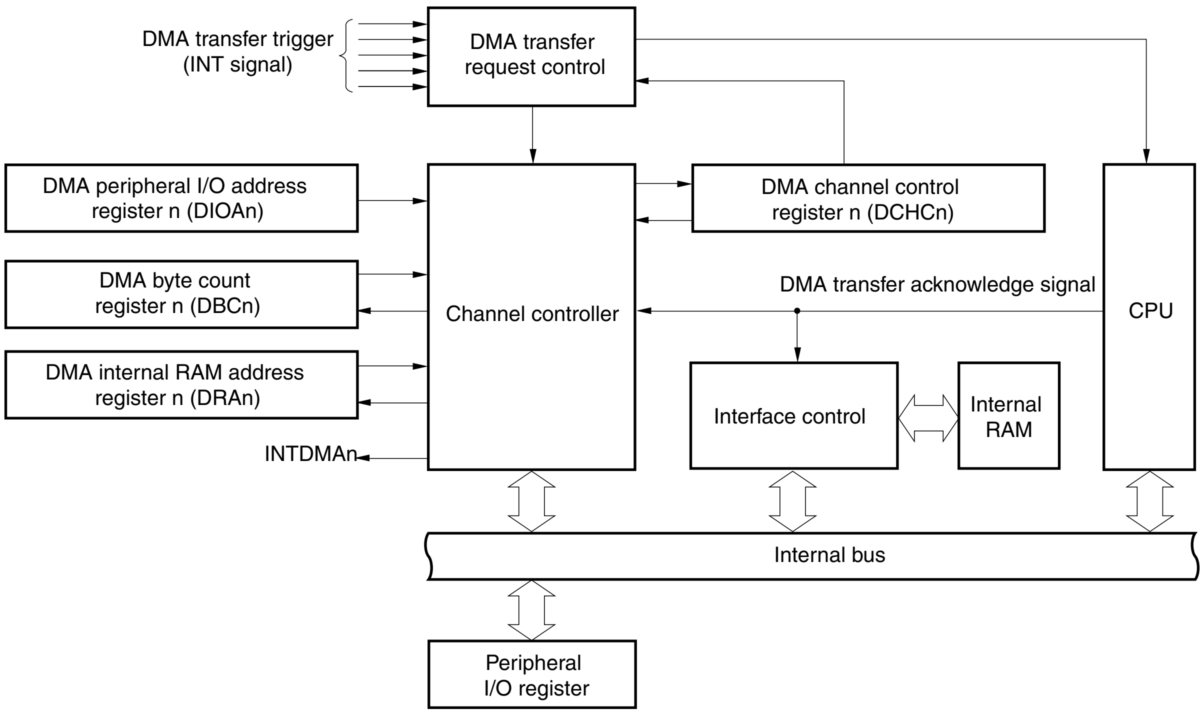
DMA0 > DMA1 > DMA2 > DMA3 > DMA4 > DMA5

12.2 Transfer Completion Interrupt Request

After a DMA transfer has occurred a specified number of times and the TCn bit in the corresponding DMA channel control register (DCHCn) has been set to 1, a DMA transfer completion interrupt request (INTDMA0 to INTDMA5) is sent to the interrupt controller for each channel.

★ 12.3 Configuration

Figure 12-1. Block Diagram of DMA



Remark n = 0 to 5

(1) DMA transfer request control block

The DMA transfer request control block generates a DMA transfer request signal for the CPU when the required DMA transfer trigger (INT signal) 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.

For the INT signal, refer to the TTYPn1 and TTYPn0 bits in **12.4 (5) DMA channel control registers 0 to 5 (DCHC0 to DCHC5)**.

(2) Channel control block

The channel control block distinguishes the DMA transfer channel (DMA0 to DMA5) 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 5 (DIOA0 to DIOA5)

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 DIOA3 FFFFF1B0H
DIOA1 FFFFF190H DIOA4 FFFFF1C0H
DIOA2 FFFFF1A0H DIOA5 FFFFF100H

	15	14	13	12	11	10	9 to 1	0
DIOAn (n = 0 to 5)	0	0	0	0	0	0	IOAn9 to IOAn1	0

Caution The following peripheral I/O registers must not be set.
P4, P5, P6, P9, P11, PM4, PM5, PM6, PM9, PM11, MM, DWC, BCC, SYC, PSC, PCC, SYS, PRCMD, DIOAn, DRAn, DBCn, DCHCn, CORCN, CORRQ, CORADn, Interrupt control register (xxICn), ISPR

(2) DMA internal RAM address registers 0 to 5 (DRA0 to DRA5)

These registers set DMA channel n internal RAM addresses (n = 0 to 5).

Since each product has a different internal RAM capacity, the internal RAM areas that are usable for DMA differ depending on the product. The internal RAM areas that can be set in the DRAn registers for each product are shown below.

Table 12-1. Internal RAM Area Usable in DMA

Product		Internal RAM Capacity	RAM Size Usable in DMA	RAM Area Usable in DMA
V850/SB1	μPD703031B, 703031BY	8 KB	8 KB	xxFFD000H to xxFFEFFFH
V850/SB2	μPD703034B, 703034BY			
V850/SB1	μPD703031A, 703031AY,	12 KB	12 KB	xxFFC000H to xxFFEFFFH
V850/SB2	μPD703034A, 703034AY			
V850/SB1	μPD703033A, 703033AY, 703033B, 703033BY, 70F3033A, 70F3033AY, 70F3033B, 70F3033BY	16 KB	16 KB	xxFFB000H to xxFFEFFFH
V850/SB2	μPD703035A, 703035AY, 703035B, 703035BY, 70F3035A, 70F3035AY, 70F3035B, 70F3035BY			
V850/SB1	μPD703030B, 703030BY, 703032A, 703032AY, 703032B, 703032BY, 70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY	24 KB	16 KB	xxFF9000H to xxFFBFFFH, xxFFE000H to xxFFEFFFH
V850/SB2	μPD703036H, 703036HY, 703037A, 703037AY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037A, 70F3037AY, 70F3037H, 70F3037HY			

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 5).

These registers are can be read/written in 16-bit units.

After reset: Undefined R/W Address: DRA0 FFFFF182H DRA3 FFFFF1B2H
DRA1 FFFFF192H DRA4 FFFFF1C2H
DRA2 FFFFF1A2H DRA5 FFFFF1D2H

	15	14	13 to 0
DRAn	0	0	RAn13 to RAn0

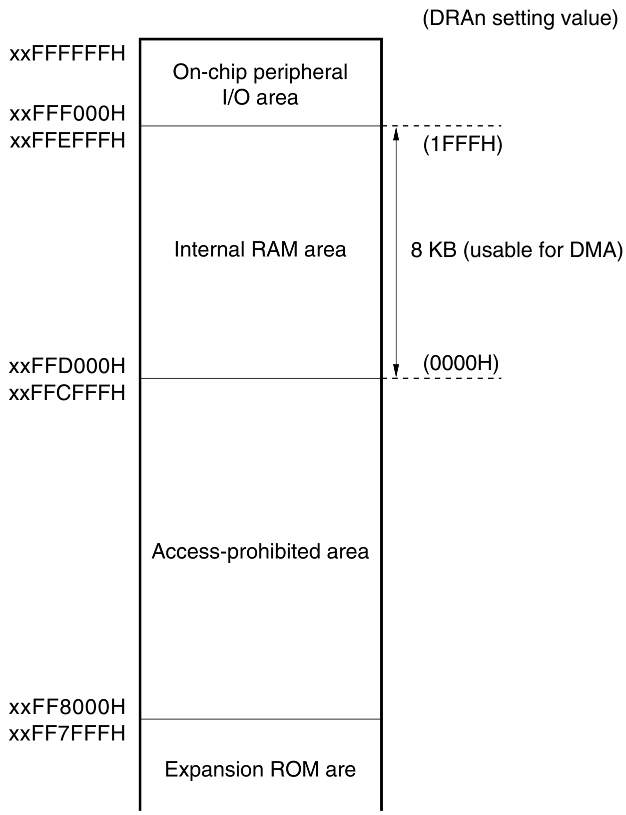
(n = 0 to 5)

The correspondence between DRAn setting value and internal RAM area is shown below.

★ **(a) V850/SB1 (μPD703031B, 703031BY), V850/SB2 (μPD703034B, 703034BY)**

Set the DRAn register to a value in the range of 0000H to 1FFFH (n = 0 to 5).
Setting is prohibited for values between 2000H and 3FFFH.

Figure 12-2. Correspondence Between DRAn Setting Value and Internal RAM (8 KB)



Cautions 1. Do not set odd addresses for 16-bit transfer (DCHCn register DSn = 1).

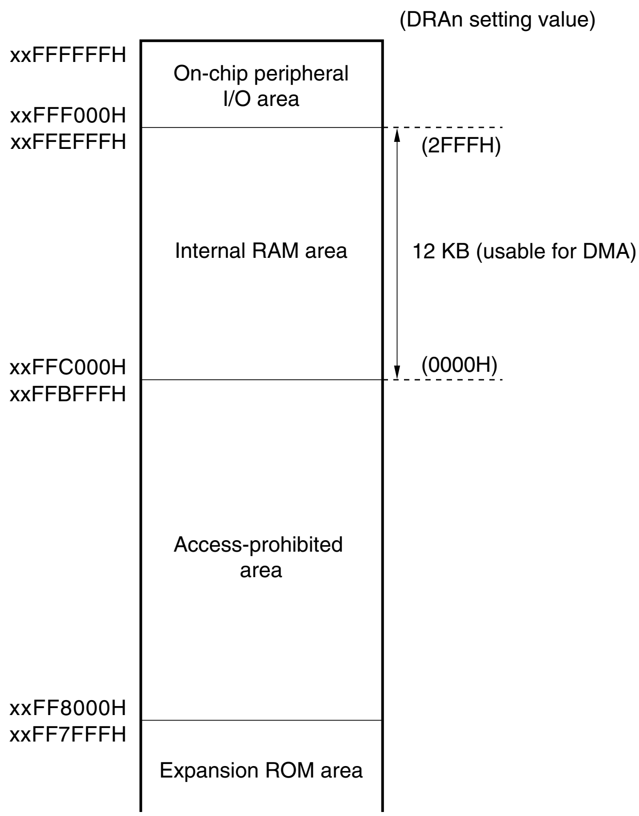
2. While the increment function is being used (DCHCn register DDADn = 0), if the DRAn register value is set to 1FFFH, it will be incremented to 2000H, and will thus become a setting-prohibited value.

Remark The DRAn register setting values are in parentheses.

★ **(b) V850/SB1 (μPD703031A, 703031AY), V850/SB2 (μPD703034A, 703034AY)**

Set the DRAn register to a value in the range of 0000H to 2FFFH (n = 0 to 5).
Setting is prohibited for values between 3000H and 3FFFH.

Figure 12-3. Correspondence Between DRAn Setting Value and Internal RAM (12 KB)



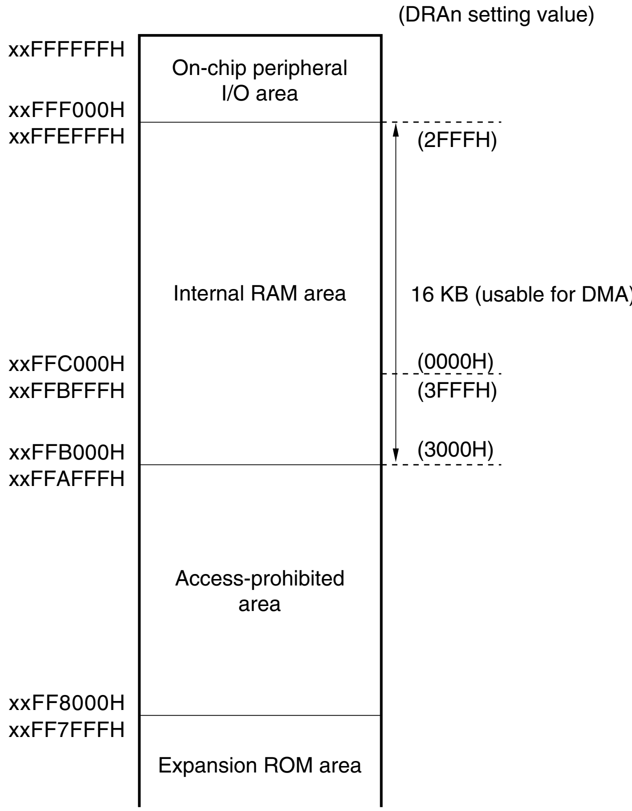
Cautions 1. Do not set odd addresses for 16-bit transfer (DCHCn register DSn = 1).

2. While the increment function is being used (DCHCn register DDADn = 0), if the DRAn register value is set to 2FFFH, it will be incremented to 3000H, and will thus become a setting-prohibited value.

Remark The DRAn register setting values are in parentheses.

★ **(c) V850/SB1**
(μPD703033A, 703033AY, 703033B, 703033BY, 70F3033A, 70F3033AY, 70F3033B, 70F3033BY)
V850/SB2
(μPD703035A, 703035AY, 703035B, 703035BY, 70F3035A, 70F3035AY, 70F3035B, 70F3035BY)
Set the DRAn register to a value in the range of 000H to 2FFFH or 3000H to 3FFFH (n = 0 to 5).

Figure 12-4. Correspondence Between DRAn Setting Value and Internal RAM (16 KB)



Caution Do not set odd addresses for 16-bit transfer (DCHCn register DSn =1).

Remark The DRAn register setting values are in parentheses.

★ **(d) V850/SB1**

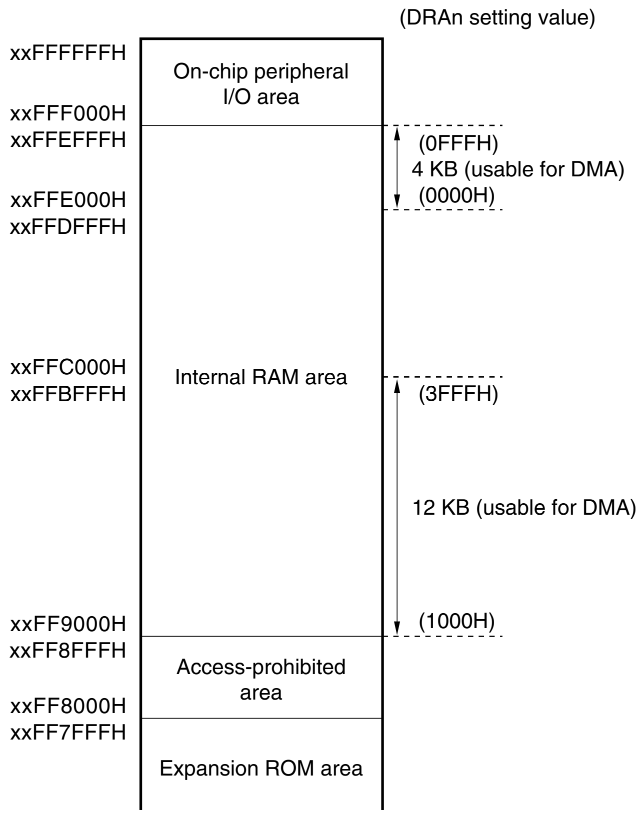
(μPD703030B, 703030BY, 703032A, 703032AY, 703032B, 703032BY, 70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY)

V850/SB2

(μPD703036H, 703036HY, 703037A, 703037AY, 703037H, 703037HY, 70F3036H, 70F3036HY, 70F3037A, 70F3037AY, 70F3037H, 70F3037HY)

Set the DRAn register to a value in the range of 0000H to 0FFFH or 1000H to 3FFFH (n = 0 to 5).

Figure 12-5. Correspondence Between DRAn Setting Value and Internal RAM (24 KB)



Caution Do not set odd addresses for 16-bit transfer (DCHCn register DSn =1).

Remark The DRAn register setting values are in parentheses.

(3) DMA byte count registers 0 to 5 (DBC0 to DBC5)

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.

These registers are can be read/written in 8-bit units.

After reset: Undefined R/W Address: DBC0 FFFFF184H DBC3 FFFFF1B4H
DBC1 FFFFF194H DBC4 FFFFF1C4H
DBC2 FFFFF1A4H DBC5 FFFFF1D4H

	7	6	5	4	3	2	1	0
DBCn	BCn7	BCn6	BCn5	BCn4	BCn3	BCn2	BCn1	BCn0

(n = 0 to 5)

Caution Values set to bit 0 are ignored during 16-bit transfers.

(4) DMA start factor expansion register (DMAS)

This is an 8-bit register for expanding the factors that start DMA.

The DMA start factor is decided according to the combination of TTYPn1 and TTYPn0 of the DCHCn register.

For setting bits DMAS2 to DMAS0, refer to **12.4 (5) DMA channel control registers 0 to 5 (DCHC0 to DCHC5)**

(n = 0 to 5).

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

After reset: 00H R/W Address: FFFFF38EH

	7	6	5	4	3	2	1	0
DMAS	0	0	0	0	0	DMAS2	DMAS1	DMAS0

(5) DMA channel control registers 0 to 5 (DCHC0 to DCHC5)

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 DCHC3 FFFFF1B6H
DCHC1 FFFFF196H DCHC4 FFFFF1C6H
DCHC2 FFFFF1A6H DCHC5 FFFFF1D6H

	<7>	6	<5>	4	3	<2>	<1>	<0>
DCHCn (n = 0 to 5)	TCn	0	DDADn	TTYPn1	TTYPn0	TDIRn	DSn	ENn

TCn	DMA transfer completed/not completed[Note 1]
0	Not completed
1	Completed

DDADn	Internal RAM address count direction control
0	Increment
1	Address is fixed

Channel n	DMAS2	DMAS1	DMAS0	TTYPn1	TTYPn0	DMA transfer start factor setting
0	x	x	x	0	0	INTCSI0/INTIIC0[Note 2]
				0	1	INTCSI1/INTSR0
				1	0	INTAD
				1	1	INTTM00
1	x	x	0	0	0	INTCSI0/INTIIC0[Note 2]
			1	0	0	INTCSI1/INTSR0
			x	0	1	INTST0
				1	0	INTP0
				1	1	INTTM10
2	x	0	x	0	0	INTIIC1[Note 2]
		1		0	0	INTCSI3/INTSR1
		x		0	1	INTP6
				1	0	INTIE1 (V850/SB2 only)
				1	1	INTAD
3	0	x	x	0	0	INTIIC1[Note 2]
	1			0	0	INTCSI3/INTSR1
	x			0	1	INTCSI2
				1	0	INTIE1 (V850/SB2 only)
				1	1	INTTM4

★ **Caution If an interrupt source is generated asynchronously to the internal system clock, do not set the interrupt source as multiple DMA transfer triggers at the same time (for example, when the serial interface is operated with external clock input).**
If it is set, the DMA transfer priority may be reversed.

(2/2)

Channel n	DMAS2	DMAS1	DMAS0	TTYPn1	TTYPn0	DMA transfer start factor setting
4	x	x	x	0	0	INTST1
				0	1	INTCSI4
				1	0	INTAD
				1	1	INTTM2
5	x	x	x	0	0	INTCSI3/INTSR1
				0	1	INTCSI4
				1	0	INTCSI2
				1	1	INTTM6

TDIRn	Transfer direction control between peripheral I/Os and internal RAM[Note 3]
0	From internal RAM to peripheral I/Os
1	From peripheral I/Os to internal RAM

DSn	Control of transfer data size for DMA transfer[Note 3]
0	8-bit transfer
1	16-bit transfer

ENn	Control of DMA transfer enable/disable status[Note 4]
0	Disabled
1	Enabled (reset to 0 after DMA transfer is completed)

Notes
1. TCn (n = 0 to 5) 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 and INTIIC1 are available only in the Y versions (products with on-chip I^2C).
3. 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).
4. After the specified number of transfers are completed, this bit is cleared to 0.

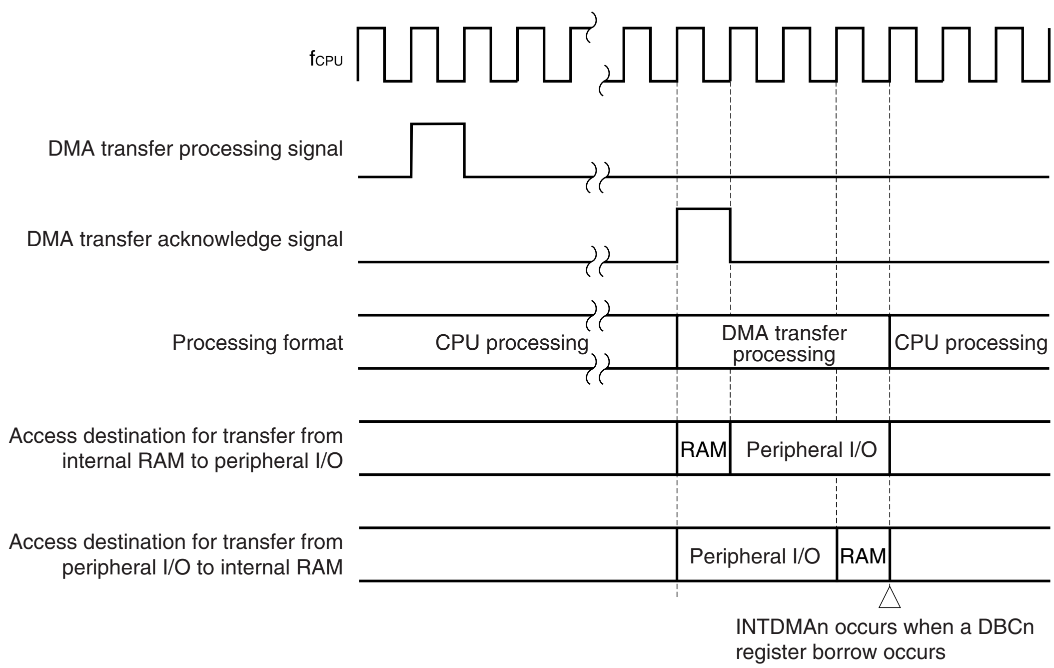
★ 12.5 Operation

When a DMA transfer request is generated during CPU processing, 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.
A DMA transfer operation timing chart is shown below.

Figure 12-6. DMA Transfer Operation Timing

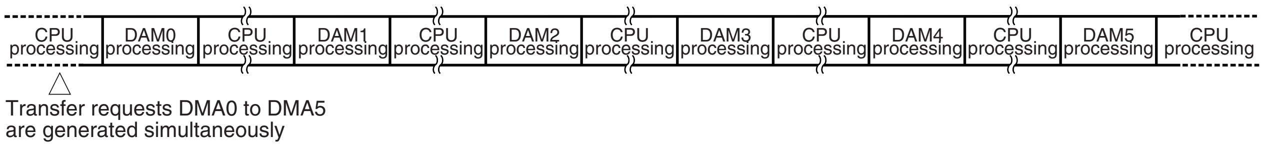


Remark n = 0 to 5

If two or more DMA transfer requests are generated simultaneously, the DMA transfer requests are executed in accordance with the following priority order: DMA0 > DMA1 > DMA2 > DMA3 > DMA4 > DMA5. 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 is executed.

The processing when the transfer requests DMA0 to DMA5 are generated simultaneously is shown below.

Figure 12-7. Processing When Transfer Requests DMA0 to DMA5 Are Generated Simultaneously



DMA operation stops only in the IDLE/STOP mode. In the HALT mode, DMA operation continues. DMA also operates during the bus hold period and after access to the external memory.

12.6 Cautions

When using the DMA function, if all the following conditions are met during the EI state (interrupt enable state), two interrupts occur when only one interrupt would occur normally.

[Occurrence conditions]

(i) A bit manipulation instruction (SET1, CLR1, NOT1, TST1) was executed to the interrupt request flag (xxIFn) of the interrupt control register (xxICn).

(ii) An interrupt was processed by hardware at the same register as the register used in (i).

Remark xx: Identifying name of peripheral unit (see **Table 5-2**)
n: Peripheral unit number (see **Table 5-2**)

For example, when using the DMA function, if an unmasked INTCSI0 interrupt occurs during bit manipulation of the interrupt request flag (CSIF0) of the CSIC0 register by the CLR1 instruction, INTCSI0 interrupt servicing occurs twice.

Under such conditions, because the interrupt request flag (xxIF) is not cleared (0) by hardware when the interrupt servicing is acknowledged, the interrupt servicing is executed again after RETI instruction execution (interrupt servicing restoration).

Therefore, use the DMA function under either of the following conditions.

[Use conditions]

(i) When bit manipulation is executed for the interrupt control register (xxICn), the DI instruction must be executed before the manipulation and the EI instruction must be executed after the manipulation.

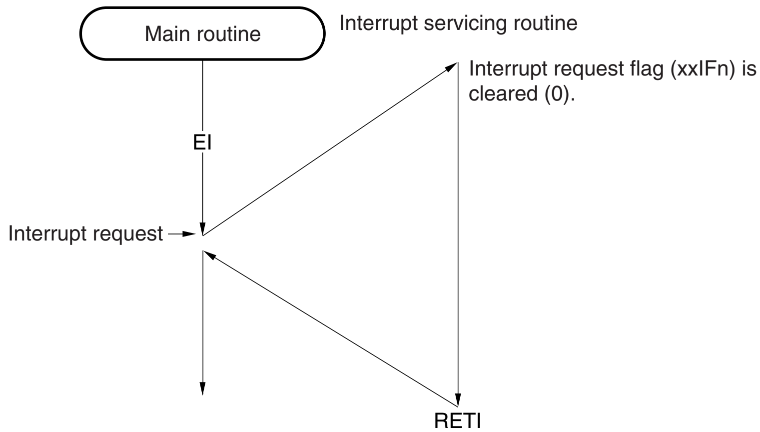
(ii) The interrupt request flag (xxIFn) must be cleared (0) at the start of the interrupt routine.

Caution When the DMA function is not used, execution of (i) or (ii) is not necessary.

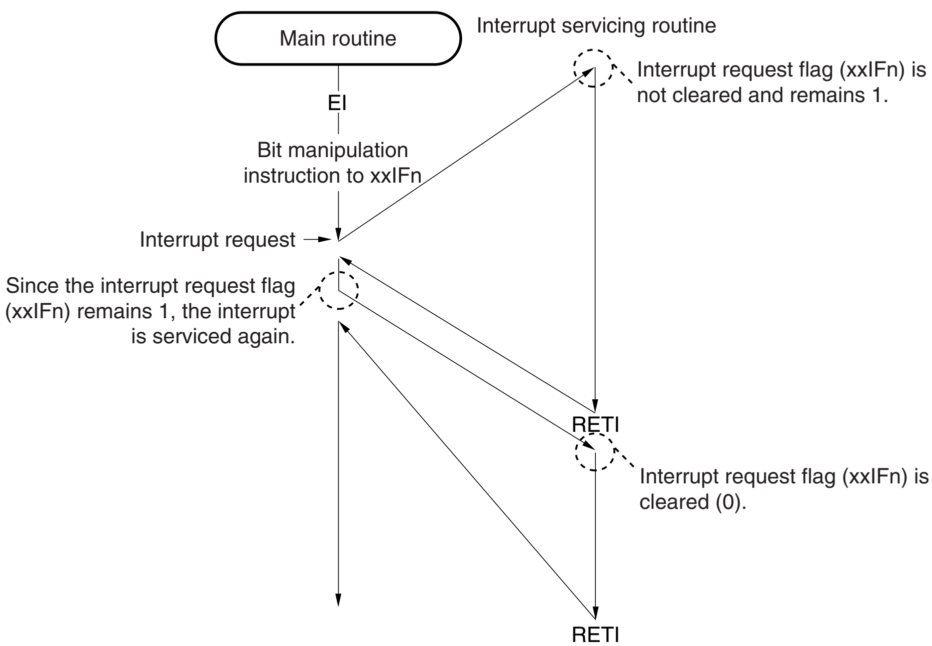
Remark xx: Identification name of each peripheral unit (see **Table 5-2**)
n: Peripheral unit number (see **Table 5-2**)

Figure 12-8. When Interrupt Servicing Occurs Twice During DMA Operation (1/2)

(a) Normal interrupt servicing



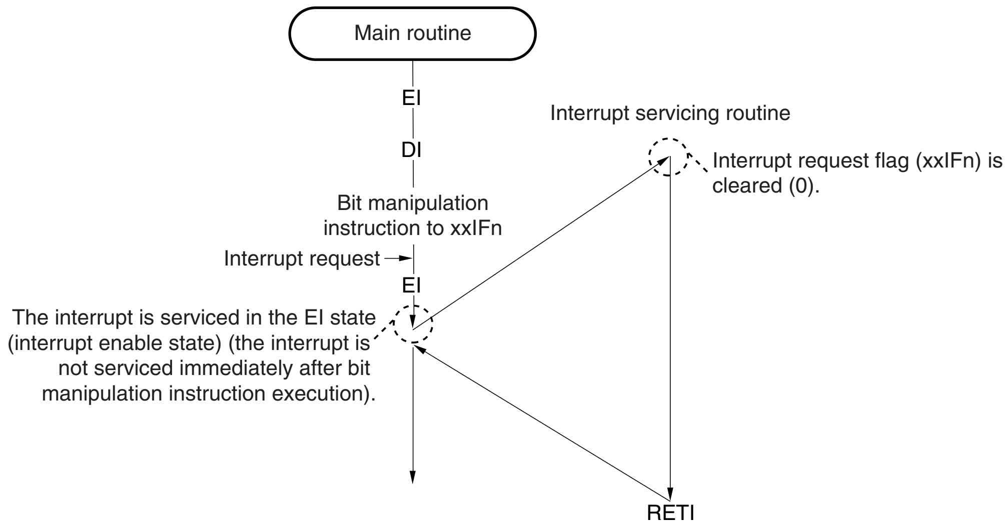
(b) Interrupt servicing when interrupt servicing occurs twice



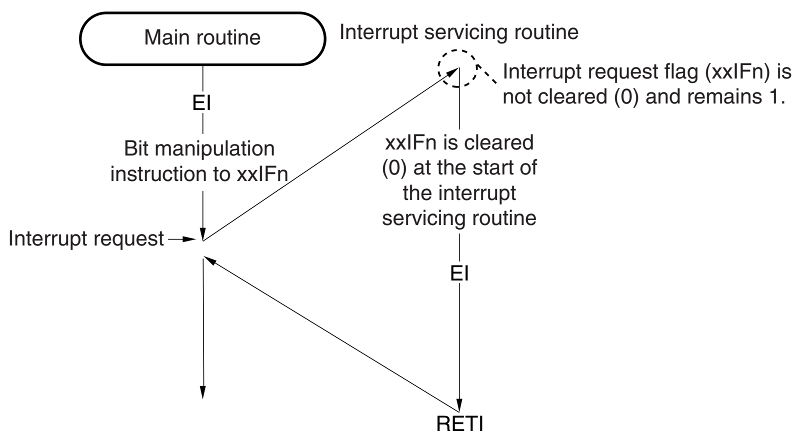
Remark xx: Identification name of each peripheral unit (see **Table 5-2**)
n: Peripheral unit number (see **Table 5-2**)

Figure 12-8. When Interrupt Servicing Occurs Twice During DMA Operation (2/2)

(c) Countermeasure (use condition (i))



(d) Countermeasure (use condition (ii))



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

n: Peripheral unit number (see **Table 5-2**)

CHAPTER 13 REAL-TIME OUTPUT FUNCTION (RTO)

13.1 Function

The V850/SB1 and V850/SB2 incorporate a real-time output function (RTO) 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.

The real-time output port can be set in port mode or real-time output port mode in 1-bit units.

★ 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.

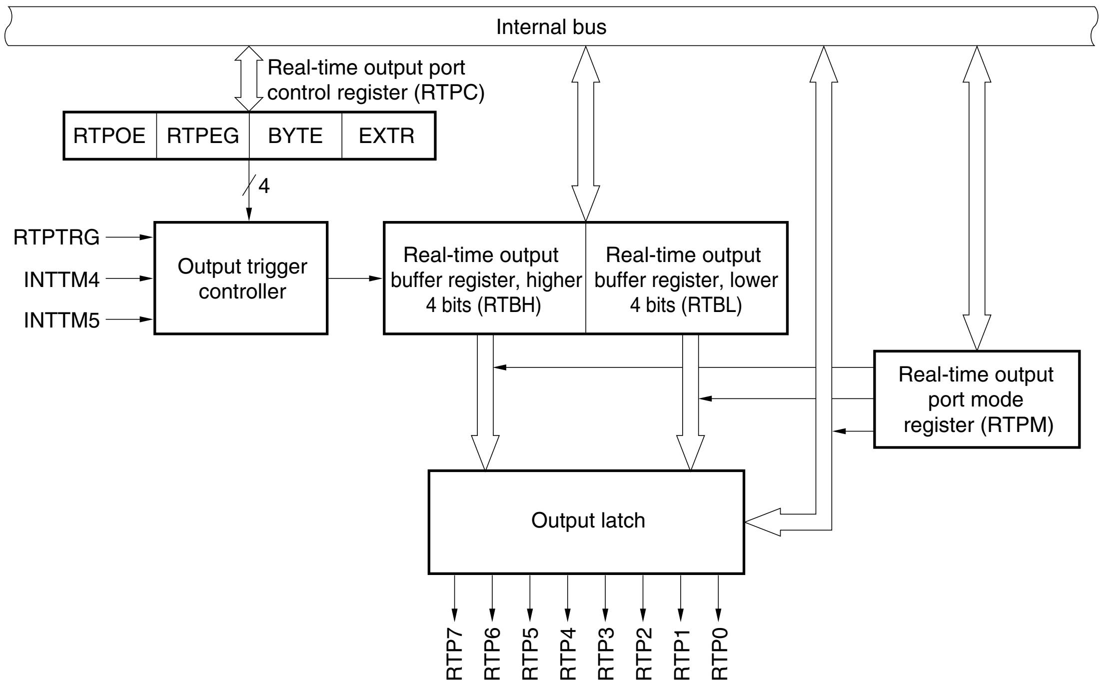
  External interrupt: RTPTRG

  Internal interrupt: INTTM4, INTTM5

13.3 Configuration

A block diagram of RTO is shown below.

Figure 13-1. Block Diagram of RTO



RTO includes 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 peripheral I/O register 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.

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

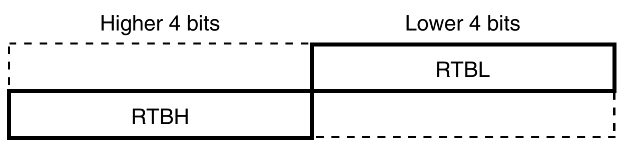


Table 13-2. Operation When Real-Time Output Buffer Registers Are Manipulated

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 and before 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 RTO Control Registers

RTO is controlled by using the following two types of 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 an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears 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. 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).

(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 an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears 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	Disable operation[Note]
1	Enable operation

RTPEG	Valid edge of RTPTRG
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	Do not use RTPTRG as real-time output trigger
1	Use RTPTRG 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) Initial setting
(i) Set the value to be output first to the real-time output port 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) Select 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 latches (P100 to P107) of port 10 to 0, and 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.

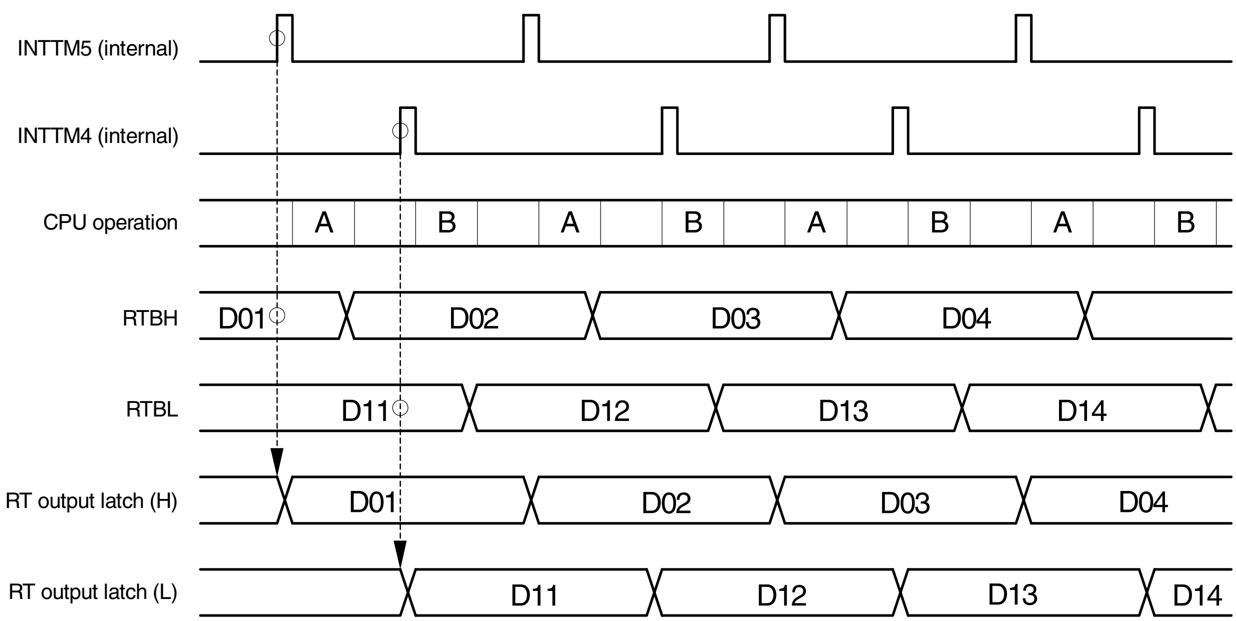
13.6 Operation

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 transmit trigger (set by EXTR and BYTE[Note]). Of the transferred data, only the data of the bits specified in the real-time output 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 the real-time output port control register (RTPC)
BYTE: Bit 5 of the 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 software to avoid a conflict.

- Conflict between the switch operation from the RTP mode to the port mode (RTPOE = 1) and the valid edge of the selected real-time output port trigger

- Conflict between the write operation to the real-time output buffer register (RTBL, RTBH) in the RTP mode and the valid edge of the selected real-time output port trigger

CHAPTER 14 PORT FUNCTION

14.1 Port Configuration

The V850/SB1 and V850/SB2 include 83 I/O port pins from ports 0 to 11 (12 port pins are input only).

There are three power supplies for the I/O buffers: AV_{DD}, BV_{DD}, and EV_{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	When using A/D converter: $4.5\ V \leq AV_{DD} \leq 5.5\ V$ When using A/D converter: $3.5\ V \leq AV_{DD} \leq 5.5\ V$
BV_{DD}	Port 4, port 5, port 6, port 9, CLKOUT	$3.0\ V \leq BV_{DD} \leq 5.5\ V$
EV_{DD}	Port 0, port 1, port 2, port 3, port 10, port 11, $\overline{\text{RESET}}$	$3.0\ V \leq EV_{DD} \leq 5.5\ V$

Caution The electrical specifications in the case of 3.0 to up to 4.0 V are different from those for 4.0 to 5.5 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 using P00 to P04 as the NMI or INTP0 to INTP3 pins, noise is eliminated by an analog noise eliminator.

When using P05 to P07 as the INTP4/ADTRG, INTP5/RTPTRG, and INTP6 pins, noise is eliminated 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, 1)
0	Output 0
1	Output 1

Remark In input mode: When the P0 register 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 the P0 register is read, the P0 register's 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. Port 0 Alternate Function Pins

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 the port 0 register (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 the P0 register. Also, the P0 register (output latch) values can be read by reading the P0 register 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 for the INTP4, INTP5, ADTRG, and RTPTRG pins if the same level is not detected three times consecutively. The noise-elimination width can be changed for the INTP6 pin (see **5.3.8 (3) Noise elimination of INTP6 pin**).

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 pulses, constant-level input is required for 3 clocks or more.

3. If noise is occurring in synchronization with the sampling clock, it may not be recognized as noise. 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 8-bit or 1-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 8-bit or 1-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 8-bit or 1-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 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF0C2H

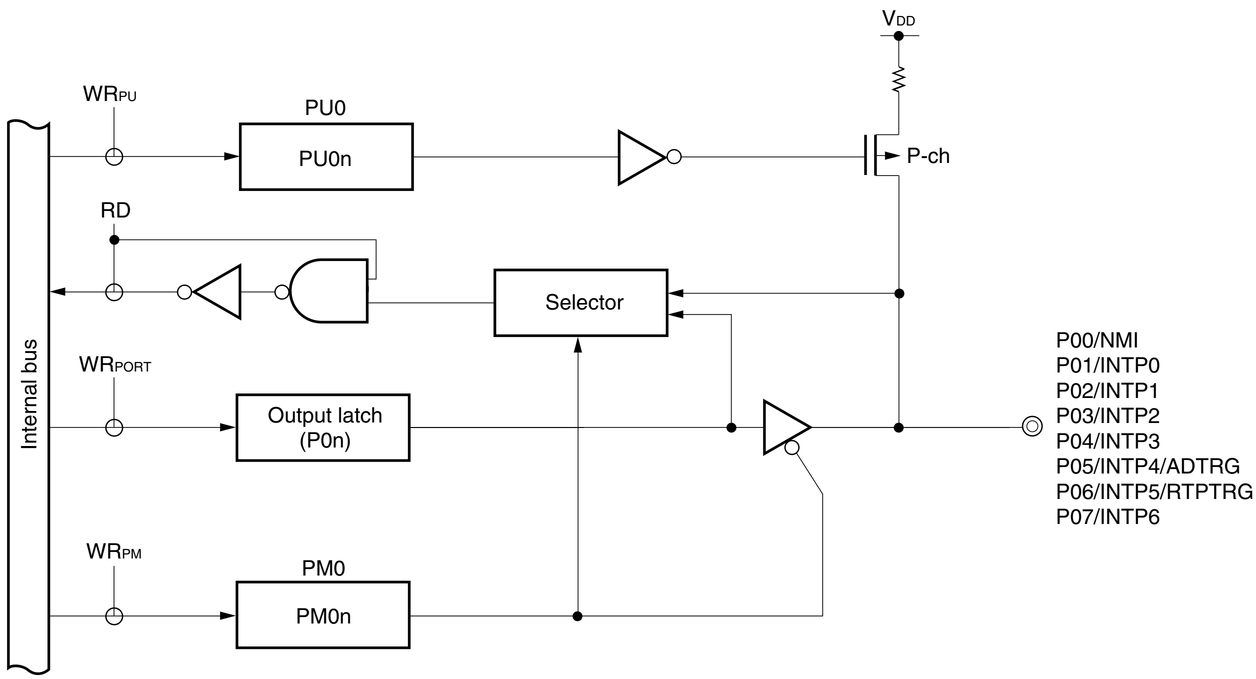
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
EGN0	EGN07	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00

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: Port 0 read signal

WR: Port 0 write signal

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 the P1 register 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 the P1 register is read, the P1 register's 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. SDA0 and SCL0 pins are available only in the Y versions (products with on-chip I^2C).

Table 14-3. Port 1 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 1	P10	SI0/SDA0	I/O	Yes	Selectable as N-ch open-drain output
	P11	SO0			
	P12	SCK0/$\overline{\text{SCL0}}$			
	P13	SI1/RXD0			–
	P14	SO1/TXD0			Selectable as N-ch open-drain output
	P15	SCK1/$\overline{\text{ASCK0}}$			

Note Software pull-up function

(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 the port 1 register (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 the P1 register. Also, the P1 register (output latch) values can be read by reading the P1 register 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 the P1 and PM1 registers 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 8-bit or 1-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 8-bit or 1-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 8-bit or 1-bit units.

After reset: 00H R/W Address: FFFFF0A2H

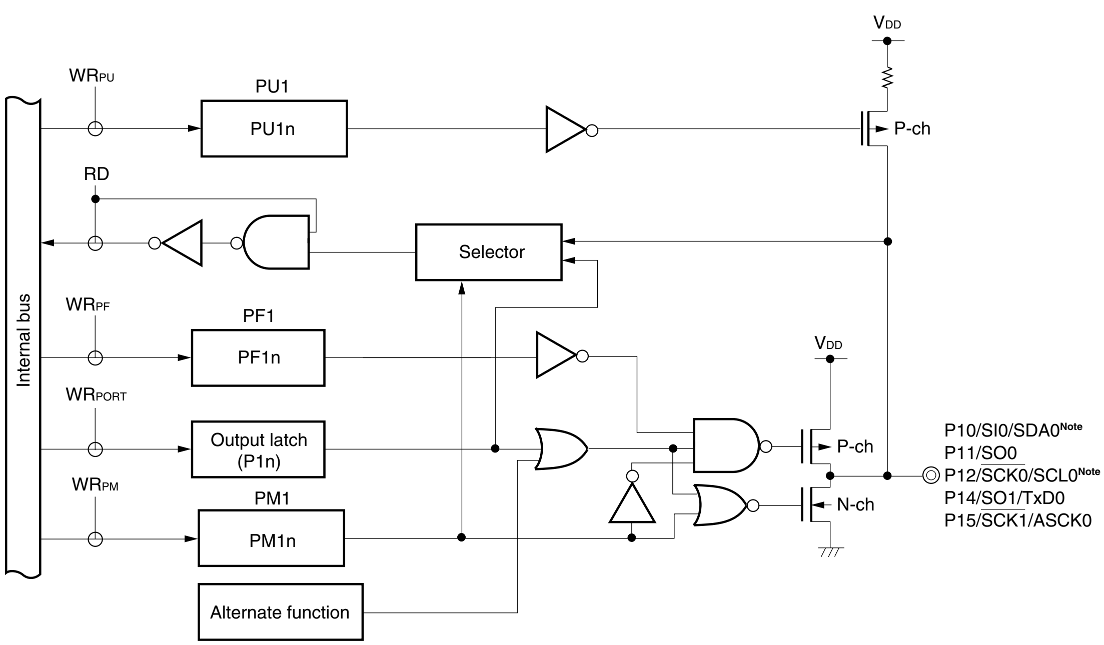
	7	6	5	4	3	2	1	0
PF1	0	0	PF15	PF14	0Note	PF12	PF11	PF10

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 as a normal output.

(3) Block diagram (Port 1)

Figure 14-2. Block Diagram of P10 to P12, P14, and P15

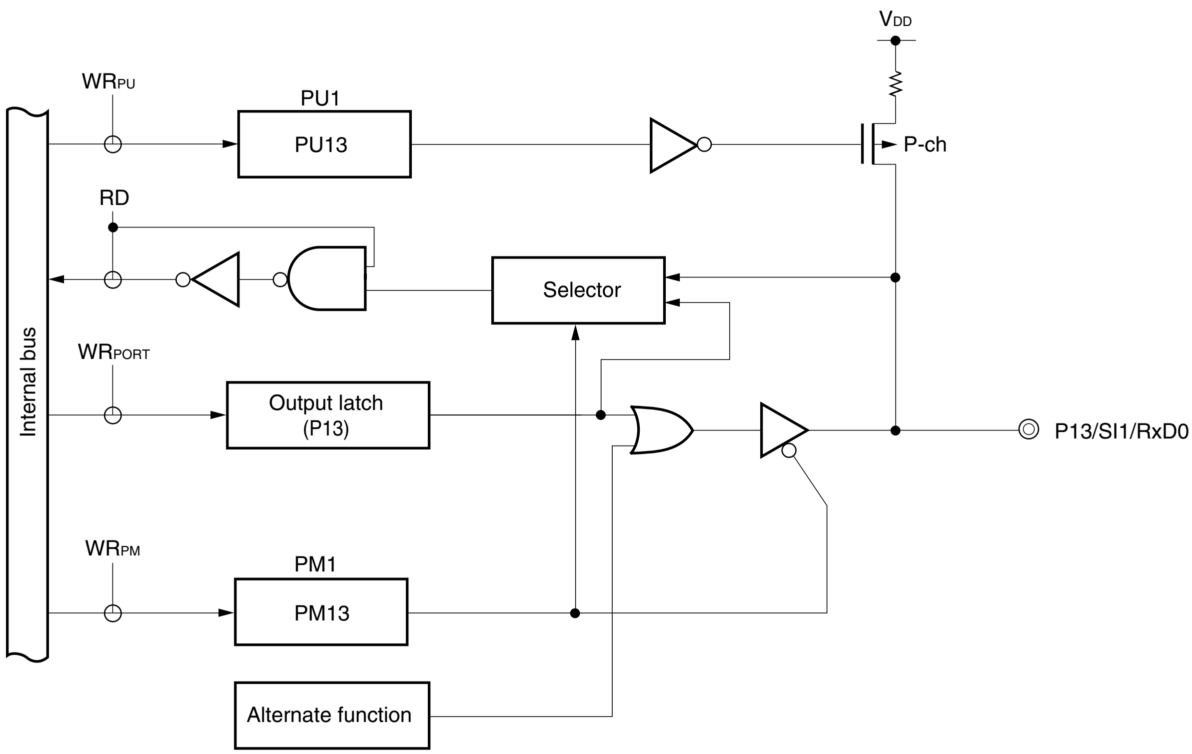


Note The SDA0 and SCL0 pins are available only in the Y versions (products with on-chip I^2C).

Remarks 1. PU1: Pull-up resistor option register 1
PF1: Port 1 function register
PM1: Port 1 mode register
RD: Port 1 read signal
WR: Port 1 write signal

2. n = 0 to 2, 4, 5

Figure 14-3. Block Diagram of P13



Remark PU1: Pull-up resistor option register 1
PM1: Port 1 mode register
RD: Port 1 read signal
WR: Port 1 write signal

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).

P20, P21, P22, P24 and P25 are selectable as normal outputs or N-ch open-drain outputs.

When P26 and P27 are used as the TI2 and 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	Outputs 0
1	Outputs 1

Remark In input mode: When the P2 register 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 the P2 register is read, the P2 register's 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. SDA1 and SCL1 are available only in the Y versions (products with on-chip I^2C).

Table 14-4. Port 2 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 2	P20	SI2/SDA1	I/O	Yes	Selectable as N-ch open-drain output
	P21	SO2			
	P22	$\overline{\text{SCK2}}$/SCL1			
	P23	SI3/RXD1			–
	P24	SO3/TXD1			Selectable as N-ch open-drain output
	P25	$\overline{\text{SCK3}}$/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 the port 2 register (P2). The port 2 function register (PF2) can be used to specify whether P20, P21, P22, P24 and P25 are normal outputs or N-ch open-drain outputs.

When using this port in input mode, the pin statuses can be read by reading the P2 register. Also, the P2 register (output latch) values can be read by reading the P2 register 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 as TI2 and TI3 pins, noise elimination is provided by a digital noise eliminator (same as digital noise eliminator for port 0).

Clear the P2 and PM2 registers 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 8-bit or 1-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 8-bit or 1-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 8-bit or 1-bit units.

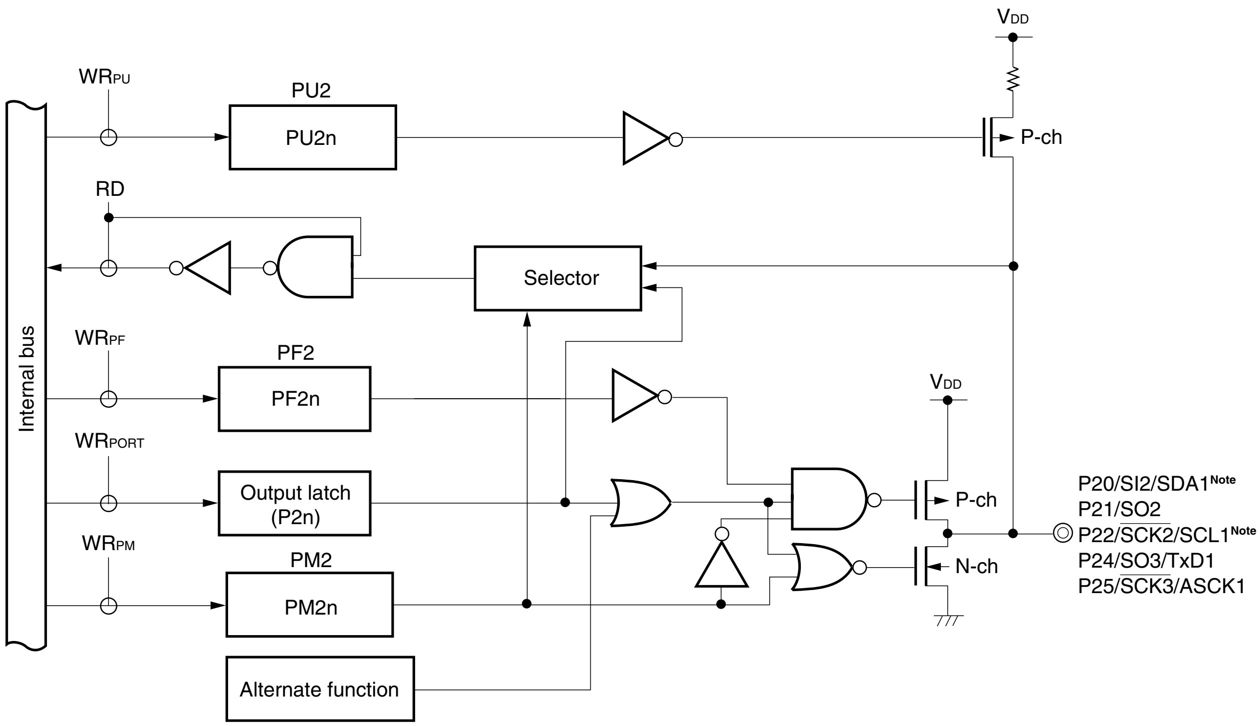
After reset: 00H R/W Address: FFFFF0A4H

	7	6	5	4	3	2	1	0
PF2	0	0	PF25	PF24	0	PF22	PF21	PF20

PF2n	Control of normal output/N-ch open-drain output (n = 0 to 2, 4, 5)
0	Normal output
1	N-ch open-drain output

(3) Block diagram (Port 2)

Figure 14-4. Block Diagram of P20 to P22, P24, and P25



Note The SDA1 and SCL1 pins are available only in the Y versions (products with on-chip I^2C).

Remarks 1. PU2: Pull-up resistor option register 2

PF2: Port 2 function register

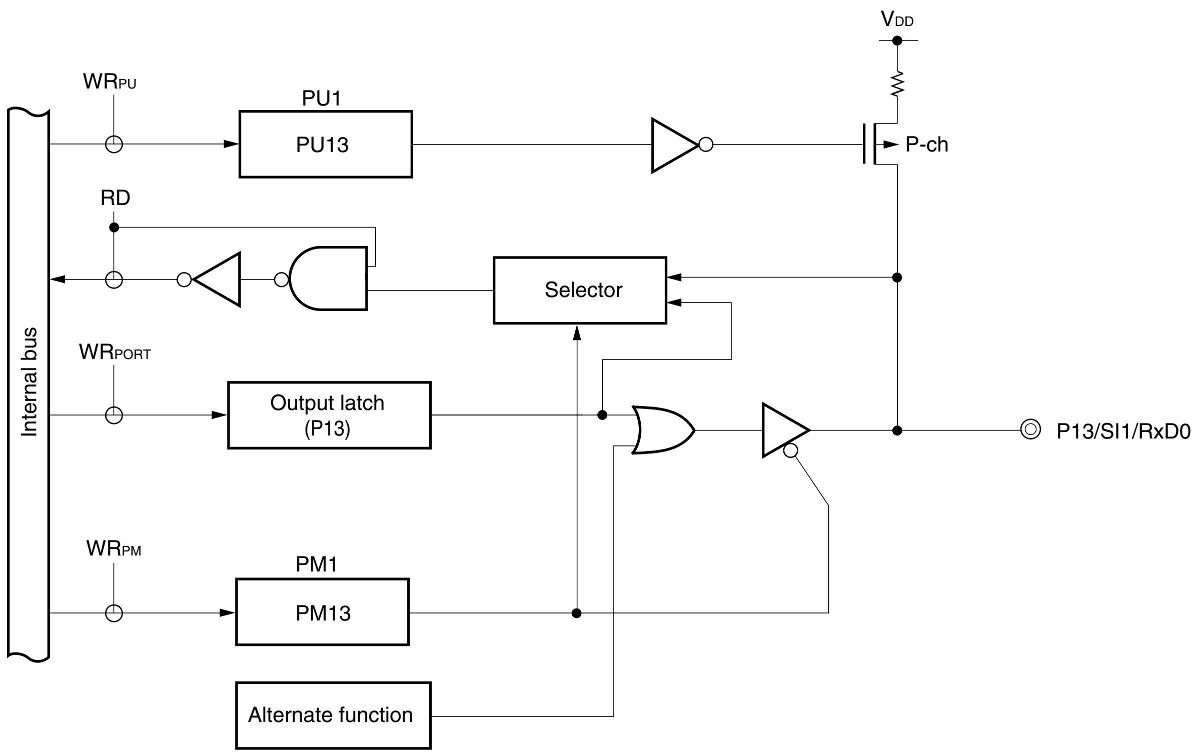
PM2: Port 2 mode register

RD: Port 2 read signal

WR: Port 2 write signal

2. n = 0 to 2, 4, 5

Figure 14-5. Block Diagram of P23, P26, and P27



Remarks 1. PU2: Pull-up resistor option register 2

PM2: Port 2 mode register

RD: Port 2 read signal

WR: Port 2 write signal

2. n = 3 , 6, or 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).
Either a normal output or N-ch open-drain out can be selected for P33 and P34.
When using P36 and P37 as the TI4 and TI5 pins, noise is eliminated by the 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 the P3 register 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 the P3 register is read, the P3 register's 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. Port 3 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 3	P30	TI00	I/O	Yes	–
	P31	TI01			
	P32	TI10/SI4			
	P33	TI11/SO4			Selectable as N-ch open-drain output.
	P34	TO0/A13/$\overline{SCK4}$			
	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 the port 3 register (P3). The port 3 function register (PF3) can be used to specify whether P33 and P34 are normal outputs or N-ch open-drain outputs.

When using this port in input mode, the pin statuses can be read by reading the P3 register. Also, the P3 register (output latch) values can be read by reading the P3 register 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 the PM3 registers (PM34, PM35, PM36) and the P3 registers (P34, P35, P36) to 0.

Clear the P3 and PM3 registers 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 8-bit or 1-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 8-bit or 1-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

(c) Port 3 function register (PF3)

PF3 can be read/written in 8-bit or 1-bit units.

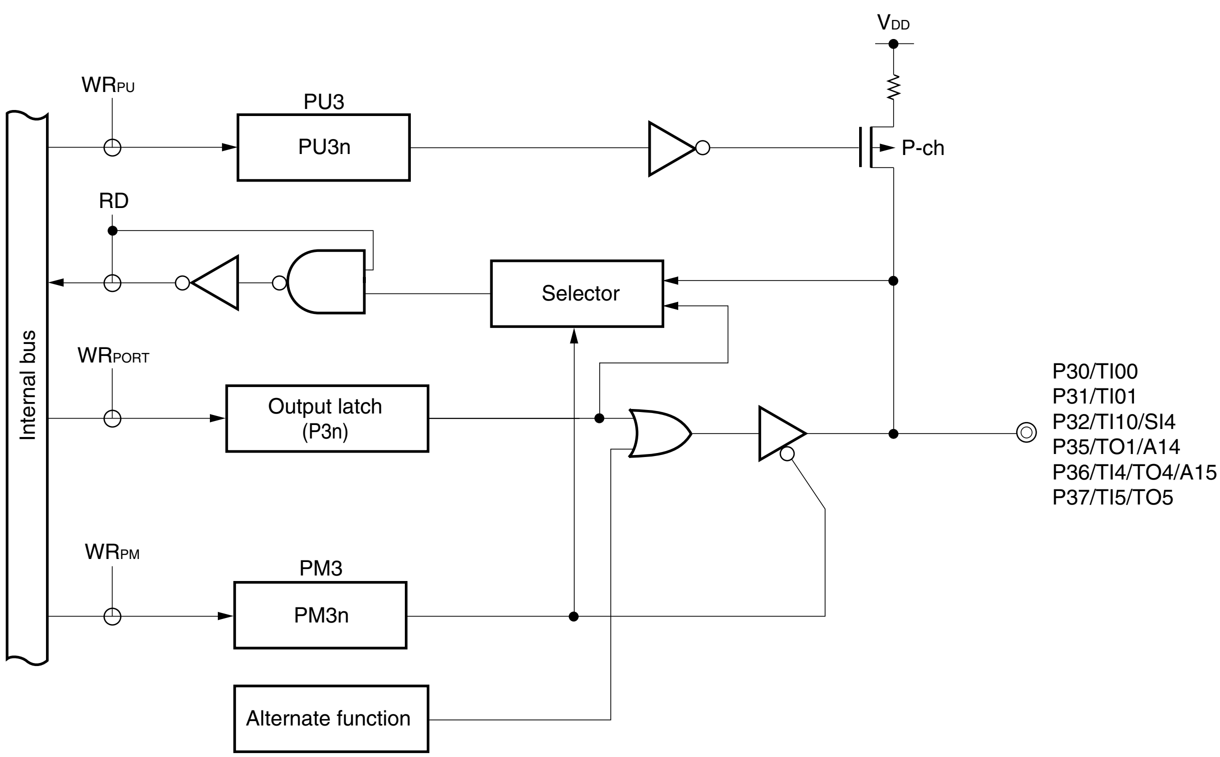
After reset: 00H R/W Address: FFFFF0A6H

	7	6	5	4	3	2	1	0
PF3	0	0	0	PF34	PF33	0	0	0

PF3n	Control of normal output/N-ch open-drain output (n = 3, 4)
0	Normal output
1	N-ch open-drain output

(3) Block diagram (Port 3)

Figure 14-6. Block Diagram of P30 to P32 and P35 to P37



Remarks 1. PU3: Pull-up resistor option register 3

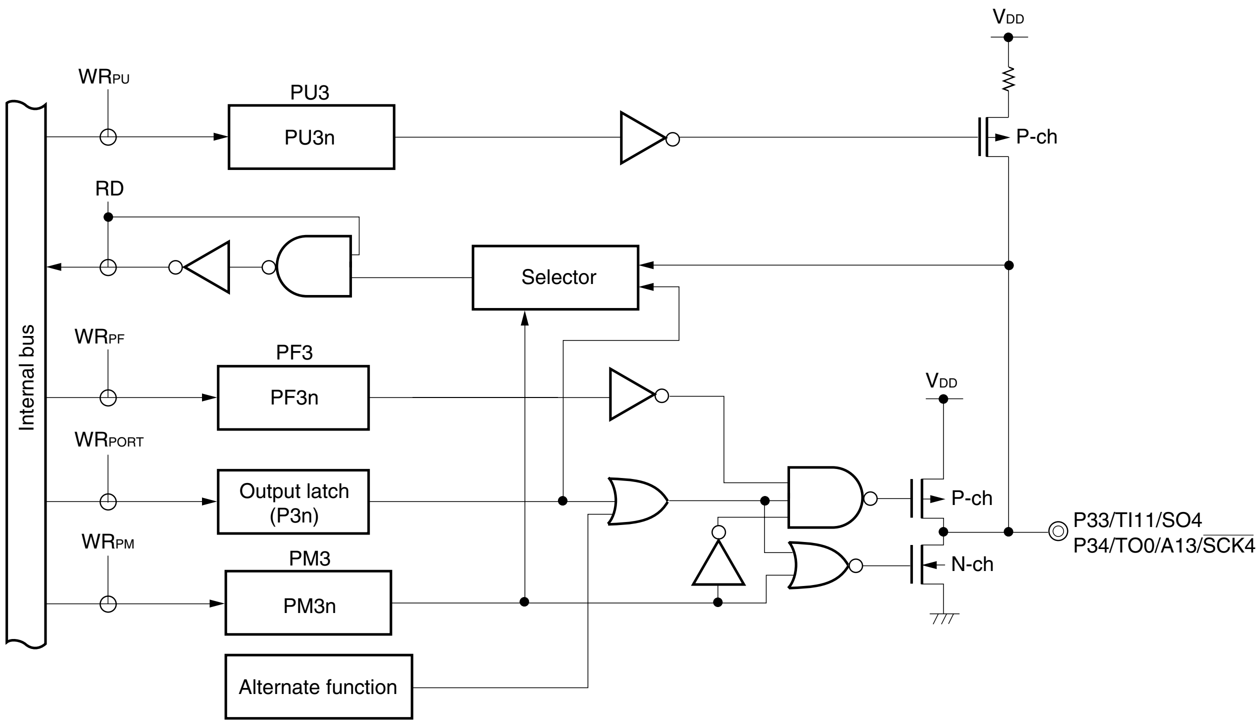
PM3: Port 3 mode register

RD: Port 3 read signal

WR: Port 3 write signal

2. n = 0 to 2, 5 to 7

Figure 14-7. Block Diagram of P33 and P34



Remarks 1. PU3: Pull-up resistor option register 3
RF3: Port 3 function register
PM3: Port 3 mode register
RD: Port 3 read signal
WR: Port 3 write signal

2. n = 3, 4

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	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 the P4 and P5 registers 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 the P4 and P5 registers 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 Function Pins 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) the and port 5 mode register (PM5).

In output mode, the values set to each bit are output to the port 4 and 5 registers (P4 and P5).

When using these ports in input mode, the pin statuses can be read by reading the P4 and P5 registers. Also, the P4 and P5 register (output latch) values can be read by reading the P4 and P5 registers while in output mode.

A software pull-up function is not implemented.

When using the alternate function as AD0 to AD15, set the pin functions via the memory expansion register (MM). This does not affect the PM4 and PM5 registers.

When a reset is input, the settings are initialized to input mode.

(2) Control register

(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.

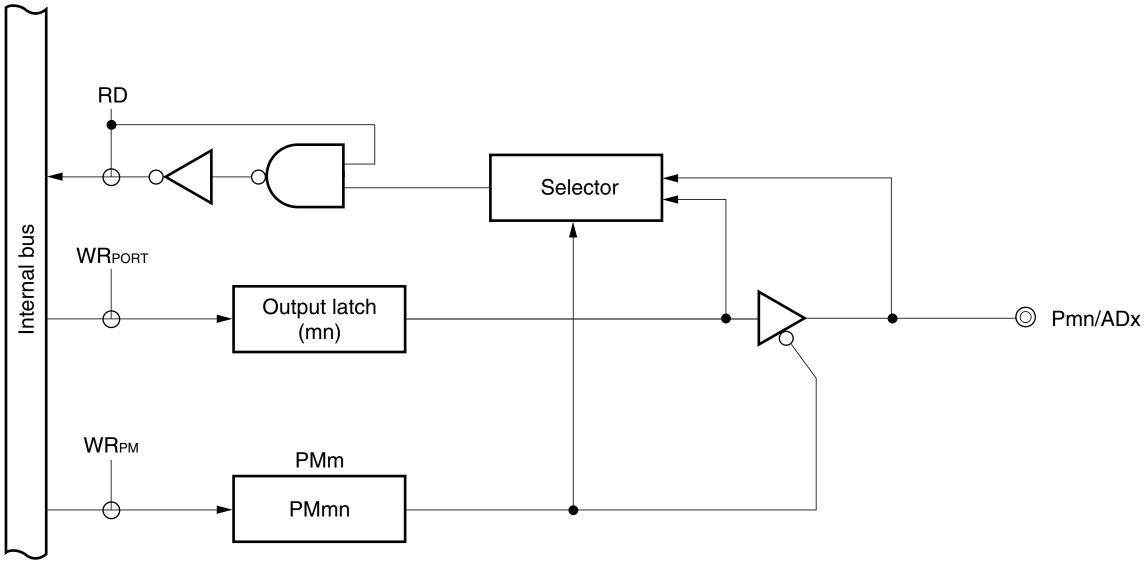
After reset: FFH R/W Address: FFFFF028H, FFFFF02AH

	7	6	5	4	3	2	1	0
PMn (n = 4, 5)	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

PMnx	Control of I/O mode (n = 4, 5, x = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagram (Ports 4 and 5)

Figure 14-8. Block Diagram of P40 to P47 and P50 to P57



Remarks 1. PMm: Port m mode register

RD: Port m read signal

WR: Port m write signal

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	Outputs 0
1	Outputs 1

Remark In input mode: When the P6 register 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 the P6 register is read, the P6 register's 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. Port 6 Alternate Function Pins

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 the port 6 register (P6).

When using this port in input mode, the pin statuses can be read by reading the P6 register. Also, the P6 register (output latch) values can be read by reading the P6 register 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 register (MM). This does not affect the PM6 register.

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 8-bit or 1-bit units.

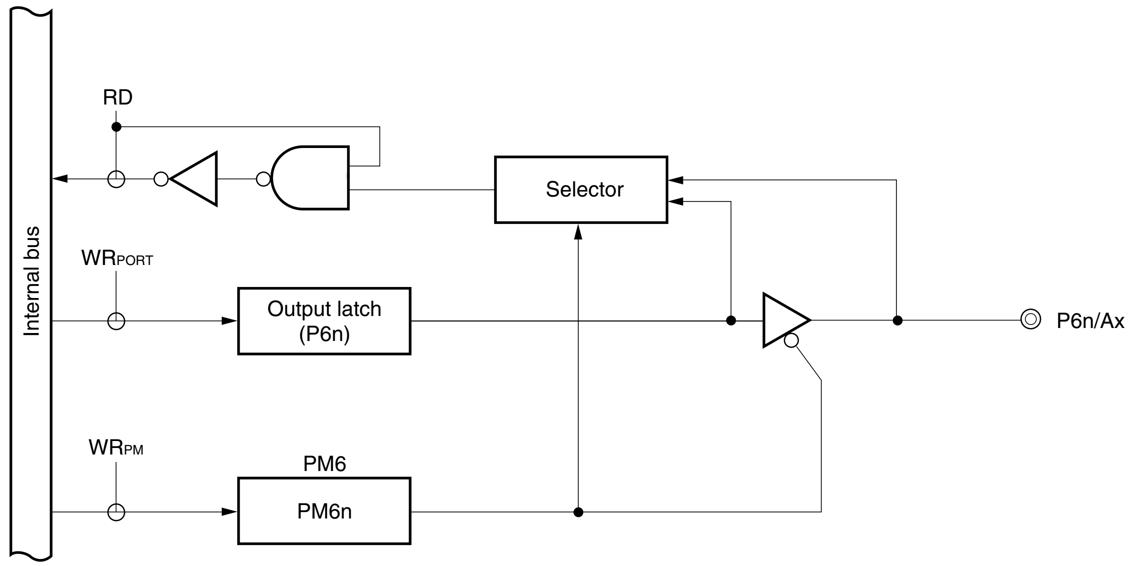
After reset: 3FH R/W Address: FFFFF02CH

	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 14-9. Block Diagram P60 to P65



Remarks 1. PM6: Port 6 mode register
RD: Port 6 read signal
WR: Port 6 write signal

2. n = 0 to 5
x = 16 to 21

14.2.7 Ports 7 and 8

Port 7 is an 8-bit input port and port 8 is a 4-bit input 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 Function Pins 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.

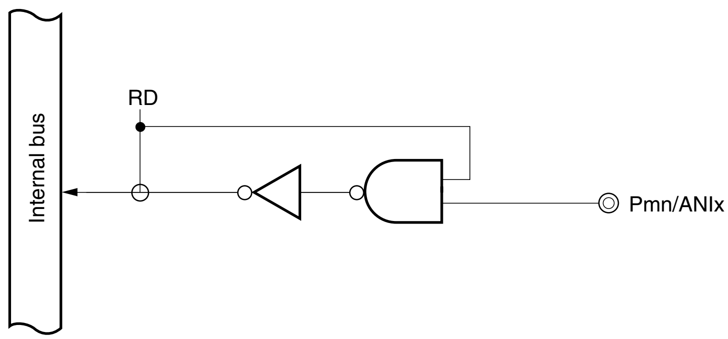
The pin statuses can be read by reading the port 7 and 8 registers (P7 and P8). Data cannot be written to P7 or P8.

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 (Ports 7 and 8)

Figure 14-10. Block Diagram of P70 to P77 and P80 to P83



Remarks 1. RD: Port 7, port 8 read signals

2. m = 7, 8

n = 0 to 7 (m = 7), 0 to 3 (m = 8)

x = 0 to 7 (m = 7), 8 to 11 (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 the P9 register 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 the P9 register is read, the P9 register's 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. Port 9 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 9	P90	$\overline{\text{LBEN}}$/$\overline{\text{WRL}}$	I/O	No	–
	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	$\overline{\text{HLDRQ}}$			

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 the port 9 register (P9).

When using this port in input mode, the pin statuses can be read by reading the P9 register. Also, the P9 register (output latch) values can be read by reading the P9 register while in output mode.

A software pull-up function is not implemented.

When using the P9 for control signals in expansion mode, 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.

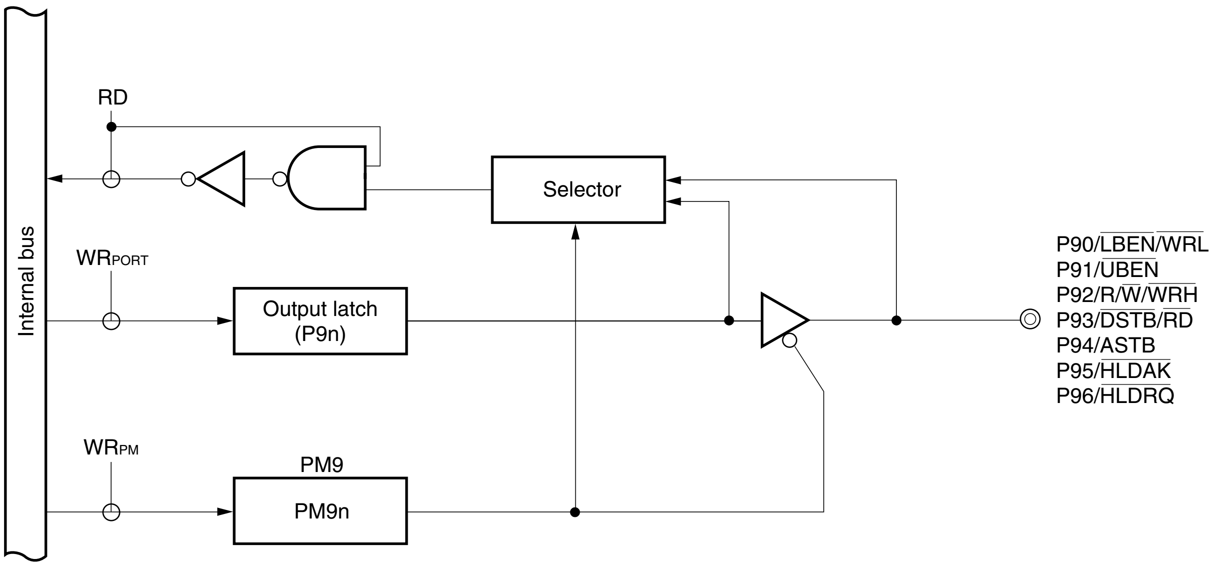
After reset: 7FH R/W Address: FFFFF032H

	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 diagram (Port 9)

Figure 14-11. Block Diagram of P90 to P96



Remarks 1. PM9: Port 9 mode register
RD: Port 9 read signal
WR: Port 9 write signal

2. n = 0 to 6

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.
When using P100 to P107 as KR0 to KR7 pins, noise is eliminated by the analog noise eliminator.

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 the P10 register 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 the P10 register is read, the P10 register's 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. $\overline{\text{IERX}}$ and $\overline{\text{IETX}}$ pins are valid only in the V850/SB2.

Table 14-10. Port 10 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 10	P100	RTP0/A5/KR0	I/O	Yes	Selectable as N-ch open-drain outputs Analog noise elimination
	P101	RTP1/A6/KR1			
	P102	RTP2/A7/KR2			
	P103	RTP3/A8/KR3			
	P104	RTP4/A9/KR4/$\overline{\text{IERX}}$			
	P105	RTP5/A10/KR5/$\overline{\text{IETX}}$			
	P106	RTP6/A11/KR6			
	P107	RTP7/A12/KR7			

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 the port 10 register (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 the P10 register. Also, the P10 register (output latch) values can be read by reading the P10 register 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, see the pin functions via the memory address output mode register (MAM). At this time, be sure to set P10 and PM10 to 0.

When used as the KR0 to KR7 pins, noise is eliminated by the analog noise eliminator.

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 register

(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 8-bit or 1-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 8-bit or 1-bit units.

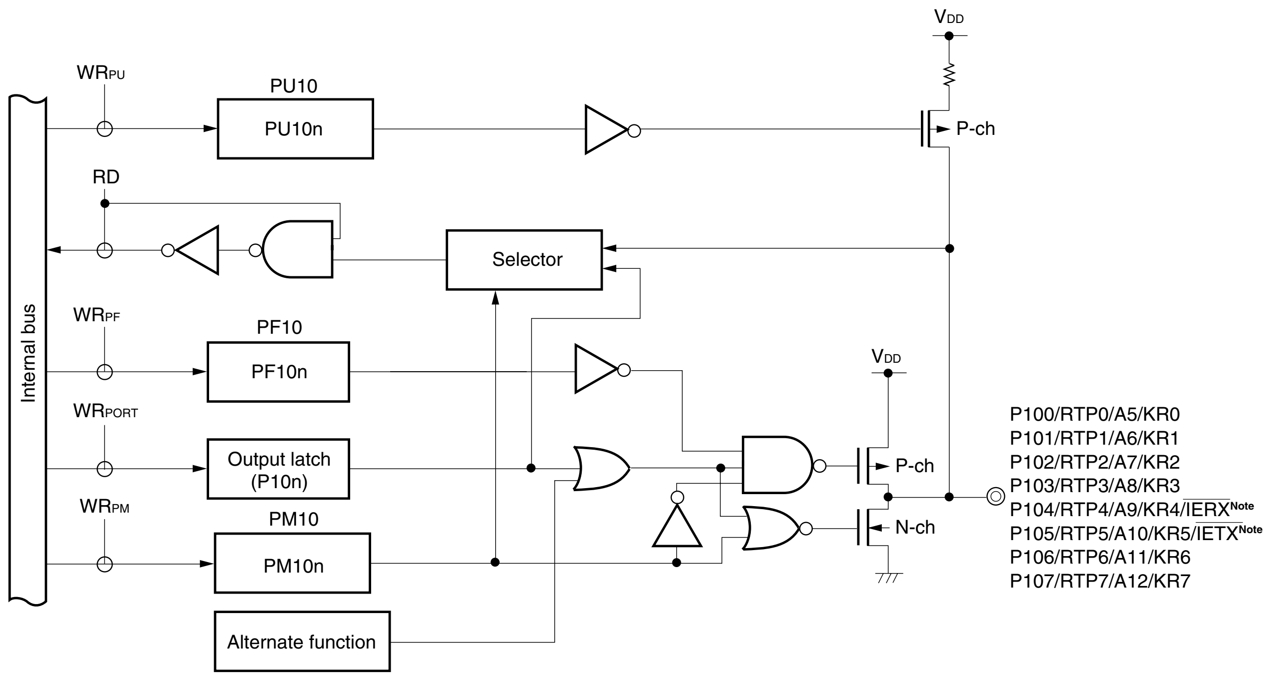
After reset: 00H R/W Address: FFFFF0B4H

	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 14-12. Block Diagram of P100 to P107



Note The $\overline{\text{IERX}}$, $\overline{\text{IETX}}$ pins apply only to the V850/SB2.

Remarks 1. PU10: Pull-up resistor option register 10

RF10: Port 10 function register

PM10: Port 10 mode register

RD: Port 10 read signal

WR: Port 10 write signal

2. n = 0 to 7

14.2.10 Port 11

Port 11 is a 4-bit port. 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 8-bit or 1-bit units.
The on/off of wait function can be switched with a port alternate-function control register (PAC).

Caution When using the wait function, set BC_{DD} to the same potential as EV_{DD}.

After reset: 00H R/W Address: FFFFF016H

	7	6	5	4	3	2	1	0
P11	0	0	0	Undefined	P113	P112	P111	P110

P11n	Control of output data (in output mode) (n = 0 to 3)
0	Output 0
1	Output 1

Remark In input mode: When the P11 register 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 the P11 register is read, the P11 register's values are read. Writing to P11 writes the values to that register, and those values are immediately output.

Port 11 includes the following alternate functions.

Table 14-11. Port 11 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 11	P110	A1/$\overline{\text{WAIT}}$	I/O	Yes	–
	P111	A2			
	P112	A3			
	P113	A4			

Note Software pull-up function

(1) Function of P11 pins

Port 11 is a 4-bit (total) 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 the port register (P11).

When using this port in input mode, the pin statuses can be read by reading the P11 register. Also, the P11 register (output latch) values can be read by reading the P11 register 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).

The on/off of wait function can be switched with a port-alternate function control register (PAC).

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 clear P11 and PM11 to 0.

When a reset is input, the settings are initialized to input mode.

Caution A wait function generated by the $\overline{\text{WAIT}}$ pin cannot be used while a separate bus is being used. However, a programmable wait is possible.

(2) Control register

(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 8-bit or 1-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

(c) Port alternate-function control register (PAC)

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

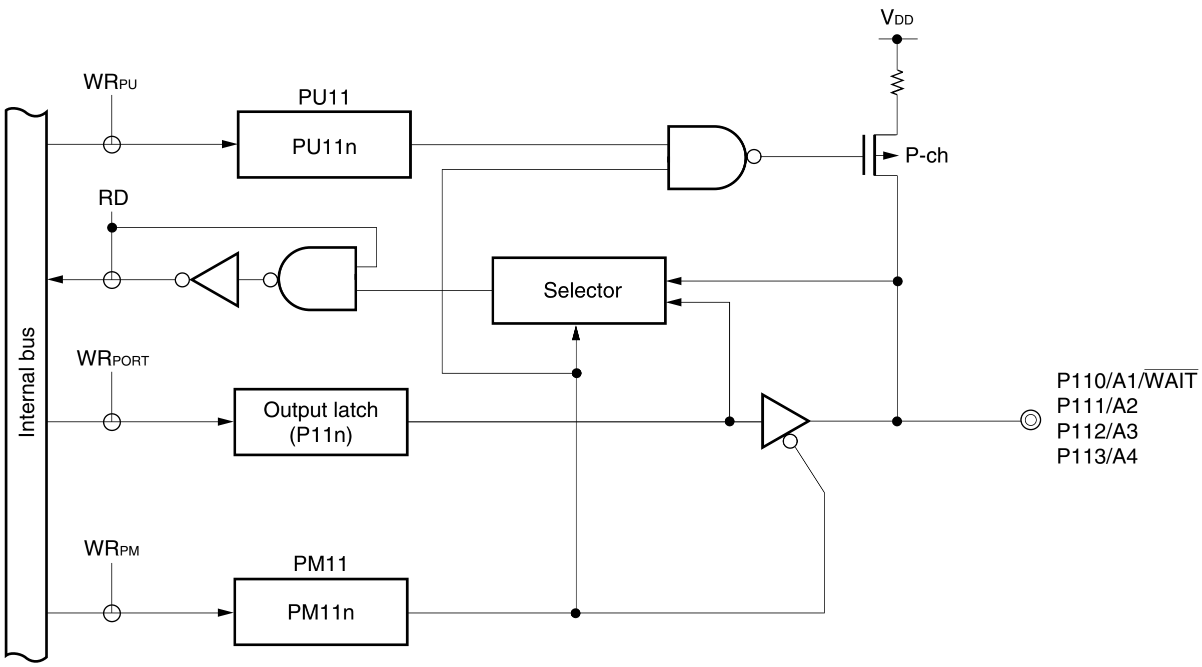
After reset: 00H R/W Address: FFFFF040H

	7	6	5	4	3	2	1	<0>
PAC	0	0	0	0	0	0	0	WAC

P120	Control of output data (in output mode)
0	Wait function off
1	Wait function on

(3) Block diagram (Port 11)

Figure 14-13. Block Diagram of P110 to P113



Remarks 1. PU11: Pull-up resistor option register 11

PM11: Port 11 mode register

RD: Port 11 read signal

WR: Port 11 write signal

2. n = 0 to 3

14.3 Setting When Port Pin Is Used as Alternate Function

When a port pin is used as an alternate function, set the port n mode register (PM0 to PM6 and PM9 to PM11) and output latch as shown in Table 14-12 below.

Table 14-12. Setting When Port Pin Is Used as Alternate Function (1/4)

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	—
★	SDA0[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	
★	SCL0[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	P12 = 0	
	ASCK0	Input	PM15 = 1	Setting not needed for P15	

Note Available only in the Y versions (products with on-chip I^2C)

Table 14-12. Setting When Port Pin Is Used as Alternate Function (2/4)

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	—
★	SDA1[Note]	I/O	PM20 = 0	P20 = 0	PF20 = 1
P21	SO2	Output	PM21 = 0	P21 = 0	—
P22	$\overline{SCK2}$	Input	PM22 = 1	Setting not needed for P22	—
		Output	PM22 = 0	P22 = 0	
★	SCL1[Note]	I/O			PF22 = 1
P23	SI3	Input	PM23 = 1	Setting not needed for P23	—
	RXD1	Input			
P24	SO3	Output	PM24 = 0	P24 = 0	—
	TXD1	Output			
P25	$\overline{SCK3}$	Input	PM25 = 1	Setting not needed for P25	—
		Output	PM25 = 0	P25 = 0	
	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	—
	SI4	Input			
P33	TI11	Input	PM33 = 1	Setting not needed for P33	—
	SO4	Output	PM33 = 0	P33 = 0	
P34	TO0	Output	PM34 = 0	P34 = 0	—
	A13	Output			Refer to **3.4.6 (2)** (MAM)
	$\overline{SCK4}$	Input	PM34 = 1	Setting not needed for P34	—
		Output	PM34 = 0	P34 = 0	

Note Available only in the Y versions (products with on-chip I^2C)

Table 14-12. Setting When Port Pin Is Used as Alternate Function (3/4)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
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)
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 = 1	Refer to **3.4.6 (1)** (MM)
P96	$\overline{\text{HLDRQ}}$	Input	Setting not needed for PM96	Setting not needed for P96 = 1	Refer to **3.4.6 (1)** (MM)

Table 14-12. Setting When Port Pin Is Used as Alternate Function (4/4)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P100 to P103	RTP0 to RTP3	Output	PM100 to PM103 = 0	P100 to P103 = 0	–
	A5 to A8	Output			Refer to **3.4.6 (2)** (MAM)
	KR0 to KR3	Input	PM100 to PM103 = 1	Setting not needed for P100 to P103	–
P104	RTP4	Output	PM104 = 0	P104 = 0	–
	A9	Output			Refer to **3.4.6 (2)** (MAM)
	KR4	Input	PM104 = 1	Setting not needed for P104	–
	$\overline{\text{IERX}}$ Note	Input			–
P105	RTP5	Output	PM105 = 0	P105 = 0	–
	A10	Output			Refer to **3.4.6 (2)** (MAM)
	KR5	Input	PM105 = 1	Setting not needed for P105	–
	$\overline{\text{IETX}}$ Note	Output	PM105 = 0	P105 = 0	–
P106, P107	RTP6, RTP7	Output	PM106, PM107 = 0	P106, P107 = 0	–
	A11, A12	Output			Refer to **3.4.6 (2)** (MAM)
	KR6, KR7	Input	PM106, PM107 = 1	Setting not needed for P106 and P107	–
P110	A1	Output	PM110 = 0	P110 = 0	Refer to **3.4.6 (2)** (MAM)
	$\overline{\text{WAIT}}$	Input	PM110 = 1	Setting not needed for P110	WAC = 1 (PAC)
P111 to P113	A2 to A4	Output	PM111 to PM113 = 0	P111 to P113 = 0	Refer to **3.4.6 (2)** (MAM)

Note Only in the V850/SB2.

Caution When changing the output level of port 0 by setting the 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.

Remark PMnx bit of PMn register and Pnx bit of Pn register

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 3)

★ ## 14.4 Port Function Operation

Port operation differs according to the input/output mode setting, as follows.

14.4.1 Write operation to I/O port

(1) In output mode

A value is written to the output latch using a transfer instruction, and the contents of the output latch are output from the pin.

Once data has been written to the output latch, it is held until the next data is written to the output latch.

(2) In input mode

A value is written to the output latch using a transfer instruction. However, since the output buffer is OFF, the pin status does not change.

Once data has been written to the output latch, it is held until the next 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 Read operation from I/O port

(1) In output mode

The output latch contents are read using a transfer instruction. The output latch contents remain unchanged.

(2) In input mode

The pin status is read using a transfer instruction. The output latch contents remain unchanged.

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.

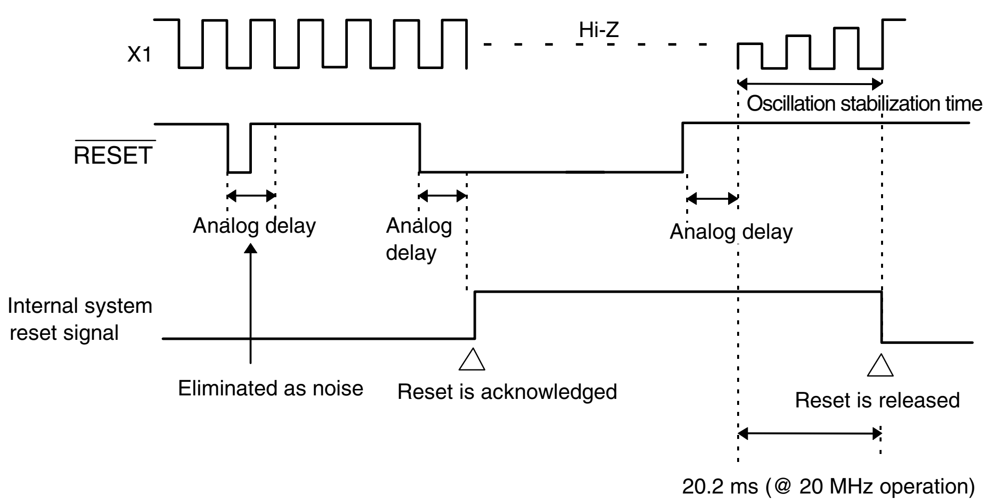
An on-chip noise eliminator uses analog delay to prevent noise-related malfunction at the $\overline{\text{RESET}}$ pin.

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, REGC, AV_{REF}, V_{DD}, V_{SS}, AV_{DD}, AV_{SS}, BV_{DD}, BV_{SS}, EV_{DD}, EV_{SS}, and V_{PP}/IC).

Accordingly, if connected to an external memory device, be sure to attach a pull-up (or pull-down) resistor for each pin. 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 REGULATOR

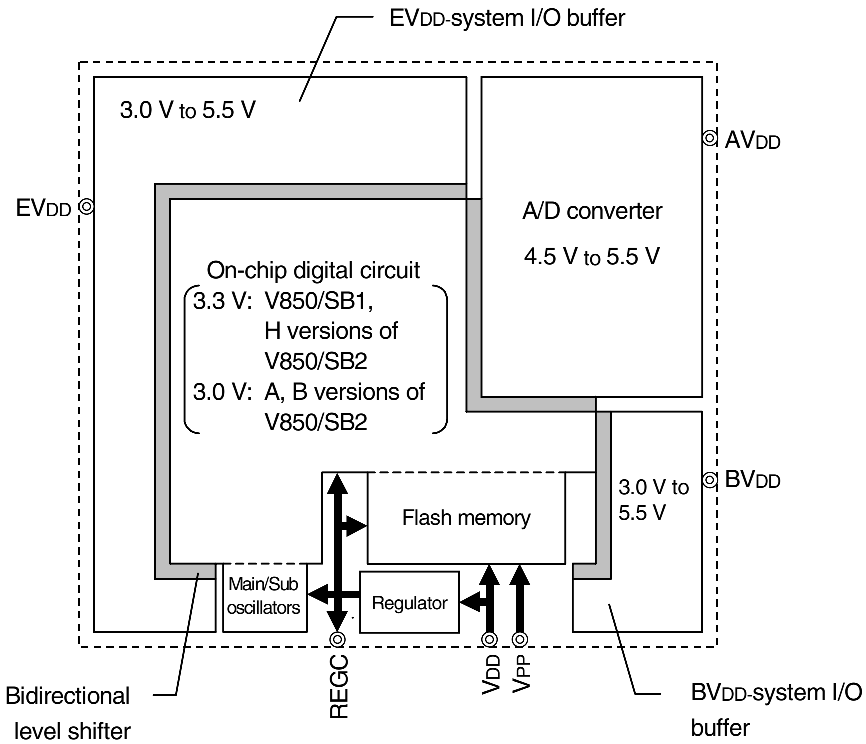
16.1 Outline

The V850/SB1 and V850/SB2 incorporate a regulator to realize a 5 V single power supply, low power consumption, and to reduce noise.

This regulator supplies a voltage obtained by stepping down the V_{DD} power supply voltage to the oscillation blocks and on-chip logic circuits (excluding the A/D converter and output buffers). The regulator output voltage is set to 3.3 V (V850/SB1, H versions of V850/SB2) or 3.0 V (A and B versions of V850/SB2).

Refer to **2.4 I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins** for the power supply corresponding to each pin.

Figure 16-1. Regulator



16.2 Operation

The regulators of the V850/SB1 and V850/SB2 operate in every mode (STOP, IDLE, HALT).

For stabilization of regulator outputs, connect an electrolytic capacitor of about 1 μF to the REGC pin.

CHAPTER 17 ROM CORRECTION FUNCTION

17.1 General

The ROM correction function provided in the V850/SB1 and V850/SB2 is a function that replaces part of a program in the mask ROM with a program in the internal RAM.

First, the instruction of the address where the program replacement should start is replaced with the JMP r0 instruction and the program is instructed to jump to 00000000H. The correction request register (CORRQ) is then checked. At this time, if the CORRQn flag is set to 1, program control shifts to the internal RAM after jumping to the internal ROM area by an instruction such as a jump instruction.

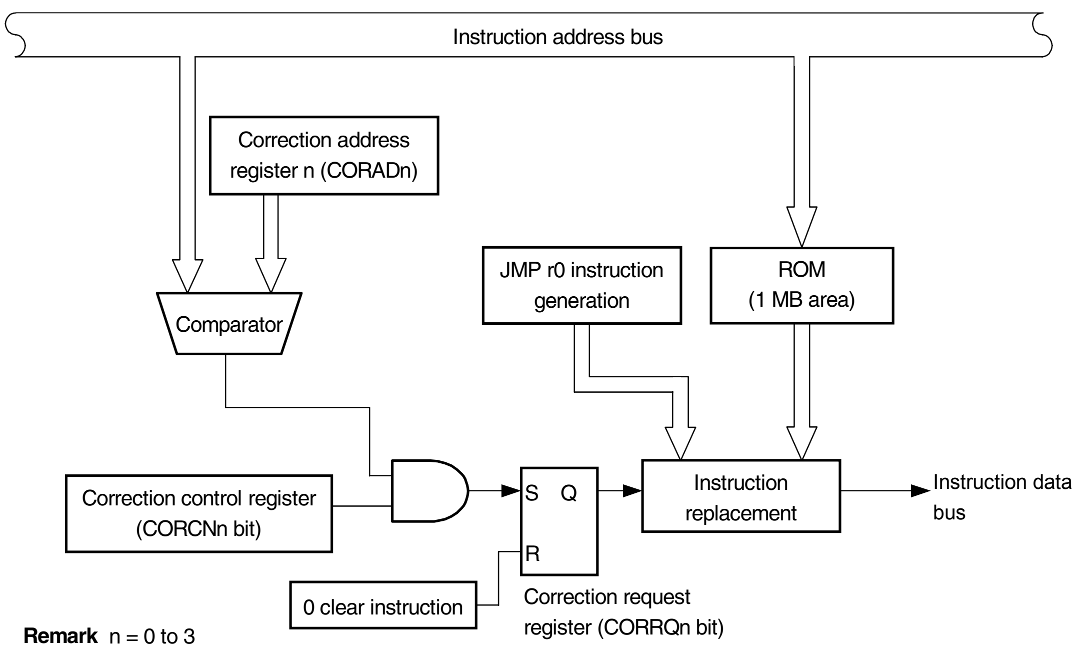
Instruction bugs found in the mask ROM can be avoided, and program flow can be changed by using the ROM correction function.

Up to four correction addresses can be specified.

Cautions 1. The ROM correction function cannot be used for the data in the internal ROM; it can only be used for instruction codes. If ROM correction is carried out on data, that data will replace the instruction code of the JMP r0 instruction.

2. ROM correction for the instructions that access the registers CORCN, CORRQ, or CORAD0 to CORAD3 is prohibited.

Figure 17-1. Block Diagram of ROM Correction



Remark n = 0 to 3

17.2 ROM Correction Peripheral I/O Registers

(1) Correction control register (CORCN)

CORCN controls whether or not the instruction of the correction address is replaced with the JMP r0 instruction when the correction address matches the fetch address (n = 0 to 3).

Whether match detection by a comparator is enabled or disabled can be set for each channel.

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

After reset: 00H R/W Address: FFFFF36CH

	7	6	5	4	<3>	<2>	<1>	<0>
CORCN	0	0	0	0	COREN3	COREN2	COREN1	COREN0

CORENn	CORADn register and fetch address match detection control
0	Match detection disabled (not detected)
1	Match detection enabled (detected)

Remark n = 0 to 3

(2) Correction request register (CORRQ)

CORRQ saves the channel in which ROM correction occurred. The JMP r0 instruction makes the program jump to 00000000H after the correction address matches the fetch address. At this time, the program can judge the following cases by reading CORRQ.

- Reset input: CORRQ = 00H
- ROM correction generation: CORRQn bit = 1 (n = 0 to 3)
- Branch to 00000000H by user program: CORRQ = 00H

After reset: 00H R/W Address: FFFFF36EH

	7	6	5	4	<3>	<2>	<1>	<0>
CORRQ	0	0	0	0	CORRQ3	CORRQ2	CORRQ1	CORRQ0

CORRQn	Channel n ROM correction request flag
0	No ROM correction request occurred.
1	ROM correction request occurred.

Remark n = 0 to 3

(3) Correction address registers 0 to 3 (CORAD0 to CORAD3)

CORADn sets the start address of an instruction to be corrected (correction address) in the ROM.

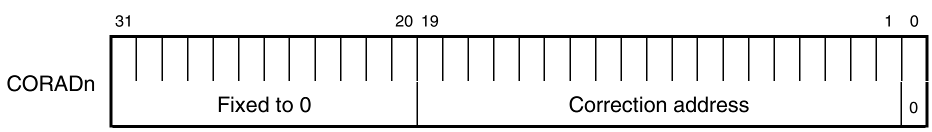
Up to four points of the program can be corrected at once since the V850/SB1 and V850/SB2 have four correction address registers (CORADn) (n = 0 to 3).

Since the ROM capacity varies depending on the product, set the correction address within following ranges.

μPD703031A, 703031AY, 703031B, 703031BY,
703034A, 703034AY, 703034B, 703034BY (128 KB): 00000000H to 0001FFFEH
μPD703033A, 703033AY, 703033B, 703033BY,
703035A, 703035AY, 703035B, 703035BY (256 KB): 00000000H to 0003FFFEH
μPD703030B, 703030BY, 703036H, 703036HY (384 KB): 00000000H to 0005FFFEH
μPD703032A, 703032AY, 703032B, 703032BY,
703037A, 703037AY, 703037H, 703037HY (512 KB): 00000000H to 0007FFFEH

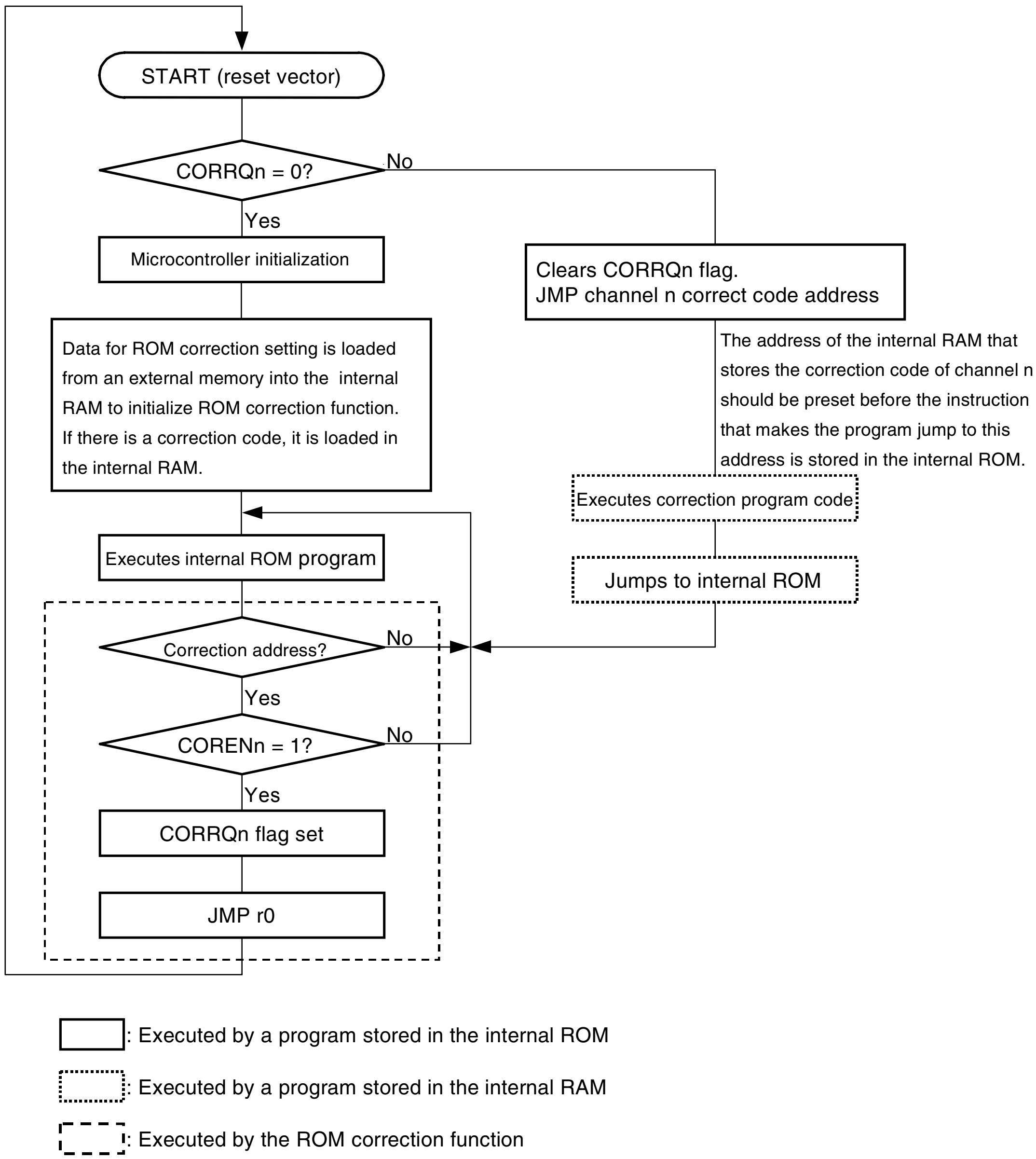
Bits 0 and 20 to 31 should be fixed to 0.

After reset: 00000000H R/W Address: CORAD0: FFFFF370H CORAD2: FFFFF378H
CORAD1: FFFFF374H CORAD3: FFFFF37CH



(n = 0 to 3)

Figure 17-2. ROM Correction Operation and Program Flow



Caution **Check the ROM correction generation from the vector table with a high interrupt level when executing ROM correction during a vector interrupt routine. If an interrupt conflicts with ROM correction, processing is branched to an interrupt vector, where, if ROM correction is being re-executed, CORRQn is set (1) again and multiple CORRQn flags are set (1). The channel for which ROM correction is to be executed is determined by the interrupt level.**

Remark n = 0 to 3

CHAPTER 18 FLASH MEMORY

The following products are the flash memory versions of the V850/SB1 and V850/SB2.

Caution The flash memory versions and mask ROM versions differ in noise immunity and noise radiation. If replacing a flash memory version with a mask ROM version when changing from experimental production to mass production, make a thorough evaluation by using the CS model (not ES model) of the mask ROM version.

★ **(1) V850/SB1**

μPD70F3033A, 70F3033AY, 70F3033B, 70F3033BY:	256 KB flash memory versions
μPD70F3030B, 70F3030BY:	384 KB flash memory versions
μPD70F3032A, 70F3032AY, 70F3032B, 70F3032BY:	512 KB flash memory versions

★ **(2) V850/SB2**

μPD70F3035A, 70F3035AY, 70F3035B, 70F3035BY:	256 KB flash memory versions
μPD70F3036H, 70F3036HY:	384 KB flash memory versions
μPD70F3037A, 70F3037AY, 70F3037H,70F3037HY:	512 KB flash memory versions

In the instruction fetch to this flash memory, 4 bytes can be accessed by a single clock, the same as in the mask ROM version.

Writing to flash memory can be performed with the memory mounted on the target system (on board). A dedicated flash programmer 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/SB1 or V850/SB2 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.

18.1 Features

- 4-byte/1-clock access (in instruction fetch access)
- All area one-shot erase/area unit erase
- Communication via serial interface with the dedicated flash programmer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming
- Flash memory programming in area (128 KB) units by self-writing

18.1.1 Erase unit

The erase unit differs depending on the product.

★ **(1) V850/SB1 (μPD70F3033A, 70F3033AY, 70F3033B, 70F3033BY), V850/SB2 (μPD70F3035A, 70F3035AY, 70F3035B, 70F3035BY)**

The erase 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

★ **(2) V850/SB1 (μPD70F3033B, 70F3033BY), V850/SB2 (μPD70F3036H, 70F3036HY)**

The erase units for 384 KB flash memory versions are shown below.

(a) All area one-shot erase

The area of xx000000H to xx05FFFFH can be erased in one shot.

(b) Area erase

Erasure can be performed in area units (there are three 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
Area 2: The area of xx040000H to xx05FFFFH (128 KB) is erased

★ **(3) V850/SB1 (μPD70F3032A, 70F3032AY, 70F3032B, 70F3032BY), V850/SB2 (μPD70F3037A, 70F3037AY, 70F3037H, 70F3037HY)**

The erase units for 512 KB flash memory versions are shown below.

(a) All area one-shot erase

The area of xx000000H to xx07FFFFH can be erased in one shot.

(b) Area erase

Erasure can be performed in area units (there are four 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
Area 2: The area of xx040000H to xx05FFFFH (128 KB) is erased
Area 3: The area of xx060000H to xx07FFFFH (128 KB) is erased

18.1.2 Write/read time

The write/read time is shown below.

Write time: 50 μs/byte
Read time: 50 ns (cycle time)

18.2 Writing with Flash Programmer

Writing can be performed either on-board or off-board with the dedicated flash programmer.

(1) On-board programming

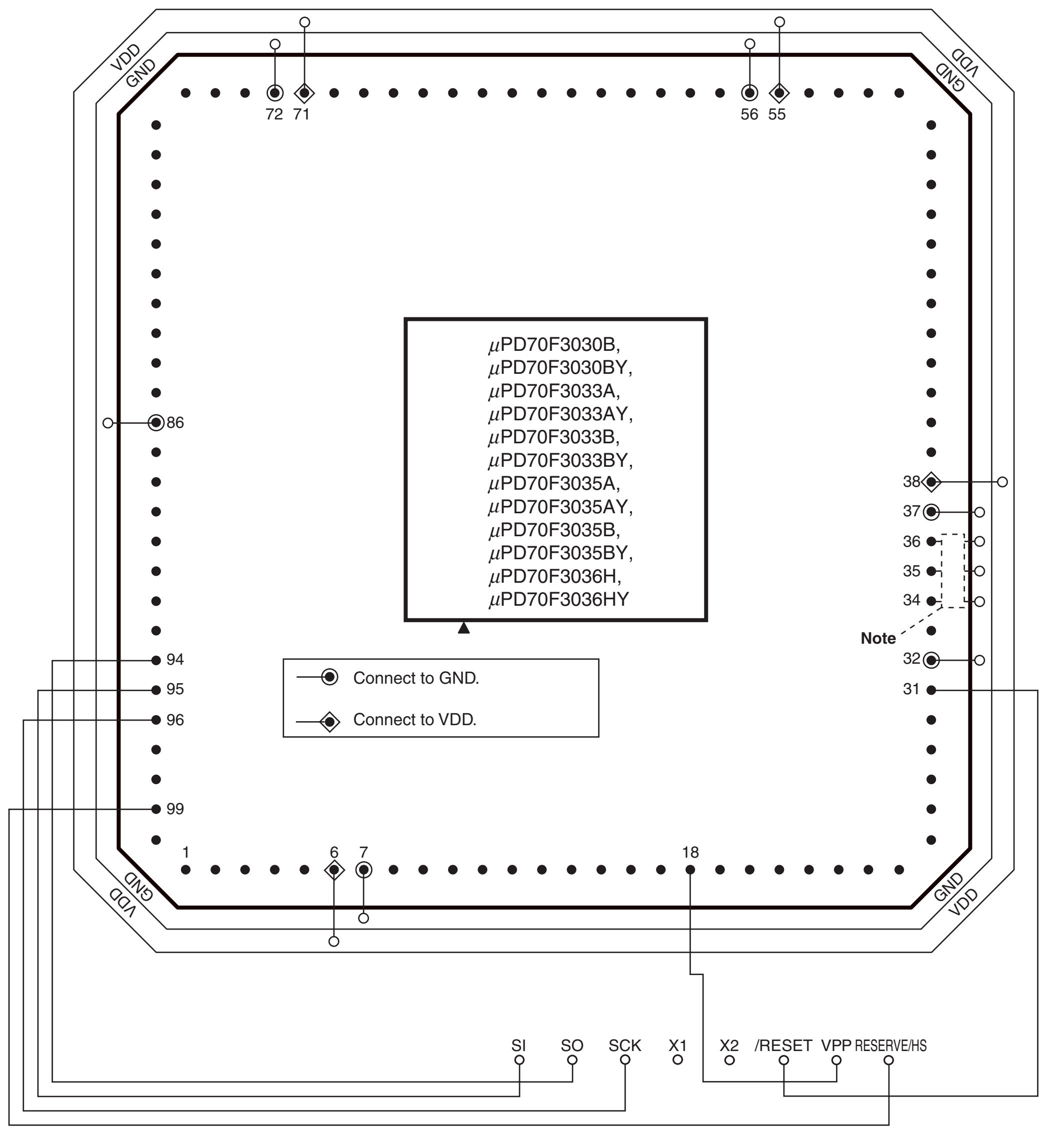
The contents of the flash memory are rewritten after the V850/SB1 or V850/SB2 is mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

(2) Off-board programming

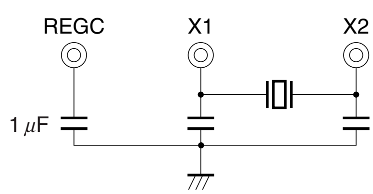
Writing to a flash memory are performed by the dedicated program adapter (FA Series), etc., before mounting the V850/SB1 or V850/SB2 on the target system.

Remark FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

★ **Figure 18-1. Wiring Example of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GC-8EU)**



Note The V850/SB1 and V850/SB2 cannot be supplied with the clock from the CLK pin of the flash programmer (PG-FP3).
Supply the clock by creating an oscillator on the flash writing adapter (broken-line portion).
An example of the oscillator is shown below.



Remarks 1. Handle the pins not described above in accordance with the recommended connection of unused pins (refer to **2.4 Pin I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins**). When connecting to VDD via a resistor, use of a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is for a 100-pin plastic LQFP (fine pitch) package.

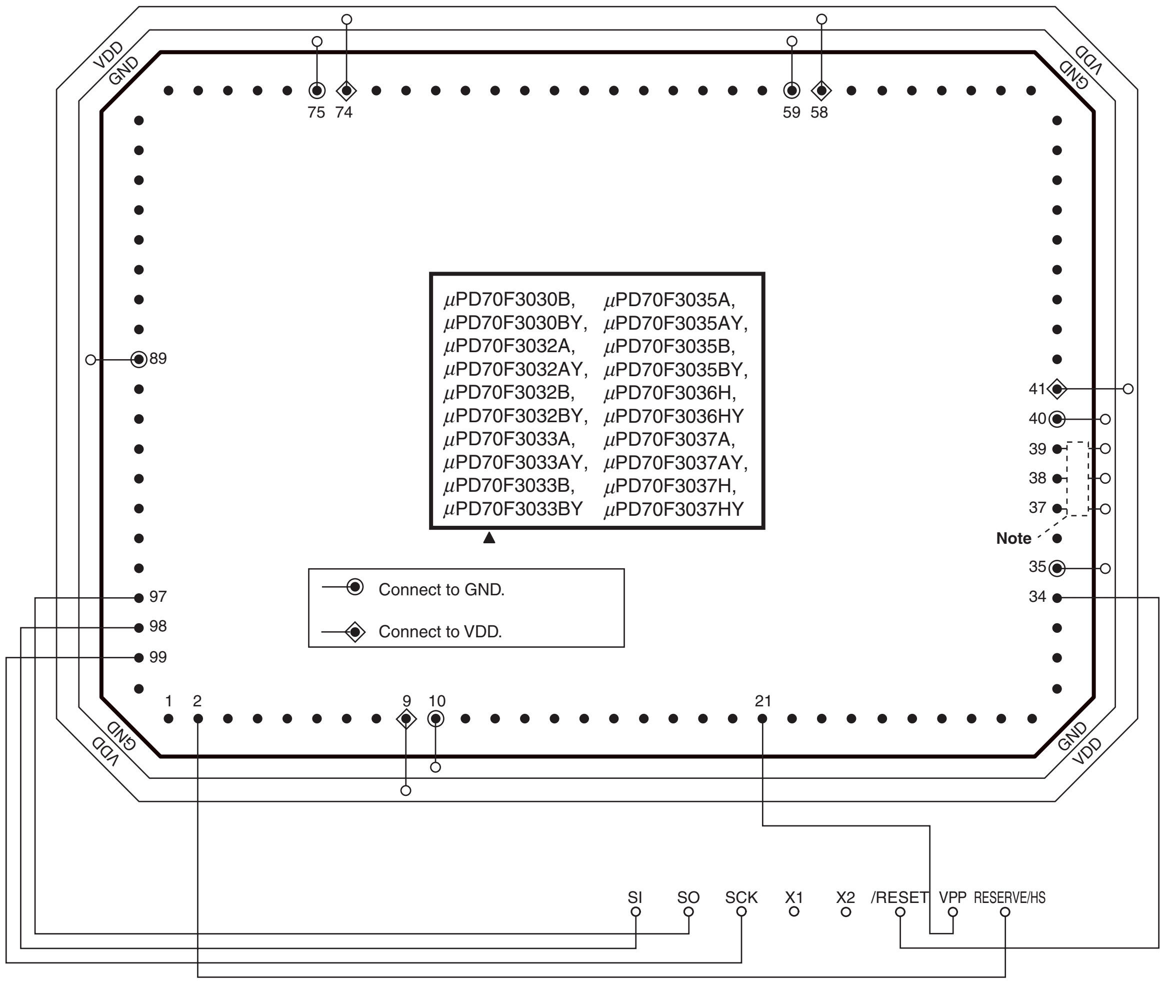
3. This diagram shows the wiring when using CSI supporting handshake.

★ **Table 18-1. Table for Wiring of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GC-8EU)**

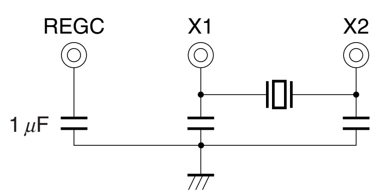
Flash Programmer (PG-FP3)			When Using CSI0 + HS		When Using CSI0		When Using UART0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RXD	Input	Receive signal	P11/SO0	95	P11/SO0	95	P14/SO1/TXD0	97
SO/TXD	Output	Transmit signal	P10/SI0/SDA0	94	P10/SI0/SDA0	94	P13/SO0/RXD0	98
SCK	Output	Transfer clock	P12/$\overline{SCK0}$/SCL0	96	P12/$\overline{SCK0}$/SCL0	96	Unnecessary	Unnecessary
CLK	–	Unused	Unnecessary	Unnecessary	Unnecessary	Unnecessary	Unnecessary	Unnecessary
/RESET	Output	Reset signal	$\overline{RESET}$	31	$\overline{RESET}$	31	$\overline{RESET}$	31
VPP	Output	Writing voltage	IC/V_{PP}	18	IC/V_{PP}	18	IC/V_{PP}	18
HS	Input	Handshake signal of CSI0 + HS communication	P15/$\overline{SCK4}$/ASCK0	99	Unnecessary	Unnecessary	Unnecessary	Unnecessary
VDD	–	VDD voltage generation/power supply monitoring	V_{DD}	38	V_{DD}	38	V_{DD}	38
			EV_{DD}	6	EV_{DD}	6	EV_{DD}	6
			BV_{DD}	5	BV_{DD}	5	BV_{DD}	5
			AV_{DD}	71	AV_{DD}	71	AV_{DD}	71
GND	–	Ground	V_{SS}	37	V_{SS}	37	V_{SS}	37
			EV_{SS}	7	EV_{SS}	7	EV_{SS}	7
			BV_{SS}	56	BV_{SS}	56	BV_{SS}	56
			AV_{SS}	72	AV_{SS}	72	AV_{SS}	72
			P00/NMI	86	P00/NMI	86	P00/NMI	86

Note The V850/SB1 and V850/SB2 cannot be supplied with the clock from the CLK pin of the flash programmer (PG-FP3).
Supply the clock by creating an oscillator on the flash writing adapter (broken-line portion).

★ **Figure 18-2. Wiring Example of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GF-3BA)**



Note The V850/SB1 and V850/SB2 cannot be supplied with the clock from the CLK pin of the flash programmer (PG-FP3).
Supply a clock by creating an oscillator on the flash writing adapter (broken-line portion).
An example of the oscillator is shown below.



Remarks 1. Handle the pins not described above in accordance with the recommended connection of unused pins (refer to **2.4 Pin I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins**). When connecting to VDD via a resistor, use of a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is for a 100-pin plastic QFP package.

3. This diagram shows the wiring when using CSI supporting handshake.

★ **Table 18-2. Table for Wiring of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GF-3BA)**

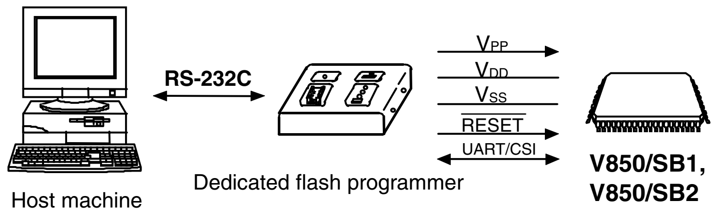
Flash Programmer (PG-FP3)			When Using CSI0 + HS		When Using CSI0		When Using UART0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RXD	Input	Receive signal	P11/SO0	98	P11/SO0	98	P14/SO1/TXD0	100
SO/TXD	Output	Transmit signal	P10/SI0/SDA0	97	P10/SI0/SDA0	97	P13/SO0/RXD0	1
SCK	Output	Transfer clock	P12/$\overline{\text{SCK0}}$/SCL0	99	P12/$\overline{\text{SCK0}}$/SCL0	99	Unnecessary	Unnecessary
CLK	–	Unused	Unnecessary	Unnecessary	Unnecessary	Unnecessary	Unnecessary	Unnecessary
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	34	$\overline{\text{RESET}}$	34	$\overline{\text{RESET}}$	34
VPP	Output	Writing voltage	IC/V_{PP}	21	IC/V_{PP}	21	IC/V_{PP}	21
HS	Input	Handshake signal of CSI0 + HS communication	P15/$\overline{\text{SCK4}}$/ASCK0	2	Unnecessary	Unnecessary	Unnecessary	Unnecessary
VDD	–	VDD voltage generation/power supply monitoring	V_{DD}	41	V_{DD}	41	V_{DD}	41
			EV_{DD}	9	EV_{DD}	9	EV_{DD}	9
			BV_{DD}	8	BV_{DD}	8	BV_{DD}	8
			AV_{DD}	74	AV_{DD}	74	AV_{DD}	74
GND	–	Ground	V_{SS}	40	V_{SS}	40	V_{SS}	40
			EV_{SS}	10	EV_{SS}	10	EV_{SS}	10
			BV_{SS}	59	BV_{SS}	59	BV_{SS}	59
			AV_{SS}	75	AV_{SS}	75	AV_{SS}	75
			P00/NMI	89	P00/NMI	89	P00/NMI	89

Note The V850/SB1 and V850/SB2 cannot be supplied with the clock from the CLK pin of the flash programmer (PG-FP3).
Supply the clock by creating an oscillator on the flash writing adapter (broken-line portion).

18.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850/SB1 and V850/SB2.

Figure 18-3. Environment Required for Writing Programs to Flash Memory



A host machine is required for controlling the dedicated flash programmer.

UART0 or CSI0 is used for the interface between the dedicated flash programmer and the V850/SB1 or V850/SB2 to perform writing, erasing, etc. A dedicated program adapter (FA Series) required for off-board writing.

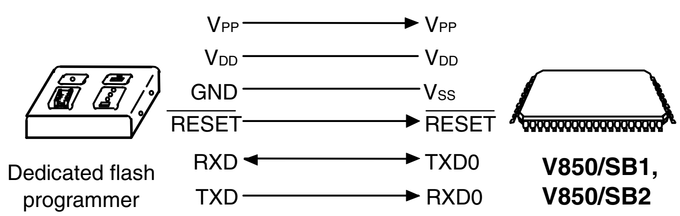
18.4 Communication Mode

The communication between the dedicated flash programmer and the V850/SB1 or V850/SB2 is performed by serial communication using UART0 or CSI0 of the V850/SB1, V850/SB2 .

(1) UART0

Transfer rate: 4800 to 76800 bps

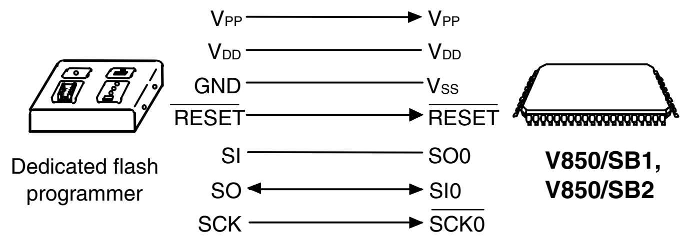
Figure 18-4. Communication with Dedicated Flash Programmer (UART0)



(2) CSI0

Serial clock: Up to 1 MHz (MSB first)

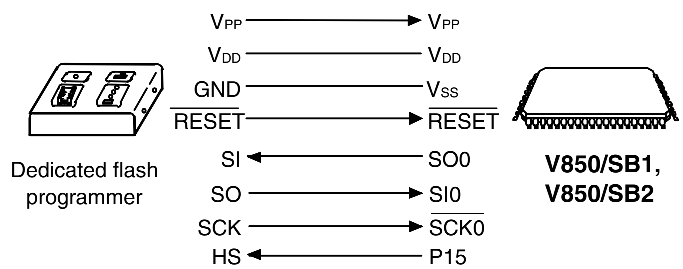
Figure 18-5. Communication with Dedicated Flash Programmer (CSI0)



(3) CSI0 + HS

Serial clock: Up to 1 MHz (MSB first)

Figure 18-6. Communication with Dedicated Flash Programmer (CSI0 + HS)



The dedicated flash programmer outputs the transfer clock, and the V850/SB1 and V850/SB2 operate as slaves.

When the PG-FP3 is used as the dedicated flash programmer, it generates the following signals to the V850/SB1 or V850/SB2 . For details, refer to the **PG-FP3 User's Manual**.

Table 18-3. Signal Generation of Dedicated Flash Programmer (PG-FP3)

PG-FP3			V850/SB1, V850/SB2	Connection Handling		
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/SB1, V850/SB2	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	P15	×	×	◎

Note Supply clocks on the target board.

Remark ◎: Always connected
○: Does not need to be connected, if generated on the target board
×: Does not need to be connected

18.5 Pin Connection

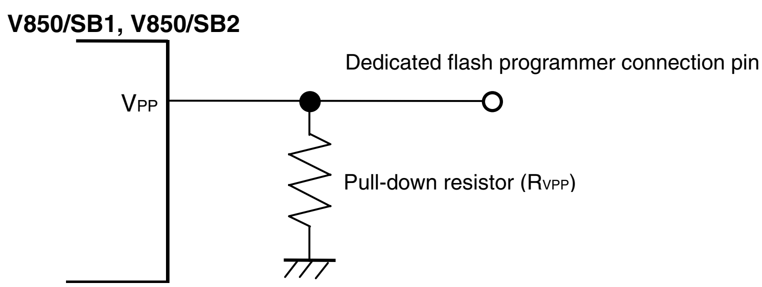
When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

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 enter the output high-impedance status, so that pin handling is required when the external device does not acknowledge the output high-impedance status.

18.5.1 V_{PP} pin

In the normal operation mode, 0 V is input to the V_{PP} pin. In the flash memory programming mode, a 7.8 V write voltage is supplied to the V_{PP} pin. The following shows an example of the connection of the V_{PP} pin.

Figure 18-7. V_{PP} Pin Connection Example



18.5.2 Serial interface pin

The following shows the pins used by each serial interface.

Table 18-4. Pins Used in Serial Interfaces

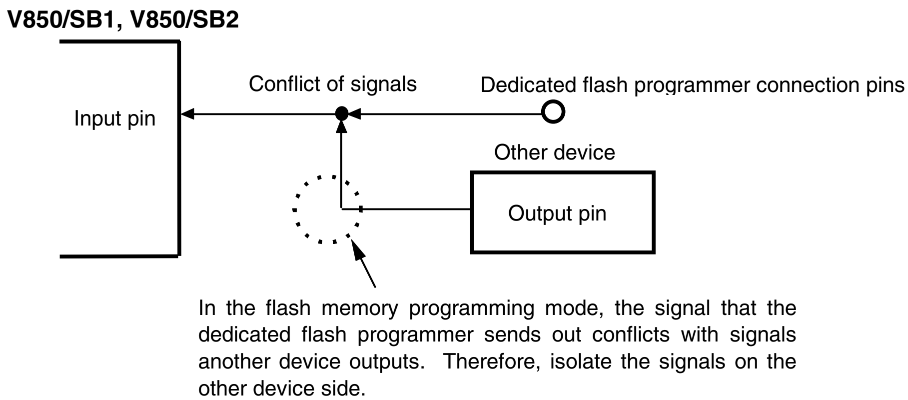
Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$
CSI0 + HS	SO0, SI0, $\overline{SCK0}$, P15
UART0	TXD0, RXD0

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.

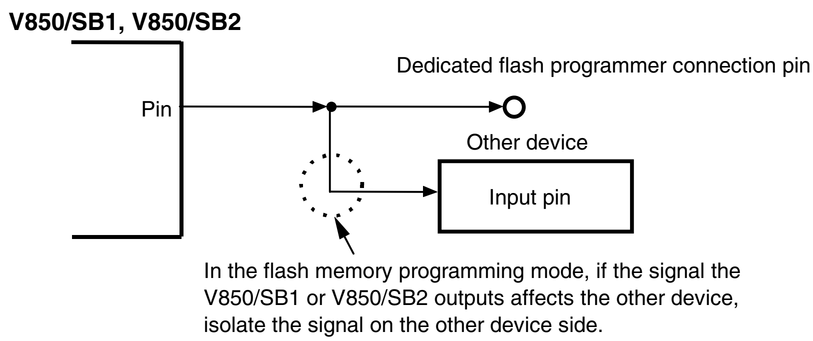
Figure 18-8. Conflict of Signals (Serial Interface Input Pin)

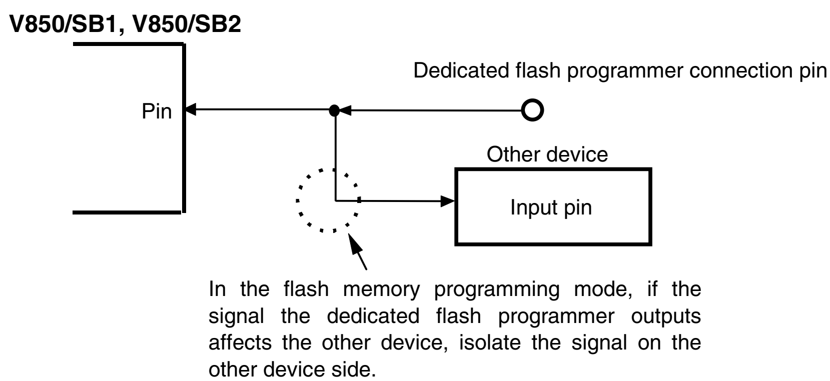


(2) Malfunction of other device

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 18-9. Malfunction of Other Device



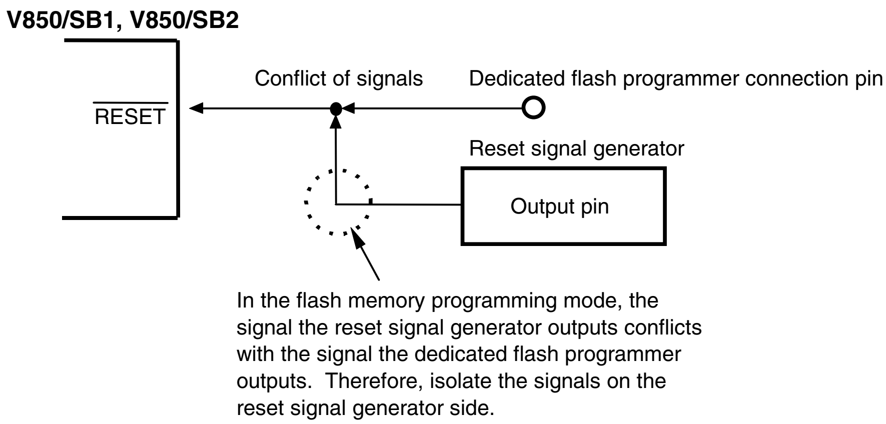


18.5.3 $\overline{\text{RESET}}$ pin

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 18-10. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



18.5.4 Port pins (including NMI)

When the flash memory programming mode is set, all the port pins except the pins that communicate with the dedicated flash programmer enter the output high-impedance status. If problems such as disabling output high-impedance status should occur in the external devices connected to the port, connect them to V_{DD} or V_{SS} via resistors.

18.5.5 Other signal pins

Connect X1, X2, XT2, and AV_{REF} to the same status as that in the normal operation mode.

18.5.6 Power supply

Supply the power supply as follows:

$V_{DD} = EV_{DD}$

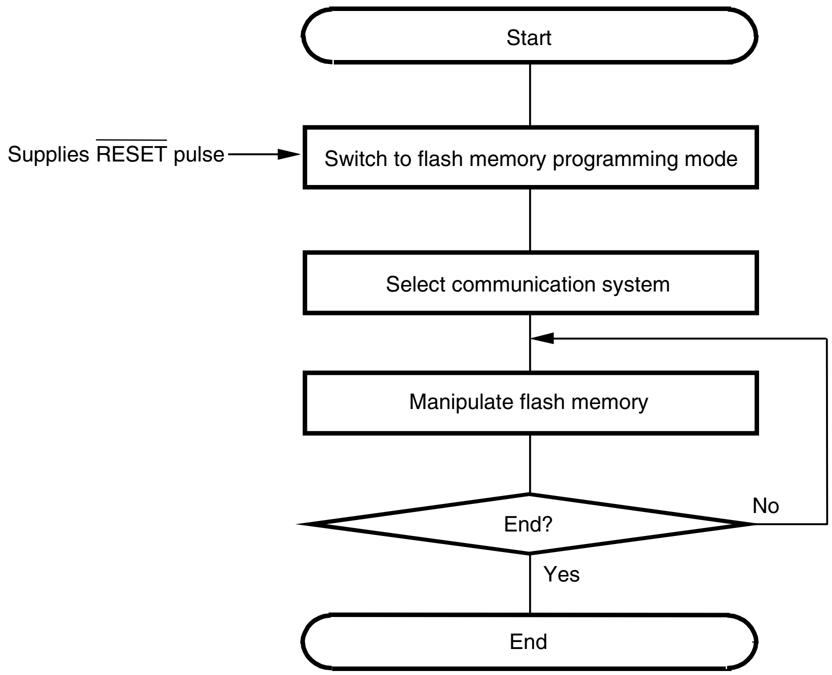
Supply the same power supply (AV_{DD}, AV_{SS}, BV_{DD}, BV_{SS}) as when in normal operation mode.

18.6 Programming Method

18.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 18-11. Procedure for Manipulating Flash Memory

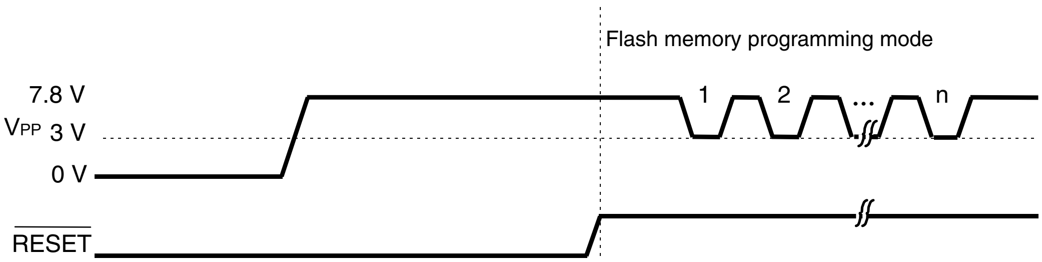


18.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850/SB1 or V850/SB2 in the flash memory programming mode. When switching modes, set the V_{PP} pin before releasing reset.

When performing on-board writing, change modes using a jumper, etc.

Figure 18-12. Flash Memory Programming Mode



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

18.6.3 Selection of communication mode

In the V850/SB1 and V850/SB2, the communication mode is selected by inputting pulses (16 pulses max.) to the V_{PP} pin after switching to the flash memory programming mode. The V_{PP} pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

Table 18-5. List of Communication Modes

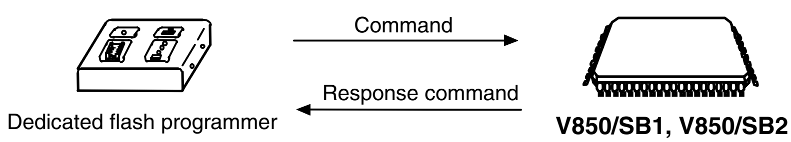
V_{PP} Pulse	Communication Mode	Remarks
0	CSI0	V850/SB1 and V850/SB2 perform slave operation, MSB first
3	CSI0 + HS	V850/SB1 and V850/SB2 perform slave operation, MSB first
8	UART0	Communication rate: 9600 bps (at reset), LSB first
Others	RFU	Setting prohibited

Caution When UART0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the V_{PP} pulse.

18.6.4 Communication command

The V850/SB1 and V850/SB2 communicate with the dedicated flash programmer by means of commands. The command sent from the dedicated flash programmer to the V850/SB1 or V850/SB2 is called a “command”. The response signal sent from the V850/SB1 or V850/SB2 to the dedicated flash programmer is called a “response command”.

Figure 18-13. Communication Command



The following shows the commands for flash memory control of the V850/SB1 and V850/SB2. All of these commands are issued from the dedicated flash programmer, and the V850/SB1 and V850/SB2 perform the various processing corresponding to the commands.

Table 18-6. Flash Memory Control Command

Category	Command Name	Function
Verify	One-short 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	Area erase command	Erases the specified area.
	Writeback command	Writes back the erased contents.
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.
	Writeback time setting command	Sets the writeback 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/SB1 and V850/SB2 send back response commands to the commands issued from the dedicated flash programmer. The following shows the response commands the V850/SB1 and V850/SB2 send out.

Table 18-7. Response Command

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

18.6.5 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.

CHAPTER 19 IEBus CONTROLLER (V850/SB2)

IEBus (Inter Equipment Bus) is a small-scale digital data transfer system that transfers data between units. To implement IEBus with the V850/SB2, an external IEBus driver and receiver are necessary because they are not provided.

The internal IEBus controller of the V850/SB2 is of negative logic.

19.1 IEBus Controller Function

19.1.1 Communication protocol of IEBus

The communication protocol of the IEBus is as follows:

(1) Multi-task mode

All the units connected to the IEBus can transfer data to the other units.

(2) Broadcasting communication function

Communication between one unit and plural units can be performed as follows:

- Group-unit broadcasting communication: Broadcasting communication to group units
- All-unit broadcasting communication: Broadcasting communication to all units.

(3) Effective transfer rate

The effective transfer rate is in mode 1 (the V850/SB2 does not support modes 0 and 2 for the effective transfer rate).

- Mode 1: Approx. 17 Kbps

Caution Different modes must not be mixed on one IEBus.

(4) Communication mode

Data transfer is executed in half-duplex asynchronous communication mode.

(5) Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

The priority of the IEBus is as follows:

<1> Broadcasting communication takes precedence over individual communication (communication from one unit to another).

<2> The lower master address takes precedence.

(6) Communication scale

The communication scale of IEBus is as follows:

- Number of units: 50 MAX.
- Cable length: 150 m MAX. (when twisted pair cable is used)

Caution The communication scale in an actual system differs depending on the characteristics of the cables, etc., constituting the IEBus driver/receiver and IEBus.

19.1.2 Determination of bus mastership (arbitration)

An operation to occupy the bus is performed when a unit connected to the IEBus controls the other units. This operation is called arbitration.

When two or more units simultaneously start transmission, arbitration is used to grant one of the units the permission to occupy the bus.

Because only one unit is granted the bus mastership as a result of arbitration, the priority conditions of the bus are predetermined as follows:

Caution The bus mastership is released if communication is aborted.

(1) Priority by communication type

Broadcasting communication (communication from one unit to plural units) takes precedence over normal communication (communication from one unit to another).

(2) Priority by master address

If the communication type is the same, communication with the lower master address takes precedence.

A master address consists of 12 bits, with unit 000H having the highest priority and unit FFFH having the lowest priority.

19.1.3 Communication mode

Although the IEBus has three communication modes each having a different transfer rate, the V850/SB2 supports only communication mode 1. The transfer rate and the maximum number of transfer bytes in one communication frame in communication mode 1 are as shown in Table 19-1.

Table 19-1. Transfer Rate and Maximum Number of Transfer Bytes in Communication Mode 1

Communication Mode	Maximum Number of Transfer Bytes (Bytes/Frame)	Effective Transfer Rate (Kbps)[Note]
1	32	Approx. 17

Note The effective transfer rate when the maximum number of transfer bytes is transmitted.

Select the communication mode (mode 1) for each unit connected to the IEBus before starting communication. If the communication mode of the master unit and that of the partner unit (slave unit) are not the same, communication is not correctly executed.

19.1.4 Communication address

With the IEBus, each unit is assigned a specific 12-bit address. This communication address consists of the following identification numbers:

- Higher 4 bits: Group number (number to identify the group to which each unit belongs)
- Lower 8 bits: Unit number (number to identify each unit in a group)

19.1.5 Broadcasting communication

Normally, transmission or reception is performed between the master unit and its partner slave unit on a one-to-one basis. During broadcasting communication, however, two or more slave units exist and the master unit executes transmission to these slave units. Because plural slave units exist, the slave units do not return an acknowledge signal during communication.

Whether broadcasting communication or normal communication is to be executed is selected by broadcasting bit (for this bit, refer to **19.1.6 (2) Broadcasting bit**).

Broadcasting communication is classified into two types: group-unit broadcasting communication and all-unit broadcasting communication. Group-unit broadcasting and all-unit broadcasting are identified by the value of the slave address (for the slave address, refer to **19.1.6 (4) Slave address field**).

(1) Group-unit broadcasting communication

Broadcasting communication is performed to the units in a group identified by the group number indicated by the higher 4 bits of the communication address.

(2) All-unit broadcasting communication

Broadcasting communication is performed to all the units, regardless of the value of the group number.

19.1.6 Transfer format of IEBus

Figure 19-1 shows the transfer signal format of the IEBus.

Figure 19-1. IEBus Transfer Signal Format

	Header		Master address field		Slave address field			Control field			Telegraph length field			Data field						
Frame format	Start bit	Broad-casting bit	Master address bit	P	Slave address bit	P	A	Control bit	P	A	Tele-graph length bit	P	A	Data bit	P	A	...	Data bit	P	A

Remarks 1. P: Parity bit, A: $\overline{\text{ACK}}$/NACK bit

2. The master station ignores the acknowledge bit during broadcasting communication.

(1) Start bit

The start bit is a signal that informs the other units of the start of data transfer. The unit that is to start data transfer outputs a high-level signal (start bit) from the $\overline{\text{IETX}}$ pin for a specific time, and then starts outputting the broadcasting bit.

If another unit has already output its start bit when one unit is to output the start bit, this unit does not output the start bit but waits for completion of output of the start bit by the other unit. When the output of the start bit by the other unit is complete, the unit starts outputting the broadcasting bit in synchronization with the completion of the start bit output by the other unit.

The units other than the one that has started communication detect this start bit, and enter the reception status.

(2) Broadcasting bit

This bit indicates whether the master selects one slave (individual communication) or plural slaves (broadcasting communication) as the other party of communication.

When the broadcasting bit is 0, it indicates broadcasting communication; when it is 1, individual communication is indicated. Broadcasting communication is classified into two types: group-unit communication and all-unit communication. These communication types are identified by the value of the slave address (for the slave address, refer to **19.1.6 (4) Slave address field**).

Because two or more slave units exist in the case of broadcasting communication, the acknowledge bit in each field subsequent to the master address field is not returned.

If two or more units start transmitting a communication frame at the same time, broadcasting communication takes precedence over individual communication, and wins in arbitration.

If one station occupies the bus as the master, the value set to the broadcasting request bit (ALLRQ) of the IEBus control register (BCR) is output.

(3) Master address field

The master address field is output by the master to inform a slave of the master's address.

The configuration of the master address field is as shown in Figure 19-2.

If two or more units start transmitting the broadcasting bit at the same time, the master address field makes a judgment of arbitration.

The master address field compares the data it outputs with the data on the bus each time it has output one bit. If the master address output by the master address field is found to be different from the data on the bus as a result of comparison, it is assumed that the master has lost in arbitration. As a result, the master stops transmission and enters the reception status.

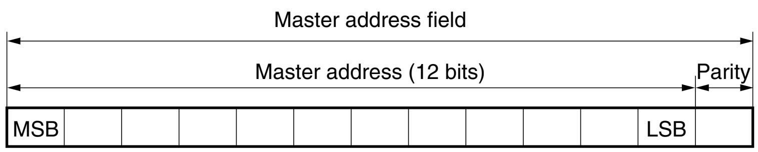
Because the IEBus is configured of wired AND, the unit having the minimum master address of the units participating in arbitration (arbitration masters) wins in arbitration.

After a 12-bit master address has been output, only one unit remains in the transmission status as one master unit.

Next, this master unit outputs a parity bit, determines the master address of other unit, and starts outputting a slave address field.

If one unit occupies the bus as the master, the address set by the IEBus unit address register (UAR) is output.

Figure 19-2. Master Address Field



(4) Slave address field

The master outputs the address of the unit with which it is to communicate.

Figure 19-3 shows the configuration of the slave address field.

A parity bit is output after a 12-bit slave address has been transmitted in order to prevent a wrong slave address from being received by mistake. Next, the master unit detects an acknowledge signal from the slave unit to confirm that the slave unit exists on the bus. When the master has detected the acknowledge signal, it starts outputting the control field. During broadcasting communication, however, the master does not detect the acknowledge bit but starts outputting the control field.

The slave unit outputs the acknowledge signal if its slave address matches and if the slave detects that the parities of both the master address and slave address are even. The slave unit judges that the master address or slave address has not been correctly received and does not output the acknowledge signal if the parities are odd. At this time, the master unit is in the standby (monitor) status, and communication ends.

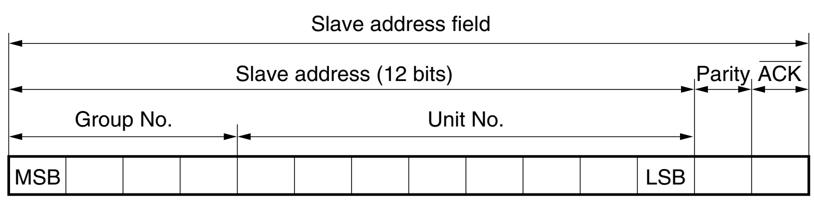
During broadcasting communication, the slave address is used to identify group-unit broadcasting or all-unit broadcasting, as follows:

If slave address is FFFH: All-unit broadcasting communication
If slave address is other than FFFH: Group-unit broadcasting communication

Remark The group No. during group-unit broadcasting communication is the value of the higher 4 bits of the slave address.

If one unit occupies the bus as the master, the address set by the slave address register (SAR) is output.

Figure 19-3. Slave Address Field



(5) Control field

The master outputs the operation it requires the slave to perform, by using this field.

The configuration of the control field is as shown in Figure 19-4.

If the parity following the control bit is even and if the slave unit can execute the function required by the master unit, the slave unit outputs an acknowledge signal and starts outputting the telegraph length field. If the slave unit cannot execute the function required by the master unit even if the parity is even, or if the parity is odd, the slave unit does not output the acknowledge signal, and returns to the standby (monitor) status.

The master unit starts outputting the telegraph field after confirming the acknowledge signal.

If the master cannot confirm the acknowledge signal, the master unit enters the standby status, and communication ends. During broadcasting communication, however, the master unit does not confirm the acknowledge signal, and starts outputting the telegraph length field.

Table 19-2 shows the contents of the control bits.

Table 19-2. Contents of Control Bits

Bit 3 [Note 1]	Bit 2	Bit 1	Bit 0	Function
0	0	0	0	Reads slave status
0	0	0	1	Undefined
0	0	1	0	Undefined
0	0	1	1	Reads data and locks [Note 2]
0	1	0	0	Reads lock address (lower 8 bits) [Note 3]
0	1	0	1	Reads lock address (higher 4 bits) [Note 3]
0	1	1	0	Reads slave status and unlocks [Note 2]
0	1	1	1	Reads data
1	0	0	0	Undefined
1	0	0	1	Undefined
1	0	1	0	Writes command and locks [Note 2]
1	0	1	1	Writes data and locks [Note 2]
1	1	0	0	Undefined
1	1	0	1	Undefined
1	1	1	0	Writes command
1	1	1	1	Writes data

Notes 1. The telegraph length bit of the telegraph length field and data transfer direction of the data field change as follows depending on the value of bit 3 (MSB).
If bit 3 is '1': Transfer from master unit to slave unit
If bit 3 is '0': Transfer from slave unit to master unit

2. This is a control bit that specifies locking or unlocking (refer to **19.1.7 (4) Locking and unlocking**).

3. The lock address is transferred in 1-byte (8-bit) units and is configured as follows:

	MSB	LSB
Control bit: 4H	Lower 8 bits	
Control bit: 5H	Undefined	Higher 4 bits

If the control bit received from the master unit is not as shown in Table 19-3, the unit locked by the master unit rejects acknowledging the control bit, and does not output the acknowledge bit.

Table 19-3. Control Field for Locked Slave Unit

Bit 3[Note 1]	Bit 2	Bit 1	Bit 0	Function
0	0	0	0	Reads slave status
0	1	0	0	Reads lock address (lower 8 bits)
0	0	0	1	Reads lock address (higher 4 bits)

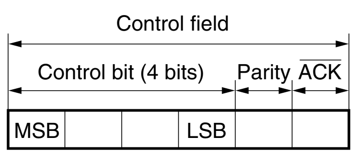
Moreover, units for which lock is not set by the master unit reject acknowledgment and do not output an acknowledge bit when the control data shown in Table 19-4 is acknowledged.

Table 19-4. Control Field for Unlocked Slave Unit

Bit 3	Bit 2	Bit 1	Bit 0	Function
0	1	0	0	Lock address read (lower 8 bits)
0	1	0	1	Lock address read (higher 4 bits)

If one unit occupies the bus as the master, the value set to the IEBus control register (CDR) is output.

Figure 19-4. Control Field



★ **Table 19-5. Acknowledge Signal Output Condition of Control Field**

(a) If received control data is AH, BH, EH, or FH

Communication Type (ALL TRANS) Independent communication = 0 Broadcasting communication = 1	Communication Target (SLVRQ) Slave Specification = 1 No Specification = 0	Lock Status (LOCK) Lock = 1 Unlock = 0	Master Unit Identification (Match with PAR) Lock Request Unit = 1 Other = 0	Slave Transmission Enable (ENSLVTX)	Slave Reception Enable (ENSLVRX)	Received Control Data			
						AH	BH	EH	FH
0	1	0	don’t care	don’t care	1	○			
		1	1						
Other than above						×			

(b) If received control data is 0H, 3H, 4H, 5H, 6H, or 7H

Communication Type (ALL TRANS) Independent communication = 0 Broadcasting communication = 1	Communication Target (SLVRQ) Slave Specification = 1 No Specification = 0	Lock Status (LOCK) Lock = 1 Unlock = 0	Master Unit Identification (Match with PAR) Lock Request Unit = 1 Other = 0	Slave Transmission Enable (ENSLVTX)	Slave Reception Enable (ENSLVRX)	Received Control Data					
						0H	3H	4H	5H	6H	7H
0	1	0	don’t care	0	don’t care	○	×	×	×	○	×
				1		○	○	×	×	○	○
		1		don’t care		○	×	○	○	×	×
			1			○	×	○	○	○	×
				1		○	○	○	○	○	○
Other than above						×					

Caution If the received control data is other than the data shown in Table 19-5, × ($\overline{\text{ACK}}$ is not returned) is unconditionally assumed.

Remarks 1. ○: $\overline{\text{ACK}}$ is returned.

×: $\overline{\text{ACK}}$ is not returned.

2. ENSLVTX: Bit 4 of the IEBus unit control register (BCR)

ENSLVRX: Bit 3 of the IEBus unit control register (BCR)

LOCK: Bit 2 of the IEBus unit status register (USR)

SLVRQ: Bit 6 of the IEBus unit status register (USR)

PAR: IEBus partner address register

(6) Telegraph length field

This field is output by the transmission side to inform the reception side of the number of bytes of the transmit data.

The configuration of the telegraph length field is as shown in Figure 19-5.

Table 19-6 shows the relationship between the telegraph length bit and the number of transmit data.

Figure 19-5. Telegraph Length Field

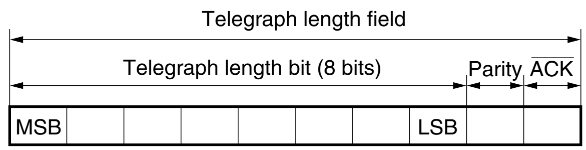


Table 19-6. Contents of Telegraph Length Bit

Telegraph Length Bit (Hex)	Number of Transmit Data Bytes
01H	1 byte
02H	2 bytes
|	|
FFH	255 bytes
00H	256 bytes

The operation of the telegraph length field differs depending on whether the master transmits data (when control bit 3 is 1) or receives data (when control bit 3 is 0).

(a) When master transmits data

The telegraph length bit and parity bit are output by the master unit and the synchronization signals of bits are output by the master unit. When the slave unit detects that the parity is even, it outputs the acknowledge signal, and starts outputting the data field. During broadcasting communication, however, the slave unit does not output the acknowledge signal.

If the parity is odd, the slave unit judges that the telegraph length bit has not been correctly received, does not output the acknowledge signal, and returns to the standby (monitor) status. At this time, the master unit also returns to the standby status, and communication ends.

(b) When master receives data

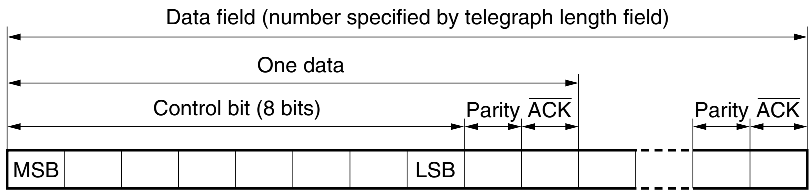
The telegraph length bit and parity bit are output by the slave unit and the synchronization signals of bits are output by the master unit. If the master unit detects that the parity bit is even, it outputs the acknowledge signal.

If the parity bit is odd, the master unit judges that the telegraph length bit has not been correctly received, does not output the acknowledge signal, and returns to the standby status. At this time, the slave unit also returns to the standby status, and communication ends.

(7) Data field

This is data output by the transmission side.
The master unit transmits or receives data to or from a slave unit by using the data field.
The configuration of the data field is as shown below.

Figure 19-6. Data Field



Following the data bit, the parity bit and acknowledge bit are respectively output by the master unit and slave unit.
Use broadcasting communication only for when the master unit transmits data. At this time, the acknowledge bit is ignored.

The operation differs as follows depending on whether the master transmits or receives data.

(a) When master transmits data

When the master units writes data to a slave unit, the master unit transmits the data bit and parity bit to the slave unit. If the parity is even and the receive data is not stored in the IEBus data register (DR) when the slave unit has received the data bit and parity bit, the slave unit outputs an acknowledge signal. If the parity is odd or if the receive data is stored in the IEBus data register (DR), the slave unit rejects receiving the data, and does not output the acknowledge signal.

If the slave unit does not output the acknowledge signal, the master unit transmits the same data again. This operation continues until the master detects the acknowledge signal from the slave unit, or the data exceeds the maximum number of transmit bytes.

If the data has continuation and the maximum number of transmit bytes is not exceeded when the parity is even and when the slave unit outputs the acknowledge signal, the master unit transmits the next data.

During broadcasting communication, the slave unit does not output the acknowledge signal, and the master unit transfers 1 byte of data at a time. If the parity is odd or the DR register is storing receive data after the slave unit has received the data bit and parity bit during broadcasting communication, the slave unit judges that reception has not been performed correctly, and stops reception.

(b) When master receives data

When the master unit reads data from a slave unit, the master unit outputs a sync signal corresponding to all the read bits.

The slave unit outputs the contents of the data and parity bits to the bus in response to the sync signal from the master unit.

The master unit reads the data and parity bits output by the slave unit, and checks the parity.

If the parity is odd, or if the DR register is storing a receive data, the master unit rejects accepting the data, and does not output the acknowledge signal. If the maximum number of transmit bytes is within the value that can be transmitted in one communication frame, the master unit repeats reading the same data.

If the parity is even and the DR register is not storing a receive data, the master unit accepts the data and returns the acknowledge signal. If the maximum number of transmit bytes is within the value that can be transmitted in one frame, the master unit reads the next data.

Caution Do not operate master reception in broadcasting communication, because the slave unit cannot be defined and data transfer cannot be performed correctly.

(8) Parity bit

The parity bit is used to check to see if the transmit data has no error.

The parity bit is appended to each data of the master address, slave address, control, telegraph length, and data bits.

The parity is an even parity. If the number of bits in data that are ‘1’ is odd, the parity bit is ‘1’. If the number of bits in the data that are ‘1’ is even, the parity bit is ‘0’.

(9) Acknowledge bit

During normal communication (communication from one unit to another), an acknowledge bit is appended to the following locations to check if the data has been correctly received.

- End of slave address field
- End of control field
- End of telegraph length field
- End of data field

The definition of the acknowledge bit is as follows:

- 0: Indicates that the transmit data is recognized ($\overline{\text{ACK}}$).
- 1: Indicates that the transmit data is not recognized (NACK).

During broadcasting communication, however, the contents of the acknowledge bit are ignored.

(a) Last acknowledge bit of slave field

The last acknowledge bit of the slave field serves as NACK in any of the following cases, and transmission is stopped.

- If the parity of the master address bit or slave address bit is incorrect
- If a timing error (error in bit format) occurs
- If a slave unit does not exist

(b) Last acknowledge bit of control field

The last acknowledge bit of the control field serves as NACK in any of the following cases, and transmission is stopped.

- If the parity of the control bit is incorrect
- If control bit 3 is ‘1’ (write operation) when the slave reception enable flag (ENSLVRX) is not set (1)[Note]
- If the control bit indicates reading of data (3H or 7H) when the slave transmission enable flag (ENSLVTX) is not set (1)[Note]
- If a unit other than that has set locking requests 3H, 6H, 7H, AH, BH, EH, or FH of the control bit when locking is set
- If the control bit indicates reading of lock addresses (4H, 5H) even when locking is not set
- If a timing error occurs
- If the control bit is undefined

Note Refer to **19.3.2 (1) IEBus control register (BCR)**.

Cautions 1. Even when the slave transmission enable flag (ENSLVTX) is not set (1), $\overline{\text{ACK}}$ is always returned if slave status request control data is received.

2. Even when slave reception enable flag (ENSLVRX) is not set (1), NACK is always returned by the acknowledge bit in the control field if data/command writing control data is acknowledged.

Slave reception can be disabled (communication stopped) by ENSLVRX flag only in the case of independent communication. In the case of broadcasting communication, communication is maintained and the data request interrupt (INTIE1) or IEBus end interrupt (INTIE2) is generated.

(c) Last acknowledge bit of telegraph length field

The last acknowledge bit of the telegraph length field serves as NACK in any of the following cases, and transmission is stopped.

- If the parity of the telegraph length bit is incorrect
- If a timing error occurs

(d) Last acknowledge bit of data field

The last acknowledge bit of the data field serves as NACK in any of the following cases, and transmission is stopped.

- If the parity of the data bit is incorrect[Note]
- If a timing error occurs after the preceding acknowledge bit has been transmitted
- If the receive data is stored in the IEBus data register (DR) and no more data can be received[Note]

Note In this case, when the communication executed is individual communication, if the maximum number of transmit bytes is within the value that can be transmitted in one frame, the transmission side executes transmission of that data field again. For broadcasting communication, the transmission side does not execute transmission again, a communication error occurs on the reception side and reception stops.

19.1.7 Transfer data

(1) Slave status

The master unit can learn why the slave unit did not return the acknowledge bit ($\overline{\text{ACK}}$) by reading the slave status.

The slave status is determined according to the result of the last communication the slave unit has executed.

All the slave units can supply information on the slave status.

The configuration of the slave status is shown below.

Figure 19-7. Bit Configuration of Slave Status

MSB							LSB
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 0[Note 1]	Meaning
0	Transmit data is not written in IEBus data register (DR)
1	Transmit data is written in IEBus data register (DR)

Bit 1[Note 2]	Meaning
0	Receive data is not stored in IEBus data register (DR)
1	Receive data is stored in IEBus data register (DR)

Bit 2	Meaning
0	Unit is not locked
1	Unit is locked

Bit 3	Meaning
0	Fixed to 0

Bit 4[Note 3]	Meaning
0	Slave transmission is stopped
1	Slave transmission is ready

Bit 5	Meaning
0	Fixed to 0

Bit 7	Bit 6	Meaning	
0	0	Mode 0	Indicates the highest mode supported by unit[Note 4].
0	1	Mode 1	
1	0	Mode 2	
1	1	Not used	

Notes 1. After reset: Bit 0 is set to 1.

2. The receive buffer size is 1 byte.

3. When the V850/SB2 serves as a slave unit, this bit corresponds to the status indicated by bit 4 (ENSLVTX) of the IEBus control register (BCR).

4. When the V850/SB2 serves as a slave unit, bits 7 and 6 are fixed to '0' and '1' (mode 1), respectively.

(2) Lock address

When the lock address is read (control bit: 4H or 5H), the address (12 bits) of the master unit that has issued the lock instruction is configured in 1-byte units as shown below and read.

Figure 19-8. Configuration of Lock Address

	MSB	LSB
Control bit: 4H	Lower 8 bits	
Control bit: 5H	Undefined	Higher 4 bits

(3) Data

If the control bit indicates reading of data (3H or 7H), the data in the data buffer of the slave unit is read by the master unit.

If the control bit indicates writing of data (BH or FH), the data received by the slave unit is processed according to the operation rule of that slave unit.

(4) Locking and unlocking

The lock function is used when a message is transferred in two or more communication frames.

The unit that is locked does not receive data from units other than the one that has locked the unit (does not receive broadcasting communication).

A unit is locked or unlocked as follows:

(a) Locking

If the communication frame is completed without succeeding to transmit or receive data of the number of bytes specified by the telegraph length bit after the telegraph length field has been transmitted or received ($\overline{ACK}$ = 0) by the control bit that specifies locking (3H, AH, or BH), the slave unit is locked by the master unit. At this time, the bit (bit 2) in the byte indicating the slave status is set to ‘1’.

(b) Unlocking

After transmitting or receiving data of the number of data bytes specified by the telegraph length bit in one communication frame by the control bit that has specified locking (3H, AH, or BH), or the control bit that has specified unlocking (6H), the slave unit is unlocked by the master unit. At this time, the bit related to locking (bit 2) in the byte indicating the slave status is reset to ‘0’.

Locking or unlocking is not performed during broadcasting communication.

Locking and unlocking conditions are shown below.

(c) **Lock setting conditions**

Control Data	Broadcasting Communication		Individual Communication	
	Communication End	Frame End	Communication End	Frame End
3H, 6H[Note]			Cannot be locked	Lock set
AH, BH	Cannot be locked	Cannot be locked	Cannot be locked	Lock set
0H, 4H, 5H, EH, FH	Cannot be locked	Cannot be locked	Cannot be locked	Cannot be locked

(d) **Unlock release conditions (while locked)**

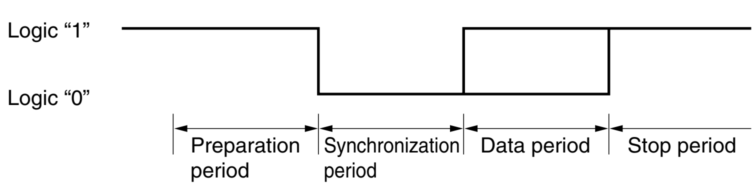
Control Data	Broadcasting Communication from Lock Request Unit		Individual Communication from Lock Request Unit	
	Communication End	Frame End	Communication End	Frame End
3H, 6H[Note]			Unlocked	Remains locked
AH, BH	Unlocked	Unlocked	Unlocked	Remains locked
0H, 4H, 5H, EH, FH	Remains locked	Remains locked	Remains locked	Remains locked

Note The frame end of control data 6H (slave status read/unlock) occurs when the parity in the data field is odd, and when the acknowledge signal from the IEBus unit is repeated up to the maximum number of transfer bytes without being output.

19.1.8 Bit format

The format of the bits constituting the communication frame of the IEBus is shown below.

Figure 19-9. Bit Format of IEBus



Preparation period:	First low-level (logic “1”) period
Synchronization period:	Next high-level (logic “0”) period
Data period:	Period indicating value of bit
Stop period:	Last low-level (logic “1”) period

The synchronization period and data period are almost equal to each other in length.

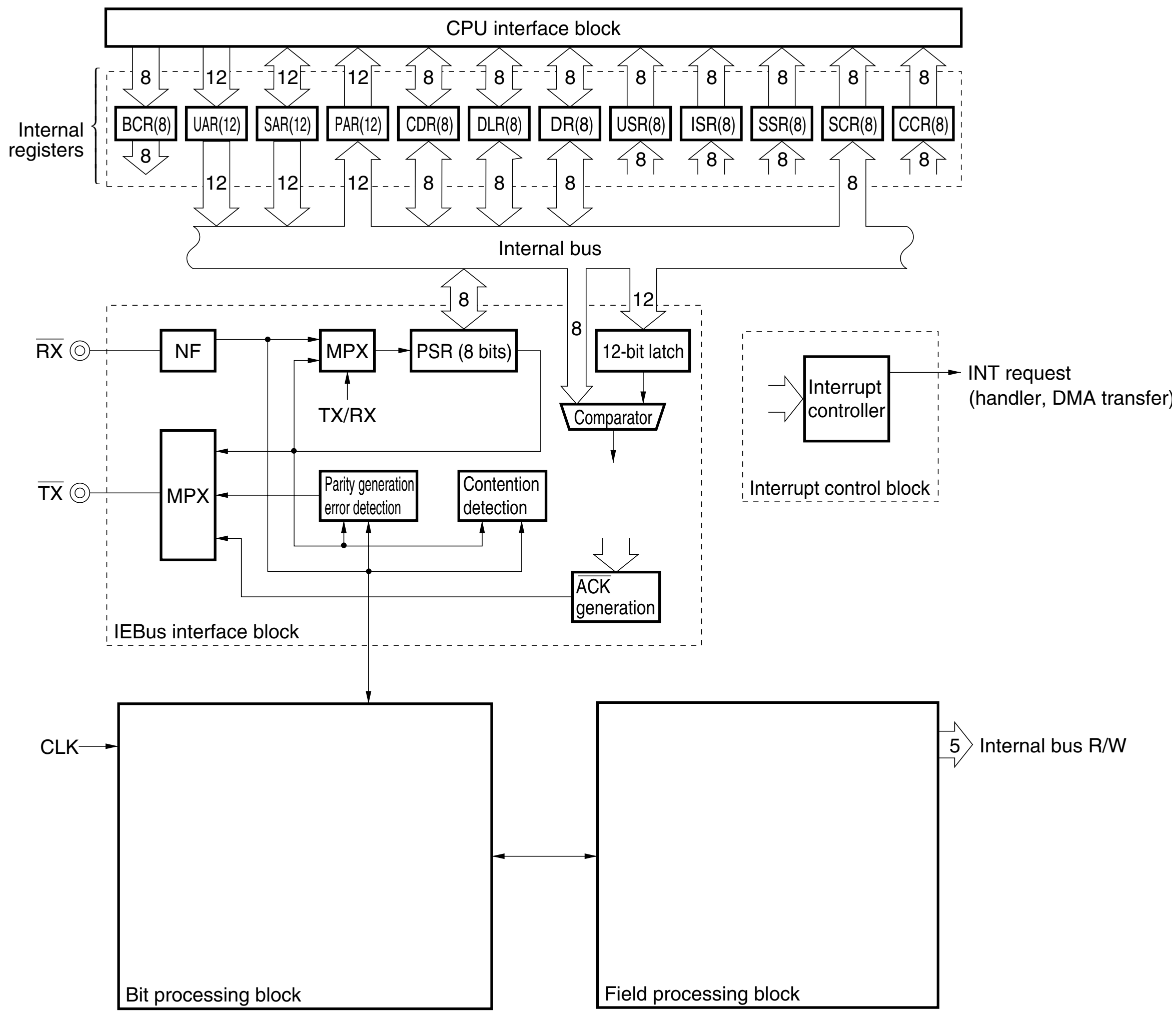
The IEBus synchronizes each bit. The specifications on the time of the entire bit and the time related to the period allocated to that bit differ depending on the type of the transmit bit, or whether the unit is the master unit or a slave unit.

The master and slave units monitor whether each period (preparation period, synchronization period, data period, and stop period) is output for specified time while they are in communication. If a period is not output for the specified time, the master and slave units report a timing error, immediately terminate communication and enter the standby status.

19.2 IEBus Controller Configuration

The block diagram of the IEBus controller is shown below.

Figure 19-10. IEBus Controller Block Diagram



(1) Hardware configuration and function

The IEBus mainly consists of the following six internal blocks.

- CPU interface block
- Interrupt control block
- Internal registers
- Bit processing block
- Field processing block
- IEBus interface block

(a) CPU interface block

This is a control block that interfaces between the CPU (V850/SB2) and IEBus.

(b) Interrupt control block

This control block transfers interrupt request signals from IEBus to the CPU.

(c) Internal registers

These registers set data to the control registers and fields that control IEBus (for the internal registers, refer to **19.3 Internal Registers of IEBus Controller**).

(d) Bit processing block

This block generates and disassembles bit timing, and mainly consists of a bit sequence ROM, 8-bit preset timer, and comparator.

(e) Field processing block

This block generates each field in the communication frame, and mainly consists of a field sequence ROM, 4-bit down counter, and comparator.

(f) IEBus interface block

This is the interface block for an external driver/receiver, and mainly consists of a noise filter, shift register, collision detector, parity detector, parity generator, and $\overline{\text{ACK}}$/NACK generator.

19.3 Internal Registers of IEBus Controller

19.3.1 Internal register list

Table 19-7. Internal Registers of IEBus Controller

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF3E0H	IEBus control register	BCR	R/W	√	√	–	00H
FFFFF3E2H	IEBus unit address register	UAR		–	–	√	0000H
FFFFF3E4H	IEBus slave address register	SAR		–	–	√	
FFFFF3E6H	IEBus partner address register	PAR	R	–	–	√	
FFFFF3E8H	IEBus control data register	CDR	R/W	–	√	–	01H
FFFFF3EAH	IEBus telegraph length register	DLR		–	√	–	
FFFFF3ECH	IEBus data register	DR		–	√	–	00H
FFFFF3EEH	IEBus unit status register	USR	R	√	√	–	
FFFFF3F0H	IEBus interrupt status register	ISR	R/W	√	√	–	
FFFFF3F2H	IEBus slave status register	SSR	R	√	√	–	41H
FFFFF3F4H	IEBus communication success counter	SCR		–	√	–	01H
FFFFF3F6H	IEBus transmit counter	CCR		–	√	–	20H
FFFFF3F8H	IEBus clock selection register	IECLK	R/W	–	√	–	00H
★ FFFFF3DEH	IEBus high-speed clock selection register	IEHCLK[Note]	R/W	–	√	–	00H

Note The IEHCLK register is available only in the V850/SB2.

19.3.2 Internal registers

The internal registers incorporated in the IEBus controller are described below.

(1) IEBus control register (BCR)

After reset: 00H RW Address: FFFFF3E0H

	<7>	<6>	<5>	<4>	<3>	2	1	0
BCR	ENIEBUS	MSTRQ	ALLRQ	ENSLVTX	ENSLVRX	0	0	0

ENIEBUS	Communication enable flag
0	IEBus unit stopped
1	IEBus unit active

MSTRQ	Master request flag
0	IEBus unit not requested as master
1	IEBus unit requested as master

ALLRQ	Broadcast request flag
0	Individual communication requested
1	Broadcasting communication requested

ENSLVTX	Slave transmission enable flag
0	Slave transmission disabled
1	Slave transmission enabled

ENSLVRX	Slave reception enable flag
0	Slave reception disabled
1	Slave reception enabled

Cautions 1. While the IEBus is operating as the master, writing to the BCR register (including bit manipulation instructions) is disabled until either the end of that communication or frame, or until communication is stopped by the occurrence of an arbitration-loss communication error. Master requests cannot therefore be multiplexed. However, if the IEBus is specified as a slave while a master request is being held pending, the BCR can be written to at the end of communication to clear the communication end/frame end flag. This is also the case when communication has been forcibly stopped (ENIEBUS flag = 0).

2. If a bit manipulation instruction for the BCR register conflicts with a hardware reset of the MSTRQ flag, the BCR register may not operate normally. The following countermeasures are recommended in this case.

- **Because the hardware reset is instigated in the acknowledgment period of the slave address field, be sure to observe Caution 1 of (b) Master request flag (MSTRQ) below.**
- **Be sure to observe the caution above regarding writing to the BCR register.**

(a) Communication enable flag (ENIEBUS)...Bit 7

<Set/reset conditions>

Set: By software

Reset: By software

Caution Before setting the ENIBUS flag, make the following setting:

- **Set the interrupt enabled (EI) status and enable the interrupt processing of INTIE2 (IEBMK = 2).**
- **Set the IEBus unit address register (UAR)**

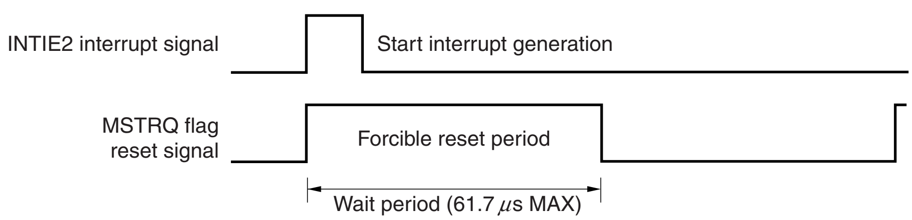
(b) Master request flag (MSTRQ)...Bit 6

<Set/reset conditions>

Set: By software

Reset: By hardware, at the end of the arbitration period. Because the reset signal is generated in the ACK period of the slave address field, if a MSTRQ flag setting instruction is sent in this period, it will be invalid.

Cautions 1. The master request should be resent by software following a loss in arbitration. When resending the master request in this case, set (1) the MSTRQ flag after securing the required wait period. This flag is unable to be set (1) before the end of this wait period.



2. When a master request has been sent and bus mastership acquired, do not set the MSTRQ, ENSLVTX, or ENSLVRX flag until the end of communication (i.e. the ISR register's communication end/frame end flag is set (1)) as setting these flags disables interrupt request generation. However, these flags can be set if communication has been aborted.

(c) Broadcast request flag (ALLRQ)...Bit 5

<Set/reset conditions>

Set: By software

Reset: By software

Caution When requesting broadcasting communication, always set the ALLRQ flag, then the MSTRQ flag.

(d) Slave transmission enable flag (ENSLVTX)...Bit 4

<Set/reset conditions>

Set: By software

Reset: By software

Cautions 1. Clear the ENSLVTX flag before setting the MSTRQ flag when making a master request. If a slave transmission request is sent in slave mode when the ENSLVTX flag is unset, NACK in the control field will be returned. Moreover, when returning to an enabled state from a disabled state, transmission becomes valid from the next frame.

2. If the controller receives control data for data/control writing (3H, 7H) when the ENSLVTX flag is unset, NACK will be returned via the acknowledge bit of the control field.

3. The status interrupt (INTIE2) will be generated and communication continued when the control data of a slave status request is returned, even if the ENSLVTX flag is in the reset status.

(e) Slave reception enable flag (ENSLVRX)...Bit 3

<Set/reset conditions>

Set: By software

Reset: By software

Caution If the ENSLVRX flag is reset when the IEBus is busy with other CPU processing, NACK will be returned via the acknowledge bit of the control field, making it possible to disable slave reception. Note that resetting this flag only disables individual communication, not broadcasting communication. If the received slave address matches the unit address during individual communication, however, the start interrupt (INTIE2) is generated. If CPU processing has priority (neither reception nor transmission occurs), be sure to stop the IEBus unit by resetting the ENIEBUS flag. Note also that when returning to an enabled state from a disabled state, transmission becomes valid from the next frame.

(2) IEBus unit address register (UAR)

This register sets the unit address of an IEBus unit. This register must be always set before starting communication.

Sets the unit address (12 bits) to bits 11 to 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset	R/W
UAR	0	0	0	0													FFFFF3E2H	0000H	R/W

(3) IEBus slave address register (SAR)

During master request, the value of this register is reflected in the value of the transmit data in the slave address field. This register must be always set before starting communication.

Sets the slave address (12 bits) to bits 11 to 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset	R/W
SAR	0	0	0	0													FFFFF3E4H	0000H	R/W

(4) IEBus partner address register (PAR)

(a) When slave unit

The value of the receive data in the master address field (address of the master unit) is written to this register.

If a request “4H” to read the lock address (lower 8 bits) is received from the master, the CPU must read the value of this register, and write it to the lower 8 bits IEBus data register (DR).

If a request “5H” to read the lock address (higher 4 bits) is received from the master, the CPU must read the value of this register and write the data of the higher 4 bits to DR.

Sets the partner address (12 bits) to bits 11 to 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset	R/W
PAR	0	0	0	0													FFFFF3E6H	0000H	R

(5) IEBus control data register (CDR)

(a) When master unit

The data of the lower 4 bits is reflected in the data transmitted in the control field. During master request, this register must be set in advance before starting communication.

(b) When slave unit

The data received in the control field is written to the lower 4 bits.

When the status transmission flag (STATUS) of the IEBus interrupt status register (ISR) is set, an interrupt (INTIE2) is issued, and each processing should be performed by software, according to the value of the lower 4 bits of CDR.

After reset: 01H R/W Address: FFFFF3E8H

	7	6	5	4	3	2	1	0
CDR	0	0	0	0	MOD	SELCL2	SELCL1	SELCL0

MOD	SELCL2	SELCL1	SELCL0	Function
0	0	0	0	Reads slave status
0	0	0	1	Undefined
0	0	1	0	Undefined
0	0	1	1	Reads data and locks
0	1	0	0	Reads lock address (lower 8 bits)
0	1	0	1	Reads lock address (lower 4 bits)
0	1	1	0	Reads slave status and unlocks
0	1	1	1	Reads data
1	0	0	0	Undefined
1	0	0	1	Undefined
1	0	1	0	Writes command and locks
1	0	1	1	Writes data and locks
1	1	0	0	Undefined
1	1	0	1	Undefined
1	1	1	0	Writes command
1	1	1	1	Writes data

Cautions 1. Because the slave unit must judge whether the received data is a “command” or “data”, it must read the value of this register after completing communication.

2. If the master unit sets an undefined value, NACK is returned from the slave unit, and communication is aborted. During broadcasting communication, however, the master unit continues communication without recognizing $\overline{\text{ACK}}$/NACK; therefore, make sure not to set an undefined value to this register during broadcasting communication.

3. In the case of defeat in a bus conflict and a slave status request is received from the unit that won, the telegraph length register (DLR) is fixed to “01H”. Therefore, when a re-request of the master follows, the appointed telegraph length must be set to DLR.

(c) Slave status return operation

When the IEBus receives a request to transfer from master to slave status or a lock address request (control data: 0H, 6H), whether $\overline{\text{ACK}}$ in the control field is returned or not depends on the status of the IEBus unit.

(1) If 0H or 6H control data was received in the unlocked state → $\overline{\text{ACK}}$ returned
(2) If 4H or 5H control data was received in the unlocked state → $\overline{\text{ACK}}$ not returned
(3) If 0H, 4H, 5H or 6H control data was received in the locked state from the unit that sent the lock request → $\overline{\text{ACK}}$ returned
(4) If 0H, 4H, or 5H control data was received in the locked state from other than the unit that sent the lock request → $\overline{\text{ACK}}$ returned
(5) If 6H control data was received in the locked state from other than the unit that sent the lock request → $\overline{\text{ACK}}$ not returned

In all of the above cases, the acknowledgment of a slave status or lock request will cause the STATUSF flag (bit 4 of the ISR register) to be set and the status interrupt (INTIE2) to be generated. The generation timing is at the end of the control field parity bit (at the start of the $\overline{\text{ACK}}$ bit). However, if $\overline{\text{ACK}}$ is not returned, a NACK error is generated after the $\overline{\text{ACK}}$ bit, and communication is terminated.

Figure 19-11. Interrupt Generation Timing (for (1), (3), and (4))

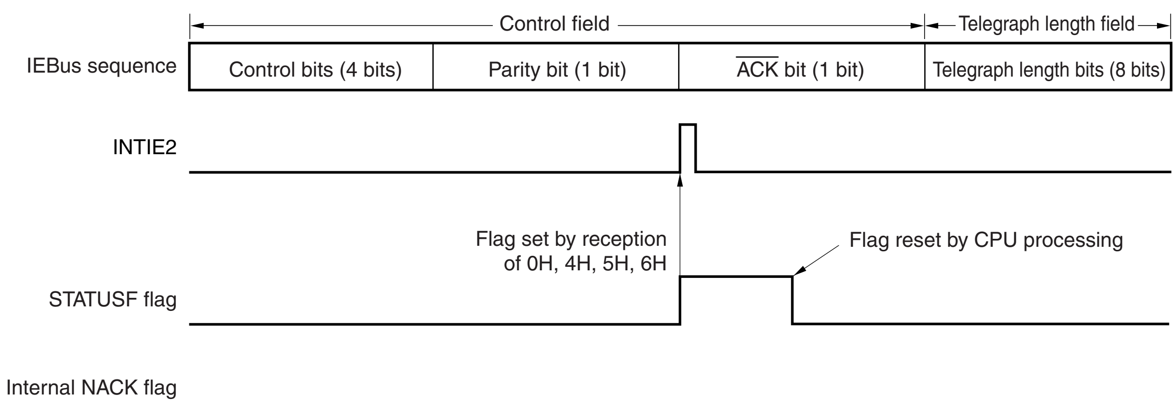
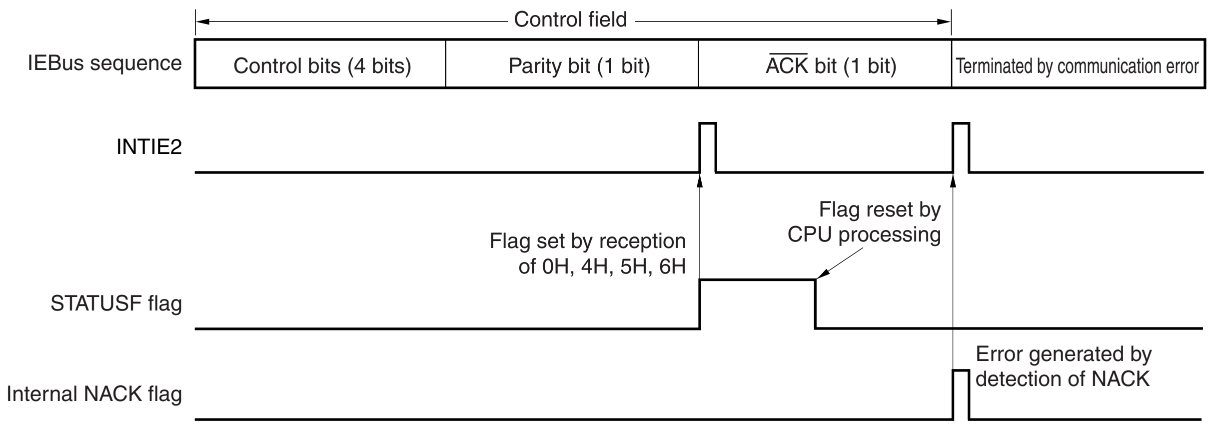
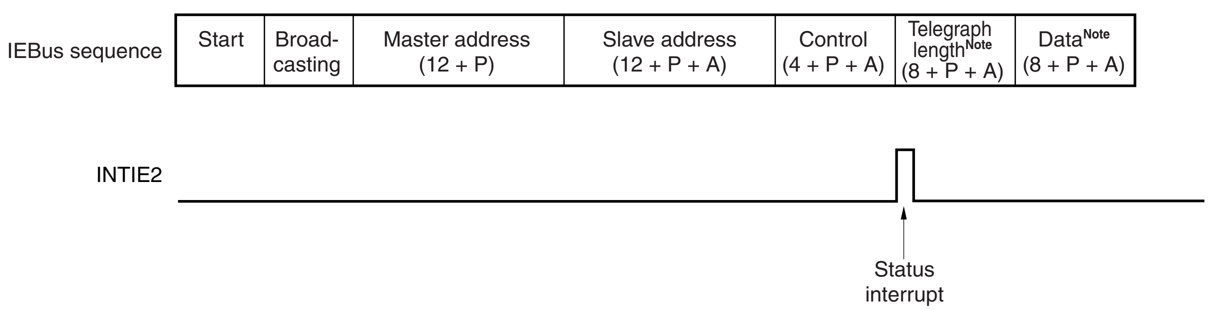


Figure 19-12. Interrupt Generation Timing (for (2) and (5))



Because in (4) and (5) the communication was from other than the unit that sent the lock request while the IEBus was in the locked state, the start or communication complete interrupt (INTIE2) is not generated, even if the IEBus unit is the communication target. The STATUSF flag (bit 4 of the ISR register) is set and the status interrupt (INTIE2) generated, however, if a slave status or lock address request is acknowledged. Note that even if the same control data is received while the IEBus is in the locked state, the interrupt generation timing for INTIE2 differs depending on whether the master unit (3) or another unit (4) is requesting the locked state.

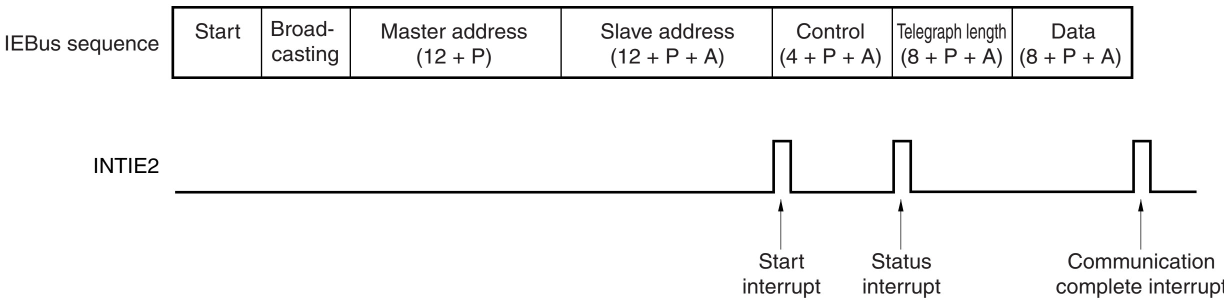
Figure 19-13. Timing of INTIE2 Interrupt Generation in Locked State (for (4) and (5))



Note The telegraph length and data modes are not set in the case of (5) because $\overline{ACK}$ is not returned.

Remark P: Parity bit, A: $\overline{ACK}$/NACK bit

Figure 19-14. Timing of INTIE2 Interrupt Generation in Locked State (for (3))



Remark P: Parity bit, A: $\overline{\text{ACK}}$/NACK bit

(6) IEBus telegraph length register (DLR)

(a) When transmission unit ... Master transmission, slave transmission

The data of this register is reflected in the data transmitted in the telegraph length field and indicates the number of bytes of the transmit data.

This register must be set in advance before transmission.

(b) When reception unit ... Master reception, slave reception

The receive data in the telegraph length field transmitted from the transmission unit is written to this register.

Remark The IEBus telegraph length register consists of a write register and a read register. Consequently, data written to this register cannot be read as is. The data that can be read is the data received during IEBus communication.

After reset: 01H R/W Address: FFFFF3EAH

	7	6	5	4	3	2	1	0
DLR								

Bit								Setting value	Remaining number of communication data bytes
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 byte
0	0	0	0	0	0	1	0	02H	2 bytes
:	:	:	:	:	:	:	:	:	:
0	0	1	0	0	0	0	0	20H	32 bytes
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	FFH	255 bytes
0	0	0	0	0	0	0	0	00H	256 bytes

Cautions 1. If the master issues a request “0H, 4H, 5H, or 6H” to transmit a slave status and lock address (higher 4 bits, lower 8 bits), the contents of this register are set to “01H” by hardware; therefore, the CPU does not have to set this register.

2. In the case of defeat in a bus conflict and a slave status request is received from the unit that won, DLR is fixed to “01H”. Therefore, if a re-request of the master follows, the appointed telegraph length must be set to DLR.

(7) IEBus data register (DR)

The IEBus data register (DR) sets the communication data. Sets the communication data (8 bits) to bits 7 to 0.

Remark The IEBus data register consists of a write register and a read register. Consequently, data written to this register cannot be read as is. The data that can be read is the data received during IEBus communication.

(a) When transmission unit

The data (1 byte) written to the IEBus data register (DR) is stored to the IEBus interface shift register of the IEBus. It is then output from the most significant bit, and an interrupt (INTIE1) is issued to the CPU each time 1 byte has been transmitted. If NACK is received after 1-byte data has been transferred during individual transfer, data is not transferred from DR to the shift register, and the same data is retransmitted. At this time, INTIE1 is not generated. INTIE1 is issued when the IEBus interface shift register stores the IEBus data register value. However, when the last byte and 32nd byte (the last byte of 1 communication frame) is stored in the shift register, INTIE1 is not issued.

(b) When reception unit

One byte of the data received by the shift register of the IEBus interface block is stored to this register.
Each time 1 byte has been correctly received, an interrupt (INTIE1) is issued.
When transmit/receive data is transferred to and from the IEBus data register, using DMA can reduce the CPU processing load.

After reset: 00H R/W Address: FFFFF3ECH

	7	6	5	4	3	2	1	0
DR								

Cautions 1. If the next data is not in time while the transmission unit is set, an underrun occurs, and a communication error interrupt (INTIE2) occurs, stopping transmission.

2. When the IEBus is a receiving unit, if the reading of the data is too late for the next data reception timing, the unit will enter the overrun state. At this time, during individual communication reception, NACK will be returned at the acknowledge bit of the data field, and the master unit will be requested to retransmit the data. If an overrun error occurs during broadcasting communication, the communication error interrupt (INTIE2) is generated.

(8) IEBus unit status register (USR)

After reset: 00H R Address: FFFFF3EEH

	7	<6>	<5>	<4>	<3>	<2>	1	0
USR	0	SLVRQ	ARBIT	ALLTRNS	ACK	LOCK	0	0

SLVRQ	Slave request flag
0	No request from master to slave
1	Request from master to slave

ARBIT	Arbitration result flag
0	Arbitration win
1	Arbitration loss

ALLTRNS	Broadcasting communication flag
0	Individual communication status
1	Broadcasting communication status

ACK	$\overline{\text{ACK}}$ transmission flag
0	NACK transmitted
1	$\overline{\text{ACK}}$ transmitted

LOCK	Lock status flag
0	Unit unlocked
1	Unit locked

(a) Slave request flag (SLVRQ)...Bit 6

A flag indicating whether there has been a slave request from the master.

<Set/reset conditions>

Set: When the unit is requested as a slave (if the received slave address and unit UAR match during individual communication reception, or if the higher 4 bits of the received slave address match or if the received slave address is FFFH during broadcasting communication reception), this flag is set by hardware when the acknowledge period of the slave address field starts.

Reset: This flag is reset by hardware when the unit is not requested as a slave. The reset timing is the same as the set timing. If the unit is requested as a slave immediately after communication has been correctly received (when the SLVRQ bit is set), and if a parity error occurs in the slave address field for that communication, the flag is not reset.

(b) Arbitration result flag (ARBIT)...Bit 5

A flag that indicates the result of arbitration.

<Set/reset conditions>

Set: When the data output by the IEBus unit during the arbitration period does not match the bus line data.

Reset: By the start bit timing.

Cautions 1. The timing at which the arbitration result flag (ARBIT) is reset differs depending on whether the unit outputs a start bit.

- If start bit is output: The flag is reset at the output start timing.
- If start bit is not output: The flag is reset at the detection timing of the start bit (approx. 160 μs after output)

2. The flag is reset at the detection timing of the start bit if the other unit outputs the start bit earlier and the unit does not output the start bit after the master request.

(c) Broadcasting communication flag (ALLTRNS)...Bit 4

A flag indicating whether the unit is performing broadcasting communication. The contents of the flag are updated in the broadcast field of each frame.

Except for initialization (reset) by system reset, the set/reset conditions vary depending on the receive data of the broadcast field bit.

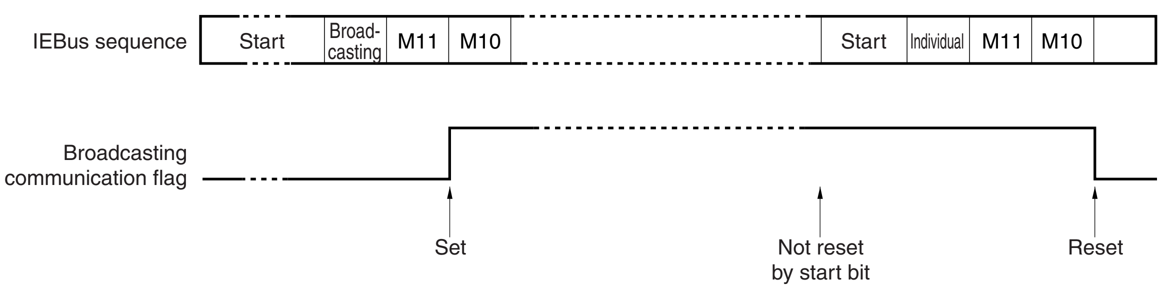
<Set/reset conditions>

Set: When "broadcasting" is received by the broadcast field

Reset: When "individual" is received by the broadcast field, or upon the input of a system reset.

Caution The broadcast flag is updated regardless of whether IEBus is the communication target or not.

Figure 19-15. Example of Broadcasting Communication Flag Operation



(d) ACK transmission flag ($\overline{\text{ACK}}$)...Bit 3

A flag that indicates whether $\overline{\text{ACK}}$ has been transmitted in the $\overline{\text{ACK}}$ period of the $\overline{\text{ACK}}$ field when the IEBus is a receiving unit. The contents of the flag are updated in the $\overline{\text{ACK}}$ period of each frame. However, if the internal circuit is initialized by the occurrence of a parity error, etc., the contents are not updated in the $\overline{\text{ACK}}$ period of that field.

(e) Lock status flag (LOCK)...Bit 2

A flag that indicates whether the unit is locked.

<Set/reset conditions>

Set: When the communication end flag goes low level and the frame end flag goes high level after receipt of a lock specification (3H, 6H, AH, BH) in the control field.

Reset: When the communication enable flag is cleared.

When the communication end flag is set after receipt of a lock release (3H, 6H, AH, BH) in the control field.

Caution Lock specification/release is not possible in broadcasting communication. In the lock status, individual communication from a unit other than the one that requests locking is not acknowledged. However, even communication from a unit other than the one that requests locking is acknowledged as long as the communication is a slave status request.

(9) IEBus interrupt status register (ISR)

This register indicates the status when IEBus issues an interrupt. The ISR is read to generate an interrupt, after which the specified interrupt processing is carried out.

Reset the ISR register after reading it. Until it is reset, the INTIE2 interrupt signal is not generated (nor held pending).

To reset the ISR register, reset each flag, satisfying the reset conditions in Table 19-8.

Table 19-8. Reset Conditions of Flags in ISR Register

Flag Name	Reset Condition	Processing Example
IEERR, STARTF, STATUSF	Byte write operation of ISR register. Any value can be written.	ISR = 00H, etc.
ENDTRNS, ENDFRAM	Set MSTRQ, ENSLVTX, or ENSLVRX flag.	BCR register = 88H or ENSLVTX = 1, etc.

Caution Even if 0 is written to the ENDTRNS or ENDFRAM flag by accessing the ISR register, these flags are not reset. Reset them as described above.

Remark MSTRQ: Bit 6 of the IEBus control register (BCR)
ENSLVTX: Bit 4 of the IEBus control register (BCR)
ENSLVRX: Bit 3 of the IEBus control register (BCR)

After reset: 00H R/W Address: FFFFF3F0H

	7	<6>	<5>	<4>	<3>	<2>	1	0
ISR	0	IEERR	STARTF	STATUSF	ENDTRNS	ENDFRAM	0	0

IEERR	Communication error flag (during communication)
0	No communication error
1	Communication error

STARTF	Start interrupt flag
0	Start interrupt does not occur
1	Start interrupt occurs

STATUSF	Status transmission flag (slave)
0	No slave status/lock address (higher 4 bits, lower 8 bits) transmission request
1	Slave status/lock address (higher 4 bits, lower 8 bits) transmission request

ENDTRNS	Communication end flag
0	Communication does not end after the number of bytes set in the telegraph length field have been transferred
1	Communication ends after the number of bytes set in the telegraph length field have been transferred

ENDFRAM	Frame end flag
0	The frame (transfer of the maximum number of bytes (32 bytes) prescribed by mode 1) does not end
1	The frame (transfer of the maximum number of bytes (32 bytes) prescribed by mode 1) ends

Caution Each of IEERR, STARTF, STATUSF, ENDTRNS, and ENDFRAM are generation triggers for the interrupt request signal (INTIE2) (see Figure 19-16). Because of this, if any one of these interrupt triggers have been set, no new interrupt will be generated by a subsequent trigger. Clear the flag of the interrupt source by the interrupt processing program, before the next interrupt occurs.

(a) Communication error flag (IEERR)...Bit 6

A flag that indicates the detection of an error during communication.

<Set/reset conditions>

Set: The flag is set if a timing error, parity error (except in the data field), NACK reception (except in the data field), underrun error, or overrun error (that occurs during broadcasting communication reception) occurs.

Reset: By software

(b) Start interrupt flag (STARTF)...Bit 5

A flag that indicates whether the interrupt was in the ACK period of the slave address field.

<Set/reset conditions>

Set: In the slave address field, upon a master request. When IEBus is a slave unit, this flag is set upon a request from the master (only if it was a slave request in the locked state from the unit requesting a lock).

Reset: By software

(c) Status transmission flag (STATUSF)...Bit 4

A flag indicating that the transmission status is either the master to slave status, or the lock address (higher 4 bits, lower 8 bits), when IEBus is a slave unit.

<Set/reset conditions>

Set: When 0H, 4H, 5H, or 6H is received in the control field from the master when the IEBus is a slave unit.

Reset: By software

(d) Communication end flag (ENDTRANS)...Bit 3

A flag that indicates whether communication ends after the number of bytes set in the telegraph length field have been transferred.

<Set/reset conditions>

Set: When the value of the SCR counter is 0.

Reset: When the MSTRQ, ENSLVTX, or ENSLVRX flag is set.

(e) Frame end flag (ENDFRAM)...Bit 2

A flag that indicates whether communication ends after the maximum number of bytes (32 bytes) prescribed by mode 1 have been transferred.

<Set/reset conditions>

Set: When the value of the CCR counter is 0.

Reset: When the MSTRQ, ENSLVTX, or ENSLVRX flag is set.

(f) Communication error triggers

- Timing error

Occurrence conditions: Occurs if the high/low level width of the communication bit has shifted from the prescribed value.

Remark: The respective prescribed values are set in the bit processing block and monitored by the internal 8-bit timer. An interrupt is generated when a timing error occurs.

- Parity error

Occurrence conditions: Occurs if the generated parity and the received parity in each field do not match when IEBus is a receiving unit.

Remark: During individual communication, an interrupt is generated if a parity error occurs in a field other than the data field.
During broadcasting communication, an interrupt is generated even if a parity error occurs in the data field.

Restriction: If there is a slave request that has lost in arbitration to a broadcast request, no interrupt is generated, even if a parity error occurs.

- NACK reception error

Occurrence conditions: This error occurs when NACK is received during the ACK period in each of the slave address, control, and telegraph length fields during individual communication, regardless of whether the unit is the master or a slave unit.
A NACK reception error only occurs in individual communication. $\overline{\text{ACK}}$ and NACK are not discriminated in broadcasting communication.

Remark: An interrupt is generated if NACK is received in a field other than the data field.

- Underrun

Occurrence conditions: Occurs during data transmission if there was insufficient time to write the next transmit data to the IEBus data register (DR) before $\overline{\text{ACK}}$ reception.

Remark: An interrupt is generated if an underrun occurs.

- Overrun

Occurrence conditions: The data interrupt request (INTIE1) that stores each byte of data in the IEBus data register (DR), and the DR register is read by DMA or software. An overrun error occurs if this reading processing is late and its timing becomes that of the next data reception.

Remark: In individual communication reception, an acknowledgment is not returned in the ACK period of this data, resulting in the retransmission of the data by the transmit unit. Consequently, the IEBus transfer counter (CCR) is decremented, whereas the IEBus communication success counter (SCR) is not. In broadcasting communication reception, reception is stopped by the occurrence of a communication error (INTIE2), at which time the DR register is not updated. The STATRX flag (bit 1 of the SSR register) also remains set (1) without generating INTIE1. The overrun state is released at the timing of the next data reception following the reading of DR.

(g) Overrun error - supplementary details

(i) When the frame ends in the overrun state during individual communication reception

If the DR register is not read after entering the overrun state and the retransmitted data reaches the maximum number of bytes (32 bytes), the frame end interrupt (INTIE2) is generated. The overrun state is maintained until the DR register is read after the end of the frame.

(ii) If the next reception is started in the case of (i) above, or if the next reception is started without the DR register being read after the final data has been received, regardless of whether the communication is broadcasting or individual

Even if communication to the IEBus unit starts in the overrun state, the cause of the overrun, NACK, is not returned in the ACK period of the slave address, control, or telegraph length field (the DR register is not updated). If the next communication is not to the IEBus unit, the DR register is not updated until it is read. Because the IEBus unit is not a communication target, the data interrupt (INTIE1) and communication error interrupt (INTIE2) are not generated.

(iii) If the next transmission occurs in the overrun state

The data to be transmitted next in the overrun state can be no more than 2 bytes long.

Because the data request interrupt (INTIE1) is not generated, the transmit data cannot be set, resulting in an underrun error. Therefore, clear the overrun status before starting transmission.

(iv) Overrun state release

The overrun state can only be released by reading the DR register or by a system reset. Therefore, be sure to read DR in a communication error interrupt processing program.

(10) IEBus slave status register (SSR)

This register indicates the communication status of the slave unit. After receiving a slave status transmission request from the master, the CPU reads this register, and writes a slave status to the IEBus data register (DR) to transmit the slave status. At this time, the telegraph length is automatically set to “01H” that setting of the IEBus telegraph length register (DLR) is not required (because it is preset by hardware).

Bits 6 and 7 indicate the highest mode supported by the unit, and are fixed to “01H” (mode 1).

After reset: 41H R Address: FFFFF3F2H

	7	6	5	<4>	3	<2>	<1>	<0>
SSR	0	1	0	STATSLV	0	STATLOCK	STATRX	STATTX

STATSLV	Slave transmission status flag
0	Slave transmission stops
1	Slave transmission enabled

STATLOCK	Lock status flag
0	Unlock status
1	Lock status

STATRX	DR receive status
0	Receiving data not stored in DR
1	Receiving data stored in DR

STATTX	DR transmit status
0	Transmission data not stored in DR
1	Transmission data stored in DR

(a) Slave transmission status flag (STATSLV)...Bit 4

Reflects the contents of slave transmission enable flag.

(b) Lock status flag (STATLOCK)...Bit 2

Reflects the contents of locked flag.

(c) DR reception status (STATRX)...Bit 1

This flag indicates the DR reception state.

(d) DR transmission status (STATTX)...Bit 0

This flag indicates the DR transmission state.

(11) IEBus success count register (SCR)

The IEBus success count register (SCR) indicates the number of remaining communication bytes.

This register reads the count value of the counter that decrements the value set by the telegraph length register by $\overline{\text{ACK}}$ in the data field. When the count value has reached “00H”, the communication end flag (ENDTRNS) of the IEBus interrupt status register (ISR) is set.

After reset: 01H R Address: FFFFF3F4H

	7	6	5	4	3	2	1	0
SCR								

Bit								Setting value	Remaining number of communication data bytes
7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	01H	1 byte
0	0	0	0	0	0	1	0	02H	2 bytes
:	:	:	:	:	:	:	:	:	:
0	0	1	0	0	0	0	0	20H	32 bytes
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	FFH	255 bytes
0	0	0	0	0	0	0	0	00H	0 byte (end of communication) or 256 bytes[Note]

Note The actual hard counter consists of 9 bits. When “00H” is read, it cannot be judged whether the remaining number of communication data bytes is 0 (end of communication) or 256. Therefore, either the communication end flag is used, or if “00H” is read when the first interrupt occurs at the beginning of communication, the remaining number of communication data bytes is judged to be 256.

(12) IEBus communication count register (CCR)

The IEBus communication count register (CCR) indicates the number of remaining bytes in the communication byte number specified in the communication mode.

Bits 7 to 0 of the IEBus communication count register (CCR) indicate the number of transfer bytes.

This register reads the count value of the counter that is preset to the maximum number of transmitted bytes (32 bytes) per frame specified in mode 1. Whereas SCR (IEBus communication success counter) is decremented during normal communication ($\overline{\text{ACK}}$), CCR is decremented when 1 byte has been communicated, regardless of whether $\overline{\text{ACK}}$ or NACK. When the count value has reached “00H”, the frame end flag (ENDFRAM) is set.

The maximum number of transfer bytes of the preset value of mode 1 per frame is 20H (32 bytes).

After reset: 20H R Address: FFFFF3F6H

	7	6	5	4	3	2	1	0
CCR								

(13) IEBus clock selection register (IECLK)

This register selects the clock of IEBus. The main clock frequencies that can be used are shown below. Main clock frequencies other than the following cannot be used.

- 6.0 MHz/6.291 MHz
- 12.0 MHz/12.582 MHz

★ **Remark** More IEBus clock types can be selected for the μPD703036H, 703036HY, 70F3036H, 70F3036HY, 703037H, 703037HY, 70F3037H, and 70F3037HY by setting in combination with the IEBus high-speed clock selection register (IEHCLK).

After reset: 00H R/W Address: FFFFF3F8H

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

IECS	IEBus clock selection
0	@ f_{XX} = 6.0 MHz or f_{XX} = 6.291 MHz
1	@ f_{XX} = 12.0 MHz or f_{XX} = 12.582 MHz

★ **(14) IEBus high-speed clock selection register (IEHCLK)**

This register selects the clock of IEBus. The main clock frequencies that can be used are shown below. Main clock frequencies other than the following cannot be used.

- 6.0 MHz/6.291 MHz
- 12.0 MHz/12.582 MHz
- 18.0 MHz/18.873 MHz

Caution The IEHCLK register is available only in the H versions of the V850/SB2 (μPD703036H, 703036HY, 70F3036H, 70F3036HY, 703037H, 703037HY, 70F3037H, and 70F3037HY).

After reset: 00H R/W Address: FFFFF3DEH

	7	6	5	4	3	2	1	0
IEHCLK	0	0	0	0	0	0	0	IECS1

IECS1	IECS[Note]	IEBus clock selection
0	0	@ f_{XX} = 6.0 MHz or f_{XX} = 6.291 MHz
0	1	@ f_{XX} = 12.0 MHz or f_{XX} = 12.582 MHz
1	0	@ f_{XX} = 18.0 MHz or f_{XX} = 18.873 MHz
1	1	Setting prohibited

Note Bit 0 of the IECLK register

19.4 Interrupt Operations of IEBus Controller

19.4.1 Interrupt control block

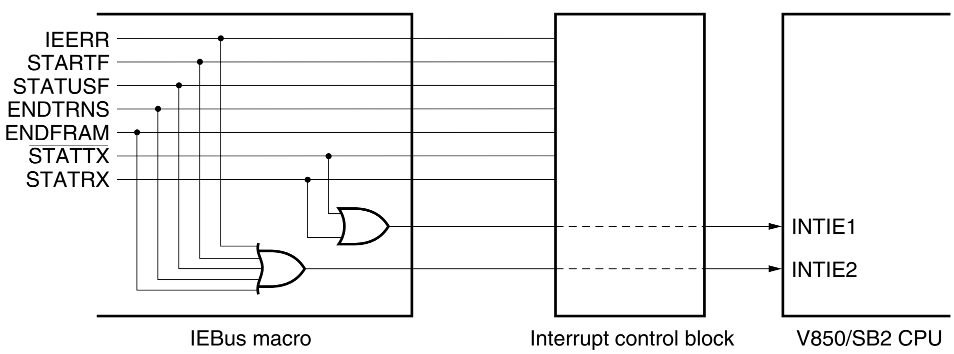
Interrupt request signal

<1>	Communication error	IEERR
<2>	Start interrupt	STARTF
<3>	Status communication	STATUSF
<4>	End of communication	ENDTRNS
<5>	End of frame	ENDFRAM
<6>	Transmit data write request	$\overline{\text{STATTX}}$
<7>	Receive data read request	STATRX

1 through 5 of the above interrupt requests are assigned to the interrupt status register (ISR). For details, refer to **Table 19-9 Interrupt Source List**.

The configuration of the interrupt control block is illustrated below.

Figure 19-16. Configuration of Interrupt Control Block



Cautions 1. OR output of STATRX and $\overline{\text{STATTX}}$ is treated as a DMA transfer start signal (INTIE1).

2. OR output of IEERR, STARTF, STATUSF, ENDTRNS, and ENDFRAM is treated as a vector interrupt request signal (INTIE2) for V850/SB2.

19.4.2 Interrupt source list

The interrupt request signals of the internal IEBus controller in the V850/SB2 can be classified into vector interrupts and DMA transfer interrupts. These interrupt request signals can be specified through software manipulation.

The interrupt sources are listed below.

Table 19-9. Interrupt Source List

Interrupt Source		Condition of Generation		CPU Processing after Generation of Interrupt	Remark
		Unit	Field		
Communication error	Timing error	Master/slave	All fields	Undo communication processing	Communication error is OR output of timing error, parity error, NACK reception, underrun error, and overrun error
	Parity error	Reception	Other than data (individual)		
			All fields (broadcasting)		
	NACK reception	Reception (Transmission)	Other than data (individual)		
	Underrun error	Transmission	Data		
	Overrun error	Reception	Data (broadcasting)		
Start interrupt		Master	Slave/address	Slave request judgment Contention judgment (If loses, remaster processing) Communication preparation processing	Interrupt always occurs if loses in contention during master request
		Slave	Slave/address	Slave request judgment Communication preparation processing	Generated only during slave request
Status transmission		Slave	Control	Refer to transmission processing example such as slave status.	Interrupt occurs regardless of slave transmission enable flag Interrupt occurs if NACK is returned in the control field.
End of communication		Transmission	Data	DMA transfer end processing	Set if SCR is cleared to 0
		Reception	Data	DMA transfer end processing Receive data processing	
End of frame		Transmission	Data	Retransmission preparation processing	Set if CCR is cleared to 0
		Reception	Data	Re-reception preparation processing	
Transmit data write		Transmission	Data	Reading of transmit data[Note].	Set after transfer transmission data to internal shift register This does not occur when the last data is transferred.
Receive data read		Reception	Data	Reading of received data[Note]	Set after normal data reception

Note If DMA transfer or software manipulation is not executed.

19.4.3 Communication error source processing list

The following table shows the occurrence conditions of the communication errors (timing error, NACK reception error, overrun error, underrun error, and parity error), error processing by the internal IEBus controller, and examples of processing by software.

Table 19-10. Communication Error Source Processing List (1/2)

		Timing Error			
Occurrence condition	Unit status	Reception		Transmission	
	Occurrence condition	If bit specification timing is not correct			
	Location of occurrence	Other than data field	Data field	Other than data field	Data field
Broadcasting communication	Hardware processing	• Reception stops. • INTIE2 occurs • To start bit waiting status **Remark** Communication between other units does not end.		• Transmission stops. • INTIE2 occurs • To start bit waiting status	
	Software processing	• Error processing (such as retransmission request)		• Error processing (such as retransmission request	
Individual communication	Hardware processing	• Reception stops. • INTIE2 occurs • NACK is returned. • To start bit waiting status		• Transmission stops. • INTIE2 occurs • To start bit waiting status	
	Software processing	• Error processing (such as retransmission request)		• Error processing (such as retransmission request	

		NACK Reception Error				
Occurrence condition	Unit status	Reception		Transmission		
	Occurrence condition	Unit NACK transmission		Unit NACK transmission		
	Location of occurrence	Other than data field	Data field	Other than data field	Data field	NACK reception of data of 32nd byte
Broadcasting communication	Hardware processing	–	–	–	–	–
	Software processing	–	–	–	–	–
Individual communication	Hardware processing	• Reception stops. • INTIE2 occurs. • To start bit waiting status	• INTIE2 does not occur. • Data retransmitted by other unit is received.	• Reception stops. • INTIE2 occurs. • To start bit waiting status	• INTIE2 does not occur. • Retrans-mission processing	• INTIE2 occurs[Note]. • To start bit waiting status
	Software processing	• Error processing (such as retransmission request)	–	• Error processing (such as retransmission request)	–	• Error processing (such as retransmission request)

Note Both ISR.6 (IEERR) and ISR.2 (ENDFRAM) are set to 1.
To reset them, satisfy the conditions in Table 19-8.

Table 19-10. Communication Error Source Processing List (2/2)

<table>
<tr><th colspan="2"></th><th colspan="2">Overrun Error</th><th colspan="2">Underrun Error</th></tr>
<tr><td rowspan="3">Occurrence condition</td><td>Unit status</td><td colspan="2">Reception</td><td colspan="2">Transmission</td></tr>
<tr><td>Occurrence condition</td><td colspan="2">DR cannot be read in time before the next data is received.</td><td colspan="2">DR cannot be written in time before the next data is transmitted.</td></tr>
<tr><td>Location of occurrence</td><td>Other than data field</td><td>Data field</td><td>Other than data field</td><td>Data field</td></tr>
<tr><td rowspan="2">Broadcasting communication</td><td>Hardware processing</td><td>–</td><td>• Reception stops.
• INTIE2 occurs.
• To start bit waiting status

Remarks 1. Communication between other units does not end.
2. Data cannot be received until the overrun status is cleared.</td><td>–</td><td>• Transmission stops.
• INTIE2 occurs.
• To start bit waiting status</td></tr>
<tr><td>Software processing</td><td>–</td><td>• DR is read and overrun status is cleared.
• Error processing (such as retransmission request)</td><td>–</td><td>• Error processing (such as retransmission request)</td></tr>
<tr><td rowspan="2">Individual communication</td><td>Hardware processing</td><td>–</td><td>• INTIE2 does not occur.
• NACK is returned.
• Data is retransmitted from other unit.

Remark Data cannot be received until overrun status is cleared.</td><td>–</td><td>• Transmission stops.
• INTIE2 occurs.
• To start bit waiting status</td></tr>
<tr><td>Software processing</td><td>–</td><td>• DR is read and overrun status is cleared.
• Error processing (such as retransmission request)</td><td>–</td><td>• Error processing (such as retransmission request)</td></tr>
</table>

<table>
<tr><th colspan="2"></th><th colspan="4">Parity Error</th></tr>
<tr><td rowspan="3">Occurrence condition</td><td>Unit status</td><td colspan="2">Reception</td><td colspan="2">Transmission</td></tr>
<tr><td>Occurrence condition</td><td colspan="2">Received data and received parity do not match.</td><td colspan="2">–</td></tr>
<tr><td>Location of occurrence</td><td>Other than data field</td><td>Data field</td><td>Other than data field</td><td>Data field</td></tr>
<tr><td rowspan="2">Broadcasting communication</td><td>Hardware processing</td><td colspan="2">• Reception stops.
• INTIE2 occurs.
• To start bit waiting status

Remark Communication between other units does not end.</td><td>–</td><td>–</td></tr>
<tr><td>Software processing</td><td colspan="2">• Error processing (such as retransmission)</td><td>–</td><td>–</td></tr>
<tr><td rowspan="2">Individual communication</td><td>Hardware processing</td><td>• Reception stops.
• INTIE2 occurs.
• To start bit waiting status</td><td>• Reception does not stop.
• INTIE2 does not occur.
• NACK is returned.
• Data retransmitted by other unit is received.</td><td>–</td><td>–</td></tr>
<tr><td>Software processing</td><td>• Error processing (such as retransmission request)</td><td>–</td><td>–</td><td>–</td></tr>
</table>

19.5 Interrupt Generation Timing and Main CPU Processing

19.5.1 Master transmission

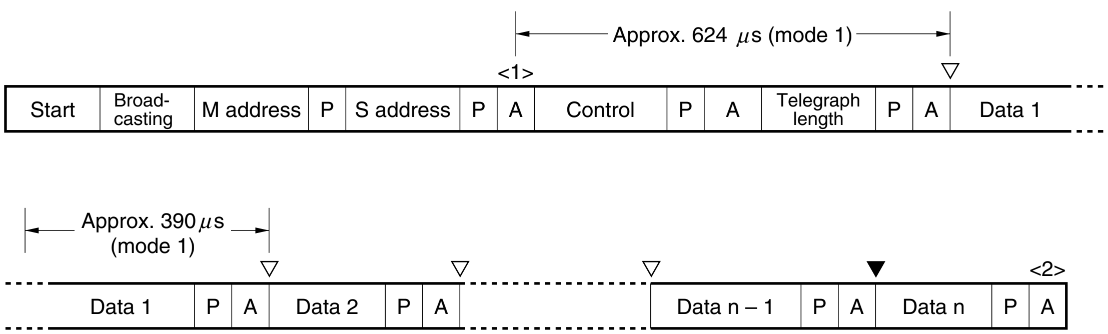
Initial preparation processing:

Sets a unit address, slave address, control data, telegraph length, and the first byte of the transmit data.

Communication start processing:

Sets the bus control register (enables communication, master request, and slave reception).

Figure 19-17. Master Transmission



<1> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing
↓
Judgment of slave request → Slave reception processing (See **19.5.1 (1) Slave reception processing**)
↓
Judgment of contention result → Remaster request processing

<2> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing
↓
Judgment of end of communication → End of communication processing
↓
Judgment of end of frame → Re-communication processing (See **19.5.1 (3) Recommunication processing**)

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **19.5.1 (2) Interrupt (INTIE1) occurrence**)
The transmit data of the second byte and those that follow are written to the IEBus data register (DR) by DMA transfer.
At this time, the data transfer direction is RAM (memory) → SFR (peripheral)

2. ▼: An interrupt (INTIE1) does not occur.

3. n = Final number of data bytes

(1) Slave reception processing

If a slave reception request is confirmed during vector interrupt processing, the data transfer direction of macro service must change from RAM (memory) ‘ SFR (peripheral) to SFR (peripheral) ‘ RAM (memory) until the first data is received. The maximum pending period of this data transfer direction changing processing is about 1040 μs in communication mode 1.

(2) Interrupt (INTIE1) occurrence

If NACK is received from the slave in the data field, an interrupt (INTIE1) is not issued to the CPU, and the same data is retransmitted by hardware.

If the transmit data is not written in time during the period of writing the next data, a communication error interrupt occurs due to occurrence of underrun, and communication ends midway.

(3) Recommunication processing

The vector interrupt processing in <2> judges whether the data has been correctly transmitted within one frame. If the data has not been correctly transmitted (if the number of data to be transmitted in one frame could not be transmitted), the data must be retransmitted in the next frame, or the remainder of the data must be transmitted.

19.5.2 Master reception

Before performing master reception, it is necessary to notify the slave of slave transmission units. Therefore, more than two communication frames are necessary for master reception.

The slave unit prepares the transmit data, set (1) the slave transmission enable flag (ENSLVTX), and waits.

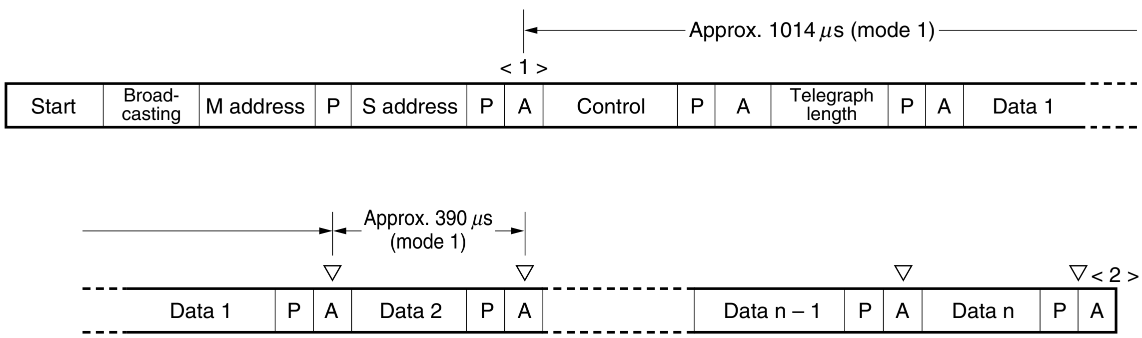
Initial preparation processing:

Sets a unit address, slave address, and control data.

Communication start processing:

Sets the bus control register (enables communication and master request).

Figure 19-18. Master Reception



<1> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing

↓

Judgment of slave request → Slave processing

↓

Judgment of collision result → Remaster request processing

<2> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing

↓

Judgment of end of communication → End of communication processing

↓

Judgment of end of frame → Frame end processing (See **19.5.2 (2) Frame end processing**)

Remarks 1. ∇: Interrupt (INTIE1) occurrence (See **19.5.2 (1) Interrupt (INTIE1) occurrence**)
The receive data stored to the IEBus data register (DR) is read by DMA transfer.
At this time, the data transfer direction is SFR (peripheral) → RAM (memory).

2. n = Final number of data bytes

(1) Interrupt (INTIE1) occurrence

If NACK is transmitted (hardware processing) in the data field, an interrupt (INTIE1) is not issued to the CPU, and the same data is retransmitted from the slave.

If the receive data is not read in time until the next data is received, the hardware automatically transmits NACK.

(2) Frame end processing

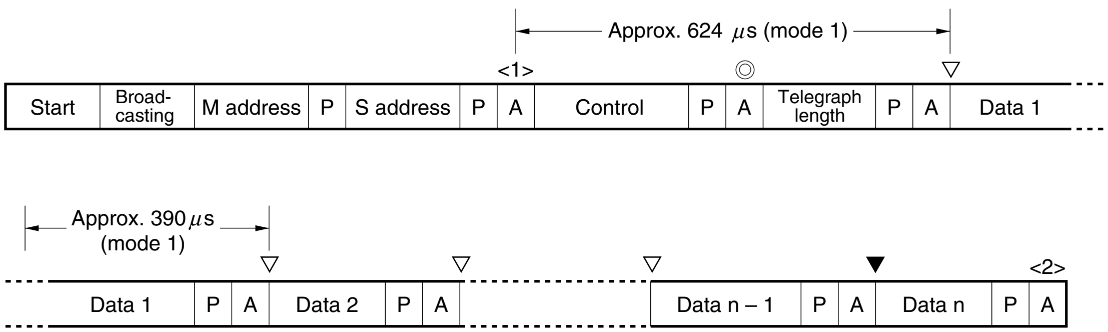
The vector interrupt processing in <2> judges whether the data has been correctly received within one frame. If the data has not been correctly received (if the number of data to be received in one frame could not be received), a request to retransmit the data must be made to the slave in the next communication frame.

19.5.3 Slave transmission

Initial preparation processing:
Sets a unit address, telegraph length, and the first byte of the transmit data.

Communication start processing:
Sets the bus control register (enables communication, slave transmission, and slave reception).

Figure 19-19. Slave Transmission



<1> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing
↓
Judgment of slave request

<2> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing
↓
Judgment of end of communication → End of communication processing
↓
Judgment of end of frame → Frame end processing (See **19.5.3 (2) Frame end processing**)

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **19.5.3 (1) Interrupt (INTIE1) occurrence**).
The transmit data of the second byte and those that follow are written to the IEBus data register (DR) by DMA transfer.
At this time, the data transfer direction is RAM (memory) → SFR (peripheral).

2. ▼: An interrupt (INTIE1) does not occur.

3. ◎: Interrupt (INTIE2) occurrence
An interrupt occurs only when 0H, 4H, 5H, or 6H is received in the control field in the slave status (for the slave status response operation during locked, refer to **19.3.2 (5) IEBus control data register (CDR)**).

4. n = Final number of data bytes

(1) Interrupt (INTIE1) occurrence

If NACK is received from the master in the data field, an interrupt (INTIE1) is not issued to the CPU, and the same data is retransmitted by hardware.

If the transmit data is not written in time during the period of writing the next data, a communication error interrupt occurs due to occurrence of underrun, and communication is abnormally ended.

(2) Frame end processing

The vector interrupt processing in <2> judges whether the data has been correctly transmitted within one frame. If the data has not been correctly transmitted (if the number of data to be transmitted in one frame could not be transmitted), the data must be retransmitted in the next frame, or the continuation of the data must be transmitted.

19.5.4 Slave reception

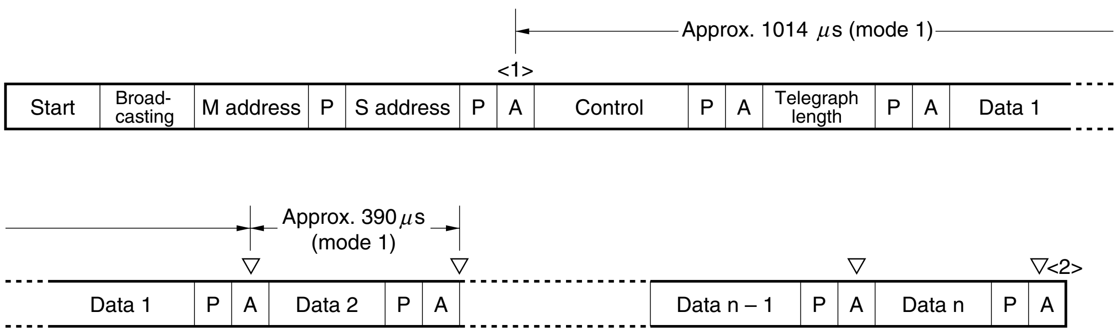
Initial preparation processing:

Sets a unit address.

Communication start processing:

Sets the bus control register (enables communication, disables slave transmission, and enables slave reception).

Figure 19-20. Slave Reception



<1> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing

↓

Judgment of slave request → Slave processing

<2> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing

↓

Judgment of end of communication → End of communication processing

↓

Judgment of end of frame → Frame end processing (See **19.5.4 (2) Frame end processing**).

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **19.5.4 (1) Interrupt (INTIE1) occurrence**).
The receive data stored to the IEBus data register (DR) is read by DMA transfer.
At this time, the data transfer direction is SFR (peripheral) → RAM (memory).

2. n = Final number of data bytes

(1) Interrupt (INTIE1) occurrence

If NACK is transmitted in the data field, an interrupt (INTIE1) is not issued to the CPU, and the same data is retransmitted from the master.

If the receive data is not read in time until the next data is received, NACK is automatically transmitted.

(2) Frame end processing

The vector interrupt processing in <2> judges whether the data has been correctly received within one frame.

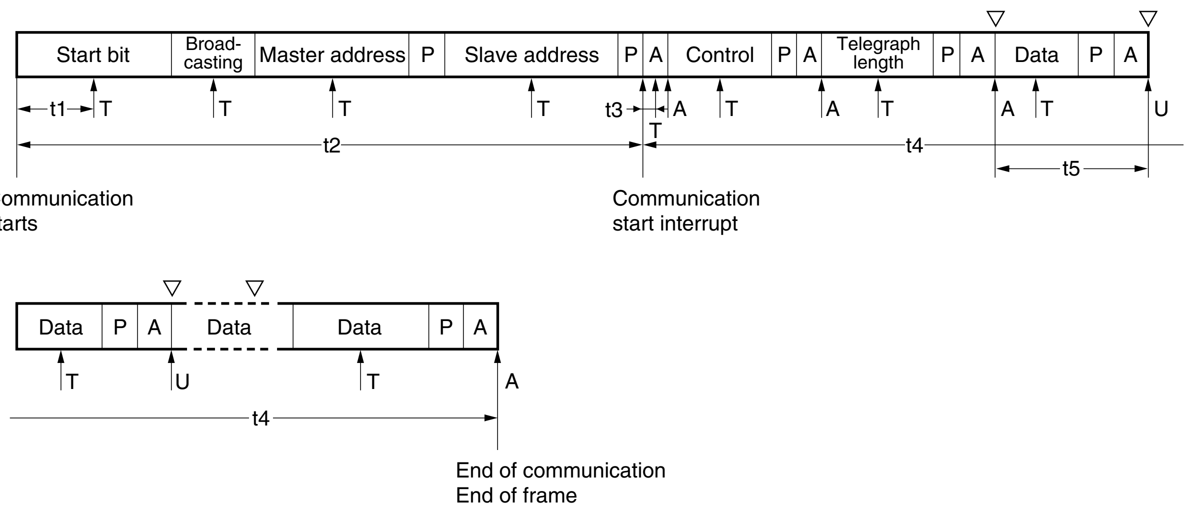
19.5.5 Interval of occurrence of interrupt for IEBus control

Each control interrupt must occur at each point of communication and perform the necessary processing until the next interrupt occurs. Therefore, the CPU must control the IEBus control block, taking the shortest time of this interrupt into consideration.

The locations at which the following interrupts may occur are indicated by ↑ in the field where it may occur. ↑ does not mean that the interrupt occurs at each of the points indicated by ↑. If an error interrupt (timing error, parity error, or $\overline{\text{ACK}}$ error) occurs, the IEBus internal circuit is initialized. As a result, the following interrupt does not occur in that communication frame.

(1) Master transmission

Figure 19-21. Master Transmission (Interval of Interrupt Occurrence)



Remarks 1. T: Timing error
A: $\overline{\text{ACK}}$ error
U: Underrun error
▽: Data set interrupt (INTIE1)

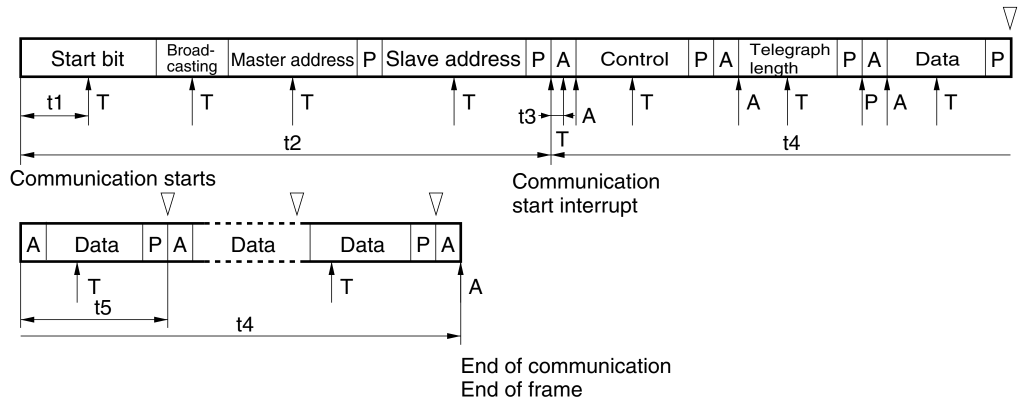
2. End of frame occurs at the end of 32-byte data.

(IEBus: at 6.29 MHz)

Item	Symbol	MIN.	Unit
Communication starts – timing error	t1	Approx. 93	μs
Communication starts – communication start interrupt	t2	Approx. 1282	μs
Communication start interrupt – timing error	t3	Approx. 15	μs
Communication start interrupt – end of communication	t4	Approx. 1012	μs
Transmission data request interrupt interval	t5	Approx. 375	μs

(2) Master reception

Figure 19-22. Master Reception (Interval of Interrupt Occurrence)



Remarks 1. T: Timing error
P: Parity error
A: $\overline{\text{ACK}}$ error
▽: Data set interrupt (INTIE1)

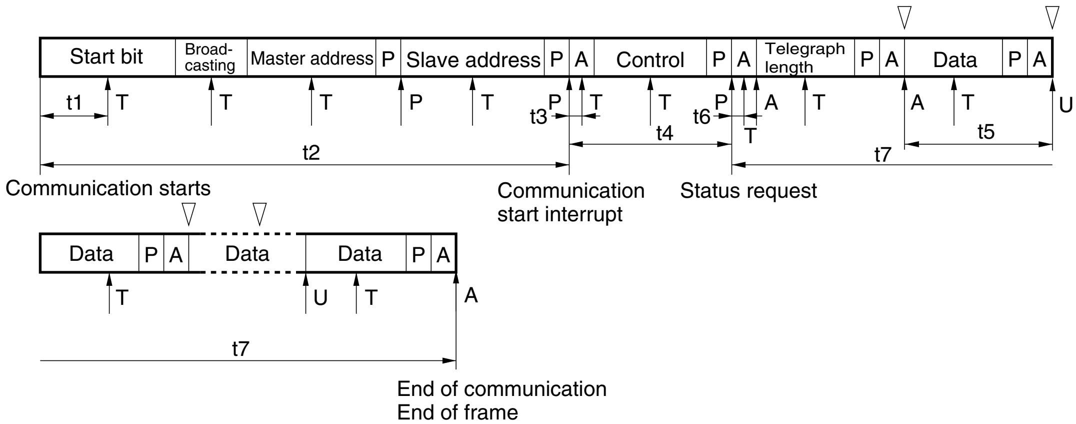
2. End of frame occurs at the end of 32-byte data.

(IEBus: at 6.29 MHz)

Item	Symbol	MIN.	Unit
Communication starts – timing error	t1	Approx. 93	μs
Communication starts – communication start interrupt	t2	Approx. 1282	μs
Communication start interrupt – timing error	t3	Approx. 15	μs
Communication start interrupt – end of communication	t4	Approx. 1012	μs
Receive data read interval	t5	Approx. 375	μs

(3) Slave transmission

Figure 19-23. Slave Transmission (Interval of Interrupt Occurrence)



Remarks 1. T: Timing error
P: Parity error
A: $\overline{\text{ACK}}$ error
U: Underrun error
▽: Data set interrupt (INTIE1)

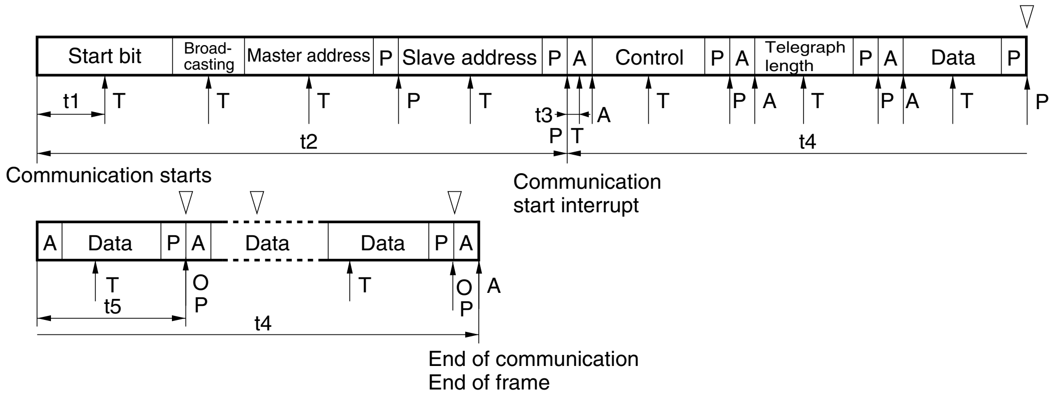
2. End of frame occurs at the end of 32-byte data.

(IEBus: at 6.29 MHz)

Item	Symbol	MIN.	Unit
Communication starts – timing error	t1	Approx. 96	μs
Communication starts – communication start interrupt	t2	Approx. 1192	μs
Communication start interrupt – timing error	t3	Approx. 15	μs
Communication start interrupt – status request	t4	Approx. 225	μs
Transmission data request interrupt interval	t5	Approx. 375	μs
Status request – timing error	t6	Approx. 15	μs
Status request – end of communication	t7	Approx. 787	μs

(4) Slave reception

Figure 19-24. Slave Reception (Interval of Interrupt Occurrence)



Remarks 1. T: Timing error

P: Parity error

A: $\overline{\text{ACK}}$ error

O: Overrun error

▽: Data set interrupt (INTIE1)

2. End of frame occurs at the end of 32-byte data.

(IEBus: at 6.29 MHz)

Item	Symbol	MIN.	Unit
Communication starts – timing error	t1	Approx. 96	μs
Communication starts – communication start interrupt	t2	Approx. 1192	μs
Communication start interrupt – timing error	t3	Approx. 15	μs
Communication start interrupt – end of communication	t4	Approx. 1012	μs
Receive data read interval	t5	Approx. 375	μs

Absolute Maximum Ratings (T_A = 25°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	V_{DD} pin	–0.5 to +7.0	V
	V_{PP}	Flash memory versions only, **Note 1**	–0.5 to +8.5	V
	AV_{DD}	AV_{DD} pin	–0.5 to +7.0	V
	BV_{DD}	BV_{DD} pin	–0.5 to +7.0	V
	EV_{DD}	EV_{DD} pin	–0.5 to +7.0	V
	AV_{SS}	AV_{SS} pin	–0.5 to +0.5	V
	BV_{SS}	BV_{SS} pin	–0.5 to +0.5	V
	EV_{SS}	EV_{SS} pin	–0.5 to +0.5	V
Input voltage	V_{I1}	**Note 2** (BV_{DD} pin)	–0.5 to BV_{DD} + 0.5 Note 5	V
	V_{I2}	**Note 3**, $\overline{\text{RESET}}$ (EV_{DD} pin)	–0.5 to EV_{DD} + 0.5 Note 5	V
Analog input voltage	V_{IAN}	**Note 4** (AV_{DD} pin)	–0.5 to AV_{DD} + 0.5 Note 5	V
Analog reference input voltage	AV_{REF}	AV_{REF} pin	–0.5 to AV_{DD} + 0.5 Note 5	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, 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, CLKOUT	–25	mA
		Total for P50 to P57, P60 to P65	–25	mA
Output voltage	V_{O1}	**Note 2**, CLKOUT (BV_{DD} pin)	–0.5 to BV_{DD} + 0.5 Note 5	V
	V_{O2}	**Note 3** (EV_{DD} pin)	–0.5 to EV_{DD} + 0.5 Note 5	V
Operating ambient temperature	T_A	Normal operation mode	–40 to +85	°C
		Flash memory programming mode Note 6	10 to +85	°C
		Flash memory programming mode Note 7	–20 to +85	
Storage temperature	T_{stg}	Mask ROM versions	–65 to +150	°C
		Flash memory versions	–40 to +125	°C

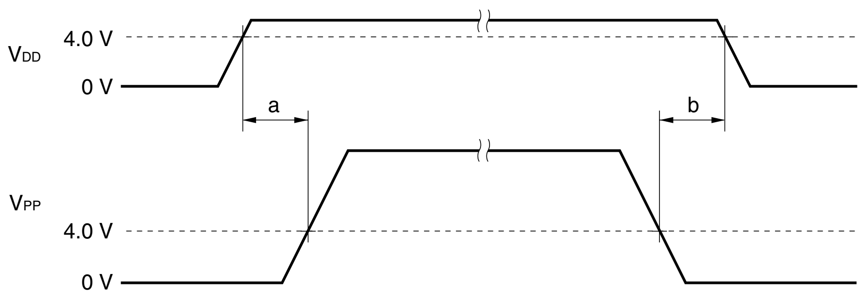
Notes 1. Make sure that the following conditions of the V_{PP} voltage application timing are satisfied when the flash memory is written.

- **When supply voltage rises**

  V_{PP} must exceed V_{DD} 1 ms or more after V_{DD} has reached the lower-limit value (4.0 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 (4.0 V) of the operating voltage range of V_{DD} (see b in the figure below).



2. Ports 4, 5, 6, 9, and their alternate-function pins

3. Ports 0, 1, 2, 3, 10, 11, and their alternate-function pins

4. Ports 7, 8, and their alternate-function pins

5. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

6. K rank products of the μPD70F3033A, 70F3033AY and K and E rank products of the μPD70F3035A, 70F3035AY

(The rank is indicated by the letter appearing as the 5th digit from the left in the lot number.)

7. E rank products of the μPD70F3033A and 70F3033AY, P rank products of the μPD70F3035A and 70F3035AY, and the μPD70F3032A, 70F3032AY, 70F3037A, 70F3037AY, 70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY, 70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, and 70F3037HY

(The rank is indicated by the letter appearing as the 5th digit from the left in the lot number.)

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Capacitance (T_A = 25°C, V_{DD} = AV_{DD} = BV_{DD} = EV_{DD} = V_{SS} = AV_{SS} = BV_{SS} = EV_{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 Frequency (f_{XX})		V_{DD}	AV_{DD}		BV_{DD}	EV_{DD}	Remark
			Note 1	**Note 2**			
2 to 20 MHz (V850SB1)		4.0 to 5.5 V	4.5 to 5.5 V	4.0 to 5.5 V	4.0 to 5.5 V	4.0 to 5.5 V	**Note 3**
2 to 17 MHz (V850/SB1)		4.0 to 5.5 V	4.5 to 5.5 V	4.0 to 5.5 V	3.0 to 5.5 V	3.0 to 5.5 V	
2 to 19 MHz (H versions of V850/SB2)		4.0 to 5.5 V	4.5 to 5.5 V	4.0 to 5.5 V	4.0 to 5.5 V	4.0 to 5.5 V	
2 to 17 MHz (H versions of V850/SB2)		4.0 to 5.5 V	4.5 to 5.5 V	4.0 to 5.5 V	3.0 to 5.5 V	3.0 to 5.5 V	
2 to 13 MHz (A and B versions of V850/SB2)		4.0 to 5.5 V	4.5 to 5.5 V	4.0 to 5.5 V	3.0 to 5.5 V	3.0 to 5.5 V	
32.768 kHz	Other than IDLE mode	4.0 to 5.5 V	4.5 to 5.5 V	4.0 to 5.5 V	3.0 to 5.5 V	3.0 to 5.5 V	–
	IDLE mode	3.5 to 5.5 V	–	4.0 to 5.5 V	3.0 to 5.5 V	3.0 to 5.5 V	**Note 4**

Notes 1. When A/D converter is used

2. When A/D converter is not used

3. In STOP mode (when only watch timer is operating), V_{DD} = 3.5 to 5.5 V. Shifting to STOP mode or restoring from STOP mode must be performed at V_{DD} = 4.0 V min.

4. Shifting to IDLE mode or restoring from IDLE mode must be performed at V_{DD} = 4.0 V min.

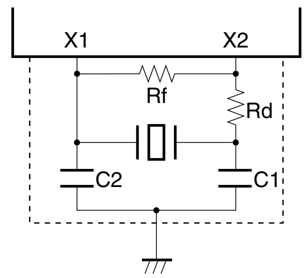
(2) CPU operating frequency

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU operating frequency	f_{CPU}	Main clock operation (V850/SB1)	0.25		20	MHz
		Main clock operation (H versions of V850/SB2)	0.25		19	MHz
		Main clock operation (A and B versions of V850/SB2)	0.25		13	MHz
		Subclock operation		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 (V850/SB1)	f_{XX}		2		20	MHz
Oscillation frequency (H versions of V850/SB2)	f_{XX}		2		19	MHz
Oscillation frequency (A and B versions of V850/SB2)	f_{XX}		2		13	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).

Cautions 1. The main clock oscillator operates on the output voltage of the on-chip regulator (3.0 V/3.3 V (for details, refer to CHAPER 16 REGULATOR). External clock input is prohibited.

2. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

3. Ensure that the duty of oscillation waveform is between 5.5 and 4.5.

4. Sufficiently evaluate the matching between the V850/SB1 and V850/SB2 devices and the resonator.

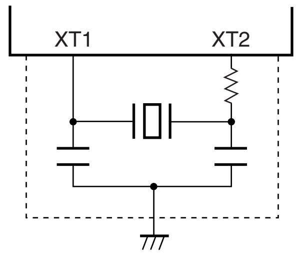
(i) Murata Mfg. Co., Ltd.: Ceramic resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	Oscillation Frequency f_{XX} (MHz)	Recommended Circuit Constant				Oscillation Voltage Range	
			C1 (pF)	C2 (pF)	Rf (kΩ)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Mfg. Co., Ltd. (A, B versions of V850/SB2)	CSTLS6M29G53-B0	6.290	On-chip	On-chip	–	0	4.0	5.5
	CSTCR6M29G53-R0		On-chip	On-chip	–	0	4.0	5.5
	CSTLA12M5T55001-B0	12.583	On-chip	On-chip	–	0	4.0	5.5
	CSTCV12M5T54J01-R0		On-chip	On-chip	–	0	4.0	5.5
Murata Mfg. Co., Ltd. (V850/SB1)	CSTLS8M00G56-B0	8.00	On-chip	On-chip	–	0	4.0	5.5
	CSTCC8M00G56-R0		On-chip	On-chip	–	0	4.0	5.5
	CSTLA12M5T55-B0	12.5	On-chip	On-chip	–	0	4.0	5.5
	CSTCV12M5T54J-R0		On-chip	On-chip	–	0	4.0	5.5
	CSALS16M0X55-B0	16.00	10	10	–	0	4.0	5.5
	CSTCV16M0X51J-R0		On-chip	On-chip	–	0	4.0	5.5
	CSTLS20M0X51-B0	20.00	On-chip	On-chip	–	0	4.0	5.5
	CSTCW20M0X51-R0		On-chip	On-chip	22 k	0	4.0	5.5

Caution This oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer.
If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.
The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the V850/SB1 and V850/SB2 so that the internal operating conditions are within the specifications of the DC and AC characteristics.

(2) Subclock oscillator (T_A = –40 to +85°C)

(a) Connection of crystal resonator



Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{XT}		32	32.768	35	kHz
Oscillation stabilization time	–			10		s

Cautions 1. The subclock oscillator operates on the output voltage of the on-chip regulator (3.0 V/3.3 V (for details, refer to CHAPTER 16 REGULATOR)).
External clock input is prohibited.

2. When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

3. Sufficiently evaluate the matching between the V850/SB1 and V850/SB2 devices and the resonator.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, AV_{DD} = 4.5 to 5.5 V (when A/D converter is used), AV_{DD} = 4.0 to 5.5 V (when A/D converter is not used), V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**	4.0 V ≤ BV_{DD} ≤ 5.5 V	$0.7BV_{DD}$		BV_{DD}	V
			3.0 V ≤ BV_{DD} < 4.0 V	$0.8BV_{DD}$		BV_{DD}	V
	V_{IH2}	**Note 2**	4.0 V ≤ EV_{DD} ≤ 5.5 V	$0.7EV_{DD}$		EV_{DD}	V
			3.0 V ≤ EV_{DD} < 4.0 V	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	**Note 3**, $\overline{\text{RESET}}$	4.0 V ≤ EV_{DD} ≤ 5.5 V	$0.7EV_{DD}$		EV_{DD}	V
			3.0 V ≤ EV_{DD} < 4.0 V	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH4}	**Note 4**		$0.7AV_{DD}$		AV_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**		BV_{SS}		$0.3BV_{DD}$	V
	V_{IL2}	**Note 2**		EV_{SS}		$0.3EV_{DD}$	V
	V_{IL3}	**Note 3**, $\overline{\text{RESET}}$		EV_{SS}		$0.3EV_{DD}$	V
	V_{IL4}	**Note 4**		AV_{SS}		$0.3AV_{DD}$	V
Output voltage, high	V_{OH1}	**Note 1**, CLKOUT	3.0 V ≤ BV_{DD} ≤ 5.5 V, I_{OH} = –100 μA	BV_{DD} – 0.5			V
			4.0 V ≤ BV_{DD} ≤ 5.5 V, I_{OH} = –3 mA	BV_{DD} – 1.0			V
	V_{OH2}	**Notes 2, 3**	3.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = –100 μA	EV_{DD} – 0.5			V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = –3 mA	EV_{DD} – 1.0			V
Output voltage, low	V_{OL}	I_{OL} = 3 mA, 3.0 V ≤ BV_{DD}, EV_{DD} ≤ 5.5 V				0.5	V
		I_{OL} = 3 mA, 4.0 V ≤ BV_{DD}, EV_{DD} ≤ 5.5 V				0.4	V
V_{PP} power supply voltage	V_{PP1}	Normal operation (A and B versions of V850/SB2)		0		0.54	V
		Normal operation (V850/SB1 and H versions of V850/SB2)		0		0.6	V
Input leakage current, high	I_{LIH}	V_I = V_{DD} = BV_{DD} = EV_{DD} = AV_{DD}				5	μA
Input leakage current, low	I_{LIL}	V_I = 0 V				–5	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD} = BV_{DD} = EV_{DD} = AV_{DD}				5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–5	μA
Pull-up resistor	R_L	V_{IN} = 0 V		10	30	100	kΩ

Notes 1. Ports 4, 5, 6, 9, and their alternate-function pins

2. P11, P14, P21, P24, P34, P35, P110 to P113, and their alternate-function pins

3. P00 to P07, P10, P12, P13, P15, P20, P22, P23, P25 to P27, P30 to P33, P36, P37, P100 to P107, and their alternate-function pins

4. Ports 7, 8, and their alternate-function pins

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, AV_{DD} = 4.5 to 5.5 V (when A/D converter is used), AV_{DD} = 4.0 to 5.5 V (when A/D converter is not used), V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current	μPD703030B, 703030BY, 703031A, 703031AY, 703031B, 703031BY, 703032A, 703032AY, 703032B, 703032BY, 703033A, 703033AY, 703033B, 703033BY, 703036H, 703036HY, 703037H, 703037HY	I_{DD1}	In normal operation mode[Note 1]			25	40	mA
		I_{DD2}	In HALT mode[Note 1]			10	20	mA
		I_{DD3}	In IDLE mode[Note 2]	Watch timer operating		1	4	mA
		I_{DD4}	In STOP mode	Watch timer, subclock oscillator operating		**Note 3**	70	μA
				Subclock oscillator stopped, XT1 = V_{SS}		**Note 4**	70	μA
		I_{DD5}	In normal operation mode (subclock operation)[Note 5]			50	150	μA
		I_{DD6}	In IDLE mode (subclock operation)[Note 5]			13	70	μA
	μPD703034A, 703034AY, 703034B, 703034BY, 703035A, 703035AY, 703035B, 703035BY, 703037A, 703037AY	I_{DD1}	In normal operation mode[Note 6]			15	25	mA
		I_{DD2}	In HALT mode[Note 6]			6	13	mA
		I_{DD3}	In IDLE mode[Note 7]	Watch timer operating		1	4	mA
		I_{DD4}	In STOP mode	Watch timer, subclock oscillator operating		**Note 8**	70	μA
				Subclock oscillator stopped, XT1 = V_{SS}		**Note 9**	70	μA
		I_{DD5}	In normal operation mode (subclock operation)[Note 5]			50	150	μA
		I_{DD6}	In IDLE mode (subclock operation)[Note 5]			13	70	μA

Notes 1. f_{CPU} = f_{XX} = 20 MHz, all peripheral functions operating (f_{XX} = 19 MHz in the μPD703036H, 703036HY, 703037H, and 703037HY)

2. f_{XX} = 20 MHz (f_{XX} = 19 MHz in the μPD703036H, 703036HY, 703037H, and 703037HY)

3. I_{DD4} = 13 μA (μPD703031A, 703031AY, 703032A, 703032AY, 703033A, 703033AY)
I_{DD4} = 10 μA (μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY, 703036H, 703036HY, 703037H, 703037HY)

4. I_{DD4} = 8 μA (μPD703031A, 703031AY, 703032A, 703032AY, 703033A, 703033AY)
I_{DD4} = 5 μA (μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY, 703036H, 703036HY, 703037H, 703037HY)

5. f_{CPU} = f_{XT} = 32.768 kHz, main clock oscillator stopped

6. f_{CPU} = f_{XX} = 13 MHz, all peripheral functions operating

7. f_{XX} = 13 MHz

8. I_{DD4} = 13 μA (μPD703034A, 703034AY, 703035A, 703035AY, 703037A, 703037AY)
I_{DD4} = 10 μA (μPD703034B, 703034BY, 703035B, 703035BY)

9. I_{DD4} = 8 μA (μPD703034A, 703034AY, 703035A, 703035AY, 703037A, 703037AY)
I_{DD4} = 5 μA (μPD703034B, 703034BY, 703035B, 703035BY)

Remark TYP. values are reference values for when T_A = 25°C, V_{DD} = BV_{DD} = EV_{DD} = AV_{DD} = 5.0 V. The current consumed by the output buffer is not included.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, AV_{DD} = 4.5 to 5.5 V (when A/D converter is used), AV_{DD} = 4.0 to 5.5 V (when A/D converter is not used), V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current	μPD70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY, 70F3033A, 70F3033AY, 70F3033B, 70F3033BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY	I_{DD1}	In normal operation mode[Note 1]	**Note 2**		33	60	mA
				Note 3		37	65	mA
				Note 4		42	70	mA
		I_{DD2}	In HALT mode[Note 1]	**Note 2**		10	20	mA
				Note 3		12	24	mA
				Note 4		14	28	mA
		I_{DD3}	In IDLE mode[Note 5]	Watch timer operating		1	4	mA
		I_{DD4}	In STOP mode	Watch timer, subclock oscillator operating		**Note 6**	100	μA
				Subclock oscillator stopped, XT1 = V_{SS}		**Note 7**	100	μA
		I_{DD5}	In normal operation mode (subclock operation)[Note 8]	**Notes 2, 3**		200	600	μA
				Note 4		300	900	μA
		I_{DD6}	In IDLE mode (subclock operation)[Note 8]	**Notes 2, 3**		90	180	μA
				Note 4		170	340	μA

Notes 1. f_{CPU} = f_{XX} = 20 MHz, all peripheral functions operating (f_{XX} = 19 MHz in the μPD70F3036H, 70F3036HY, 70F3037H, and 70F3037HY)

2. μPD70F3033A, 70F3033AY, 70F3033B, 70F3033BY

3. μPD70F3030B, 70F3030BY, 70F3036H, 70F3036HY

4. μPD70F3032A, 70F3032AY, 70F3032B, 70F3032BY, 70F3037H, 70F3037HY

5. f_{XX} = 20 MHz (f_{XX} = 19 MHz in the μPD70F3036H, 70F3036HY, 70F3037H, and 70F3037HY)

6. I_{DD4} = 13 μA (μPD70F3032A, 70F3032AY, 70F3033A, 70F3033AY)
I_{DD4} = 10 μA (μPD70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY)

7. I_{DD4} = 8 μA (μPD70F3032A, 70F3032AY, 70F3033A, 70F3033AY)
I_{DD4} = 5 μA (μPD70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY)

8. f_{CPU} = f_{XT} = 32.768 kHz, main clock oscillator stopped

Remark TYP. values are reference values for when T_A = 25°C, V_{DD} = BV_{DD} = EV_{DD} = AV_{DD} = 5.0 V. The current consumed by the output buffer is not included.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, AV_{DD} = 4.5 to 5.5 V (when A/D converter is used), AV_{DD} = 4.0 to 5.5 V (when A/D converter is not used), V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current	μPD70F3035A, 70F3035AY, 70F3035B, 70F3035BY, 70F3037A, 70F3037AY	I_{DD1}	In normal operation mode[Note 1]	**Note 2**		25	48	mA
				Note 3		30	58	mA
		I_{DD2}	In HALT mode[Note 1]	**Note 2**		7	15	mA
				Note 3		9	20	mA
		I_{DD3}	In IDLE mode[Note 4]	Watch timer operating		1	4	mA
		I_{DD4}	In STOP mode	Watch timer, subclock oscillator operating		**Note 5**	100	μA
				Subclock oscillator stopped, XT1 = V_{SS}		**Note 6**	100	μA
		I_{DD5}	In normal operation mode (subclock operation)[Note 7]	**Note 2**		200	600	μA
				Note 3		300	900	μA
		I_{DD6}	In IDLE mode (subclock operation)[Note 7]	**Note 2**		90	180	μA
				Note 3		170	340	μA

Notes 1. f_{CPU} = f_{XX} = 13 MHz, all peripheral functions operating

2. μPD70F3035A, 70F3035AY, 70F3035B, 70F3035BY

3. μPD70F3037A, 70F3037AY

4. f_{XX} = 13 MHz

5. I_{DD4} = 13 μA (μPD70F3035A, 70F3035AY, 70F3037A, 70F3037AY)
I_{DD4} = 10 μA (μPD70F3035B, 70F3035BY)

6. I_{DD4} = 8 μA (μPD70F3035A, 70F3035AY, 70F3037A, 70F3037AY)
I_{DD4} = 5 μA (μPD70F3035B, 70F3035BY)

7. f_{CPU} = f_{XT} = 32.768 kHz, main clock oscillator stopped

Remark TYP. values are reference values for when T_A = 25°C, V_{DD} = BV_{DD} = EV_{DD} = AV_{DD} = 5.0 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} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode (all functions not operating) T_A = –40 to +85°C			2.7 Note 1		5.5	V
		STOP mode (all functions not operating) T_A = –40 to +45°C			2.5 Note 1		5.5	V
		STOP mode (all functions not operating) T_A = –10 to +45°C			2.0 Note 1		5.5	V
Data retention current	I_{DDDR}	V_{DD} = V_{DDDR}, XT1 = V_{SS} (Subclock stopped)	Mask ROM versions	**Note 2**		5	70	μA
				Note 3		8	70	μA
			Flash memory versions	**Note 4**		5	100	μA
				Note 5		8	100	μA
Power supply voltage rise time	t_{RVD}				200			μs
Power supply voltage fall time	t_{FVD}				200			μs
Power 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			0.9V_{DDDR}		V_{DDDR}	V
Data retention low-level input voltage	V_{ILDR}	All input ports			0		0.1V_{DDDR}	V

Notes 1. In STOP mode (when only watch timer is operating), V_{DD} = 3.5 to 5.5 V. Shifting to STOP mode or restoring from STOP mode must be performed at V_{DD} = 4.0 V min.

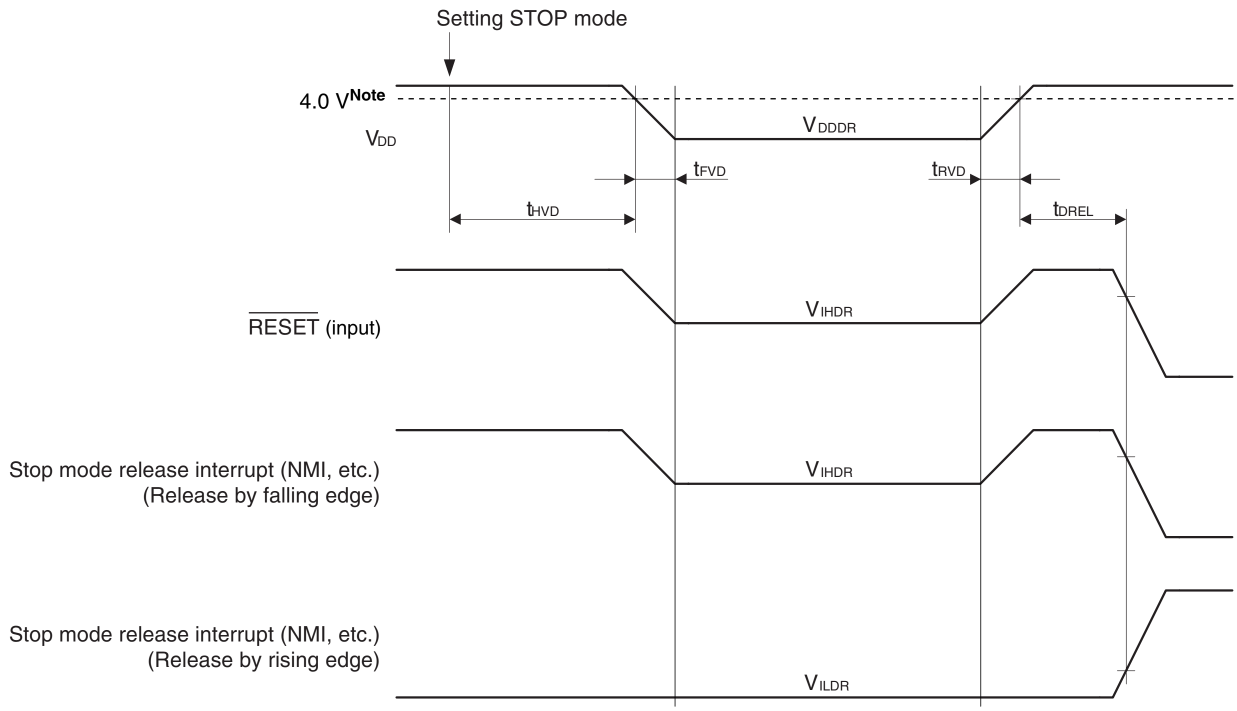
2. μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY, 703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY, 703037H, 703037HY

3. μPD703031A, 703031AY, 703032A, 703032AY, 703033A, 703033AY, 703034A, 703034AY, 703035A, 703035AY, 703037A, 703037AY

4. μPD70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY, 70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY

5. μPD70F3032A, 70F3032AY, 70F3033A, 70F3033AY, 70F3035A, 70F3035AY, 70F3037A, 70F3037AY

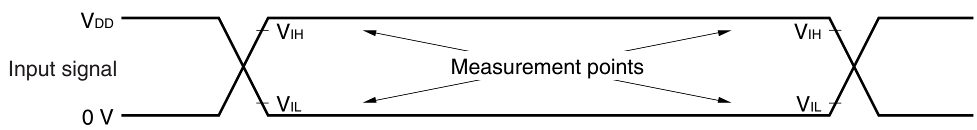
Remark TYP. values are reference values for when $T_A = 25°C$.



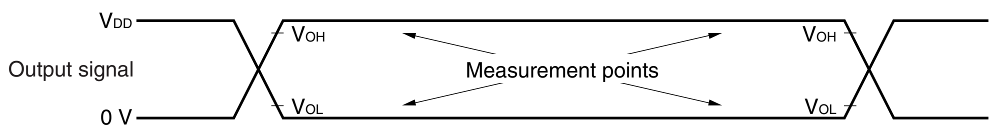
Note V_{DD} = 4.0 V indicates the minimum operating voltage of the V850/SB1 and V850/SB2.

**AC Characteristics (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V,
AV_{DD} = 4.5 to 5.5 V (when A/D converter is used),
AV_{DD} = 4.0 to 5.5 V (when A/D converter is not used), V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)**

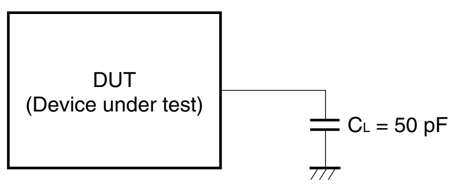
AC Test Input Measurement Point (V_{DD}: EV_{DD}, BV_{DD}, AV_{DD})



AC Test Output Measurement Points (V_{DD}: EV_{DD}, BV_{DD})



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.

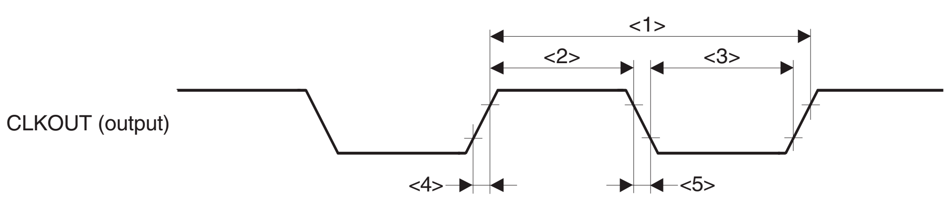
(1) Clock timing

(a) T_A = –40 to +85°C, V_{DD} = BV_{DD} = 4.0 to 5.5 V, EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
CLKOUT output cycle	<1>	t_{CYK}	V850/SB1	50 ns	31.2 μs	
			H versions of V850/SB2	52.6 ns	31.2 μs	
			A and B versions of V850/SB2	76.9 ns	31.2 μs	
CLKOUT high-level width	<2>	t_{WKH}		$0.4t_{CYK} - 12$		ns
CLKOUT low-level width	<3>	t_{WKL}		$0.4t_{CYK} - 12$		ns
CLKOUT rise time	<4>	t_{KR}			12	ns
CLKOUT fall time	<5>	t_{KF}			12	ns

(b) T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = 3.0 to 4.0 V, EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
CLKOUT output cycle	<1>	t_{CYK}	V850/SB1	58.8 ns	31.2 μs	
			H versions of V850/SB2	58.8 ns	31.2 μs	
			A and B versions of V850/SB2	76.9 ns	31.2 μs	
CLKOUT high-level width	<2>	t_{WKH}		$0.4t_{CYK} - 15$		ns
CLKOUT low-level width	<3>	t_{WKL}		$0.4t_{CYK} - 15$		ns
CLKOUT rise time	<4>	t_{KR}			15	ns
CLKOUT fall time	<5>	t_{KF}			15	ns



(2) Output waveform (other than port 4, port 5, port 6, port 9, and CLKOUT)

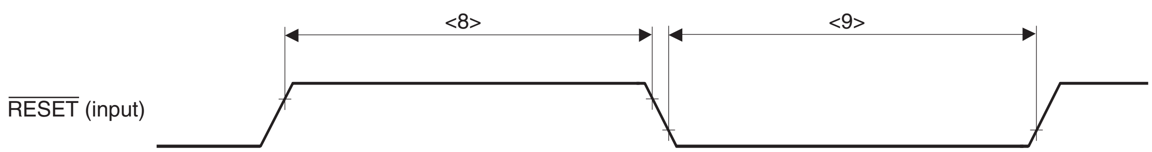
(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	<6>	t_{OR}			20	ns
Output fall time	<7>	t_{OF}			20	ns

(3) Reset timing

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ pin high-level width	<8>	t_{WRSH}		500		ns
$\overline{\text{RESET}}$ pin low-level width	<9>	t_{WRSL}		500		ns



(4) Bus timing

(a) Clock asynchronous (T_A = –40 to +85°C, V_{DD} = BV_{DD} = 4.0 to 5.5 V, EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	<10>	t_{SAST}		0.5T – 16		ns
Address hold time (from ASTB↓)	<11>	t_{HSTA}		0.5T – 15		ns
Address float delay time from $\overline{\text{DSTB}}$↓	<12>	t_{FDA}			0	ns
Data input setup time from address	<13>	t_{SAID}			(2 + n)T – 40	ns
Data input setup time from $\overline{\text{DSTB}}$↓	<14>	t_{SDID}			(1 + n)T – 40	ns
Delay time from ASTB↓ to $\overline{\text{DSTB}}$↓	<15>	t_{DSTD}		0.5T – 15		ns
Data input hold time (from $\overline{\text{DSTB}}$↑)	<16>	t_{HDID}		0		ns
Address output time from $\overline{\text{DSTB}}$↑	<17>	t_{DDA}		(1 + i)T – 15		ns
Delay time from $\overline{\text{DSTB}}$↑ to ASTB↑	<18>	t_{DDST1}		0.5T – 15		ns
Delay time from $\overline{\text{DSTB}}$↑ to ASTB↓	<19>	t_{DDST2}		(1.5 + i)T – 15		ns
$\overline{\text{DSTB}}$ low-level width	<20>	t_{WDL}		(1 + n)T – 22		ns
ASTB high-level width	<21>	t_{WSTH}		T – 15		ns
Data output time from $\overline{\text{DSTB}}$↓	<22>	t_{DDOD}			10	ns
Data output setup time (to $\overline{\text{DSTB}}$↑)	<23>	t_{SODD}		(1 + n)T – 25		ns
Data output hold time (from $\overline{\text{DSTB}}$↑)	<24>	t_{HDOD}		T – 20		ns
$\overline{\text{WAIT}}$ setup time (to address)	<25>	t_{SAWT1}	n ≥ 1		1.5T – 40	ns
	<26>	t_{SAWT2}	n ≥ 1		(1.5 + n)T – 40	ns
$\overline{\text{WAIT}}$ hold time (from address)	<27>	t_{HAWT1}	n ≥ 1	(0.5 + n)T		ns
	<28>	t_{HAWT2}	n ≥ 1	(1.5 + n)T		ns
$\overline{\text{WAIT}}$ setup time (to ASTB↓)	<29>	t_{SSTWT1}	n ≥ 1		T – 32	ns
	<30>	t_{SSTWT2}	n ≥ 1		(1 + n)T – 32	ns
$\overline{\text{WAIT}}$ hold time (from ASTB↓)	<31>	t_{HSTWT1}	n ≥ 1	nT		ns
	<32>	t_{HSTWT2}	n ≥ 1	(1 + n)T		ns
$\overline{\text{HLDRQ}}$ high-level width	<33>	t_{WHQH}		T + 10		ns
$\overline{\text{HLDAK}}$ low-level width	<34>	t_{WHAL}		T – 15		ns
Bus output delay time from $\overline{\text{HLDAK}}$↑	<35>	t_{DHAC}		–6		ns
Delay time from $\overline{\text{HLDRQ}}$↓ to $\overline{\text{HLDAK}}$↓	<36>	t_{DHQHA1}			(2n + 7.5)T + 25	ns
Delay time from $\overline{\text{HLDRQ}}$↑ to $\overline{\text{HLDAK}}$↑	<37>	t_{DHQHA2}		0.5T	1.5T + 25	ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operating 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 a read cycle (0 or 1).

4. The values in the above specifications are values for when clocks with a 5:5 duty ratio are input from X1.

(b) Clock asynchronous (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = 3.0 to 4.0 V, EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	<10>	t_{SAST}		0.5T – 20		ns
Address hold time (from ASTB↓)	<11>	t_{HSTA}	**Note 1**	0.5T – 20		ns
			Note 2	0.5T – 22		ns
Address float delay time from $\overline{\text{DSTB}}$↓	<12>	t_{FDA}			0	ns
Data input setup time from address	<13>	t_{SAID}			(2 + n)T – 50	ns
Data input setup time from $\overline{\text{DSTB}}$↓	<14>	t_{SDID}			(1 + n)T – 50	ns
Delay time from ASTB↓ to $\overline{\text{DSTB}}$↓	<15>	t_{DSTD}		0.5T – 15		ns
Data input hold time (from $\overline{\text{DSTB}}$↑)	<16>	t_{HDID}		0		ns
Address output time from $\overline{\text{DSTB}}$↑	<17>	t_{DDA}		(1 + i)T – 15		ns
Delay time from $\overline{\text{DSTB}}$↑ to ASTB↑	<18>	t_{DDST1}		0.5T – 15		ns
Delay time from $\overline{\text{DSTB}}$↑ to ASTB↓	<19>	t_{DDST2}		(1.5 + i)T – 15		ns
$\overline{\text{DSTB}}$ low-level width	<20>	t_{WDL}		(1 + n)T – 35		ns
ASTB high-level width	<21>	t_{WSTH}		T – 15		ns
Data output time from $\overline{\text{DSTB}}$↓	<22>	t_{DDOD}			10	ns
Data output setup time (to $\overline{\text{DSTB}}$↑)	<23>	t_{SODD}		(1 + n)T – 35		ns
Data output hold time (from $\overline{\text{DSTB}}$↑)	<24>	t_{HDOD}		T – 25		ns
$\overline{\text{WAIT}}$ setup time (to address)	<25>	t_{SAWT1}	n ≥ 1		1.5T – 55	ns
	<26>	t_{SAWT2}	n ≥ 1		(1.5 + n)T – 55	ns
$\overline{\text{WAIT}}$ hold time (from address)	<27>	t_{HAWT1}	n ≥ 1	(0.5 + n)T		ns
	<28>	t_{HAWT2}	n ≥ 1	(1.5 + n)T		ns
$\overline{\text{WAIT}}$ setup time (to ASTB↓)	<29>	t_{SSTWT1}	n ≥ 1		T – 45	ns
	<30>	t_{SSTWT2}	n ≥ 1		(1 + n)T – 45	ns
$\overline{\text{WAIT}}$ hold time (from ASTB↓)	<31>	t_{HSTWT1}	n ≥ 1	nT		ns
	<32>	t_{HSTWT2}	n ≥ 1	(1 + n)T		ns
$\overline{\text{HLDRQ}}$ high-level width	<33>	t_{WHQH}		T + 10		ns
$\overline{\text{HLDAK}}$ low-level width	<34>	t_{WHAL}		T – 25		ns
Bus output delay time from $\overline{\text{HLDAK}}$↑	<35>	t_{DHAC}		–6		ns
Delay time from $\overline{\text{HLDRQ}}$↓ to $\overline{\text{HLDAK}}$↓	<36>	t_{DHQHA1}			(2n + 7.5)T + 25	ns
Delay time from $\overline{\text{HLDRQ}}$↑ to $\overline{\text{HLDAK}}$↑	<37>	t_{DHQHA2}		0.5T	1.5T + 25	ns

Notes 1. μPD703031A, 703031AY, 703031B, 703031BY, 703033A, 703033AY, 703033B, 703033BY, 703034A, 703034AY, 703034B, 703034BY, 703035A, 703035AY, 703035B, 703035BY, 70F3033A, 70F3033AY, 70F3033B, 70F3033BY, 70F3035A, 70F3035AY, 70F3035B, 70F3035BY

2. μPD703030B, 703030BY, 703032A, 703032AY, 703032B, 703032BY, 703036H, 703036HY, 703037A, 703037AY, 703037H, 703037HY, 70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY, 70F3036H, 70F3036HY, 70F3037A, 70F3037AY, 70F3037H, 70F3037HY

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operating 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 a read cycle (0 or 1).

4. The values in the above specifications are values for when clocks with a 5:5 duty ratio are input from X1.

(c) Clock synchronous (T_A = –40 to +85°C, V_{DD} = BV_{DD} = 4.0 to 5.5 V, EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	<38>	t_{DKA}		0	19	ns
Delay time from CLKOUT↑ to address float	<39>	t_{FKA}		–12	10	ns
Delay time from CLKOUT↓ to ASTB	<40>	t_{DKST}		0	19	ns
Delay time from CLKOUT↑ to $\overline{\text{DSTB}}$	<41>	t_{DKD}		0	19	ns
Data input setup time (to CLKOUT↑)	<42>	t_{SIDK}		20		ns
Data input hold time (from CLKOUT↑)	<43>	t_{HKID}		5		ns
Data output delay time from CLKOUT↑	<44>	t_{DKOD}			19	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	<45>	t_{SWTK}		20		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	<46>	t_{HKWT}		5		ns
$\overline{\text{HLDRQ}}$ setup time (to CLKOUT↓)	<47>	t_{SHQK}		20		ns
$\overline{\text{HLDRQ}}$ hold time (from CLKOUT↓)	<48>	t_{HKHQ}		5		ns
Delay time from CLKOUT↑ to address float (during bus hold)	<49>	t_{DKF}			19	ns
Delay time from CLKOUT↑ to $\overline{\text{HLDAK}}$	<50>	t_{DKHA}			19	ns

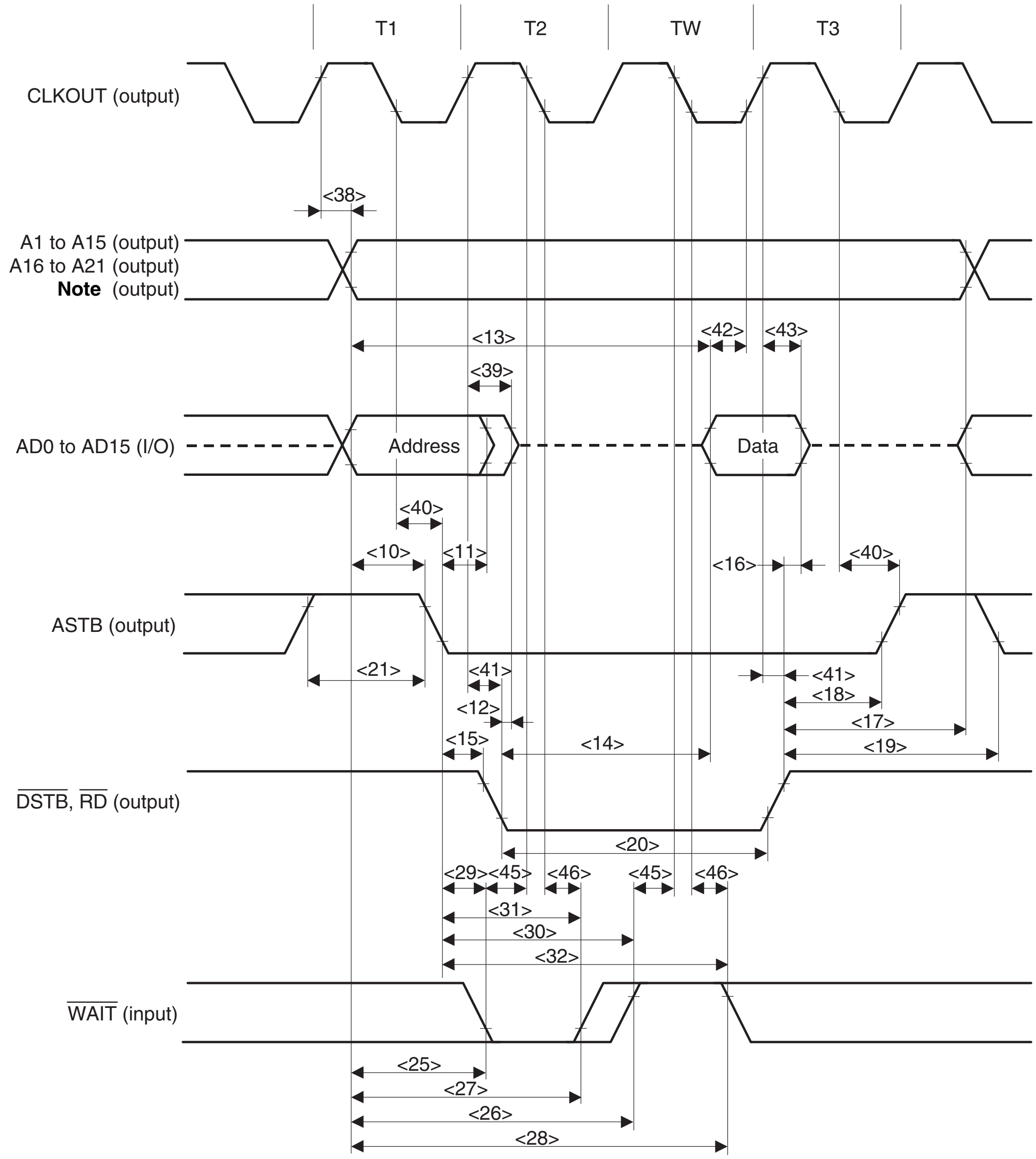
Remark The values in the above specifications are values for when clocks with a 5:5 duty ratio are input from X1.

(d) Clock synchronous (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = 3.0 to 4.0 V, EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	<38>	t_{DKA}		0	22	ns
Delay time from CLKOUT↑ to address float	<39>	t_{FKA}		–16	10	ns
Delay time from CLKOUT↓ to ASTB	<40>	t_{DKST}		0	19	ns
Delay time from CLKOUT↑ to $\overline{\text{DSTB}}$	<41>	t_{DKD}		0	22	ns
Data input setup time (to CLKOUT↑)	<42>	t_{SIDK}		20		ns
Data input hold time (from CLKOUT↑)	<43>	t_{HKID}		5		ns
Data output delay time from CLKOUT↑	<44>	t_{DKOD}			22	ns
$\overline{\text{WAIT}}$ setup time (to CLKOUT↓)	<45>	t_{SWTK}		24		ns
$\overline{\text{WAIT}}$ hold time (from CLKOUT↓)	<46>	t_{HKWT}		5		ns
$\overline{\text{HLDRQ}}$ setup time (to CLKOUT↓)	<47>	t_{SHQK}		24		ns
$\overline{\text{HLDRQ}}$ hold time (from CLKOUT↓)	<48>	t_{HKHQ}		5		ns
Delay time from CLKOUT↑ to address float (during bus hold)	<49>	t_{DKF}			19	ns
Delay time from CLKOUT↑ to $\overline{\text{HLDAK}}$	<50>	t_{DKHA}			19	ns

Remark The values in the above specifications are values for when clocks with a 5:5 duty ratio are input from X1.

(e) Read cycle (CLKOUT synchronous/asynchronous, 1 wait)

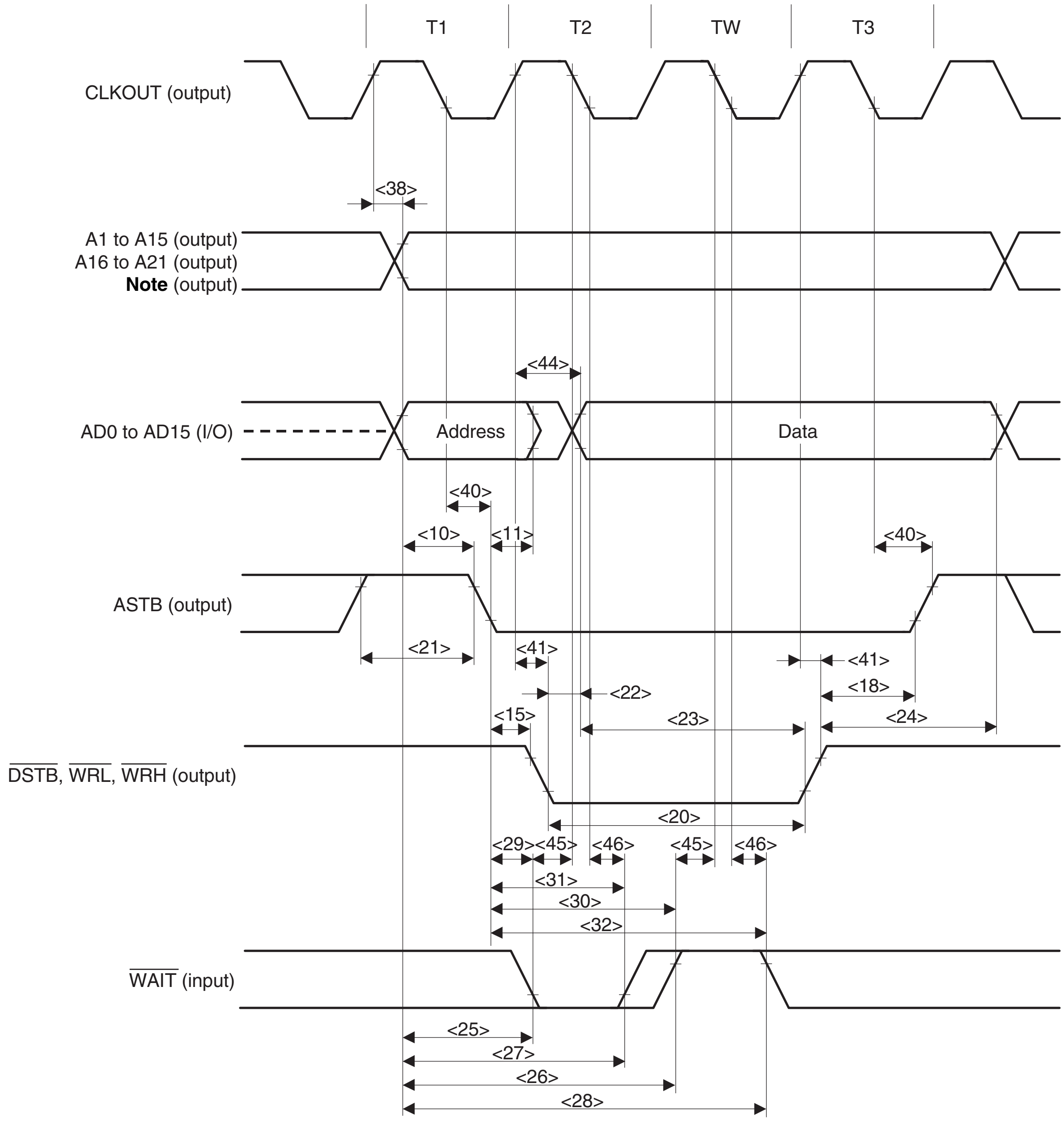


Note R/$\overline{\text{W}}$, $\overline{\text{UBEN}}$, $\overline{\text{LBEN}}$

Remarks 1. The broken lines indicate high impedance.

★ **2.** $\overline{\text{WRL}}$ and $\overline{\text{WRH}}$ are high level.

(f) Write cycle (CLKOUT synchronous/asynchronous, 1 wait)

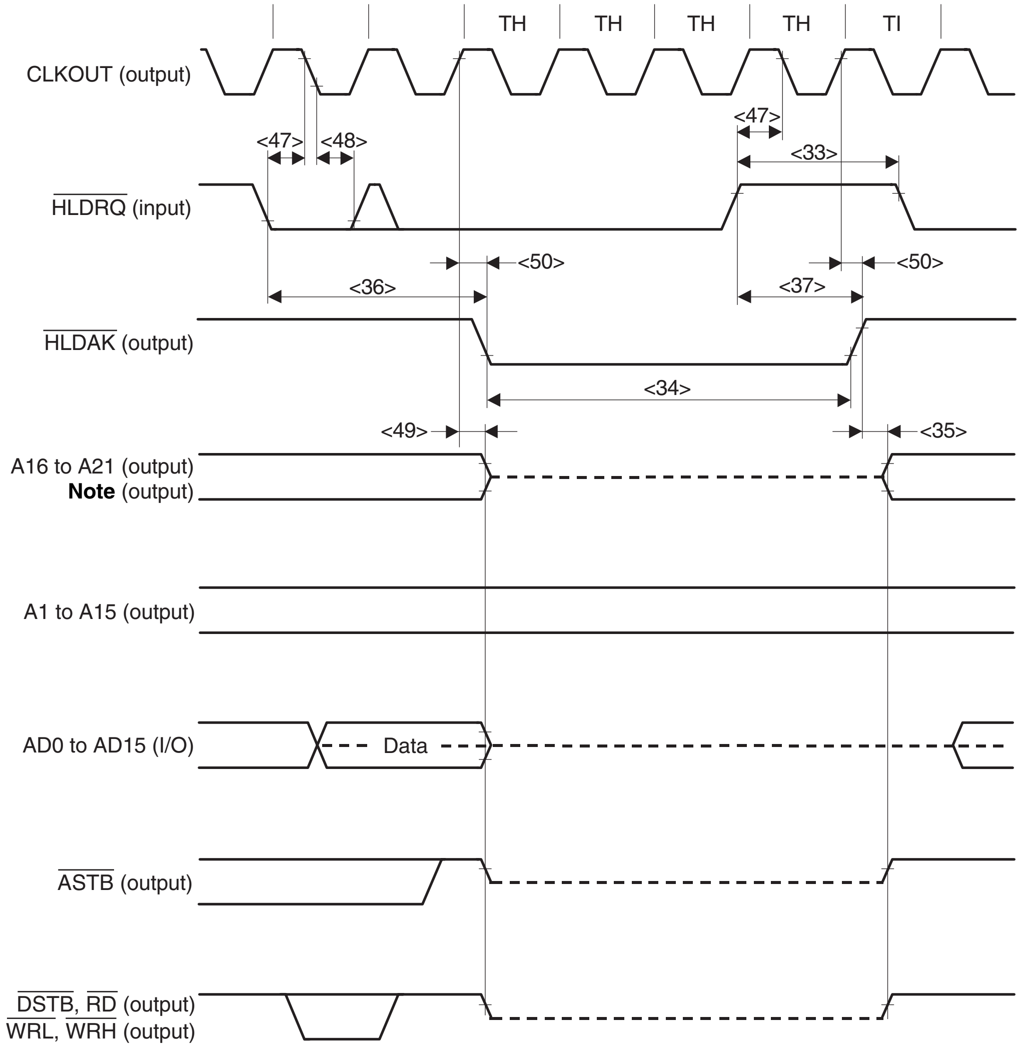


Note R/$\overline{\text{W}}$, $\overline{\text{UBEN}}$, $\overline{\text{LBEN}}$

Remarks 1. The broken lines indicate high impedance.

2. $\overline{\text{RD}}$ is high level.

(g) Bus hold timing



Note R/$\overline{W}$, $\overline{UBEN}$, $\overline{LBEN}$

Remark The broken lines indicate high impedance.

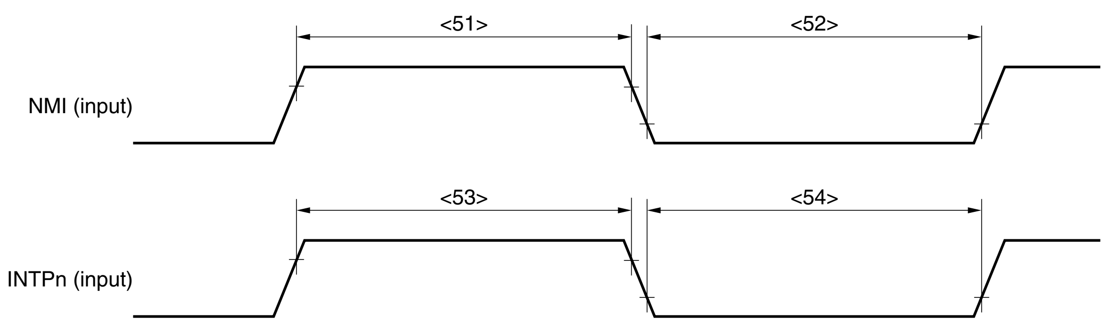
(5) Interrupt timing

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	<51>	t_{WNIH}		500		ns
NMI low-level width	<52>	t_{WNIL}		500		ns
INTPn high-level width	<53>	t_{WITH}	n = 0 to 3, analog noise elimination	500		ns
			n = 4, 5, digital noise elimination	3T + 20		ns
			n = 6, digital noise elimination	3Tsmp + 20		ns
INTPn low-level width	<54>	t_{WITL}	n = 0 to 3, analog noise elimination	500		ns
			n = 4, 5, digital noise elimination	3T + 20		ns
			n = 6, digital noise elimination	3Tsmp + 20		ns

Remarks 1. T = 1/fxx

2. Tsmp = Noise elimination sampling clock cycle



Remark n = 0 to 6

(6) RPU timing

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn0, TIn1 high-level width	<55>	t_{TIHn}	n = 0, 1	$2T_{sam}$ + 20 Note		ns
TIn0, TIn1 low-level width	<56>	t_{TILn}	n = 0, 1	$2T_{sam}$ + 20 Note		ns
TIm high-level width	<57>	t_{TIHm}	m = 2 to 5	3T + 20		ns
TIm low-level width	<58>	t_{TILm}	m = 2 to 5	3T + 20		ns

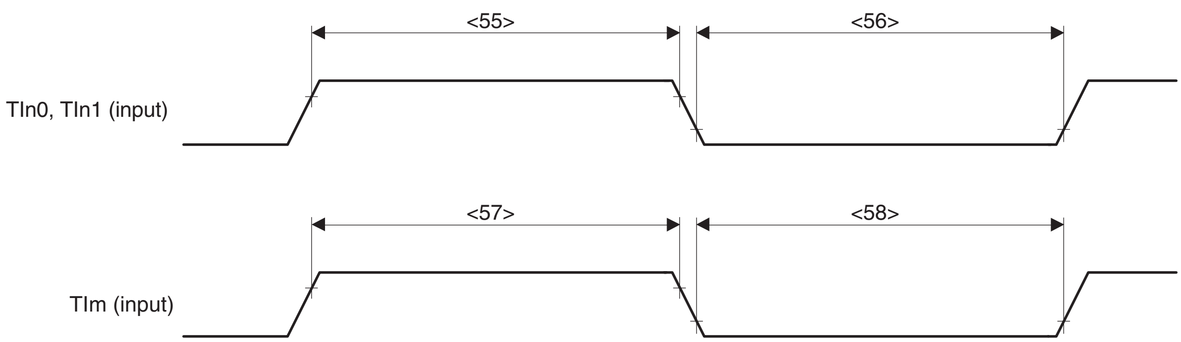
Note T_{sam} can select the following count clocks by setting the PRMn2 to PRMn0 bits of prescaler mode registers n0, n1 (PRMn0, PRMn1).

When n = 0 (TM0), T_{sam} = 2T, 4T, 16T, 64T, 256T, or 1/INTWTNI cycle

When n = 1 (TM1), T_{sam} = 2T, 4T, 16T, 32T, 128T, or 256T

However, when the TIn0 valid edge is selected as the count clock, T_{sam} = 4T.

Remark T = 1/fxx



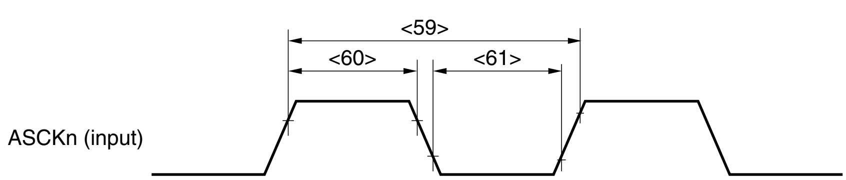
Remark n = 0, 1
m = 2 to 5

(7) Asynchronous serial interface (UART0, UART1) timing

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
ASCKn cycle time	<59>	t_{KCY13}		200		ns
ASCKn high-level width	<60>	t_{KH13}		80		ns
ASCKn low-level width	<61>	t_{KL13}		80		ns

Remark n = 0, 1



Remark n = 0, 1

(8) 3-wire serial interface (CSI0 to CSI3) timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle	<62>	t_{KCY1}		400		ns
$\overline{SCKn}$ high-level width	<63>	t_{KH1}		140		ns
$\overline{SCKn}$ low-level width	<64>	t_{KL1}		140		ns
SIn setup time (to $\overline{SCKn}$↑)	<65>	t_{SIK1}		50		ns
SIn hold time (from $\overline{SCKn}$↑)	<66>	t_{KSI1}		50		ns
Delay time from $\overline{SCKn}$↓ to SOn output	<67>	t_{KSO1}			60	ns

Remark n = 0 to 3

(b) Slave mode

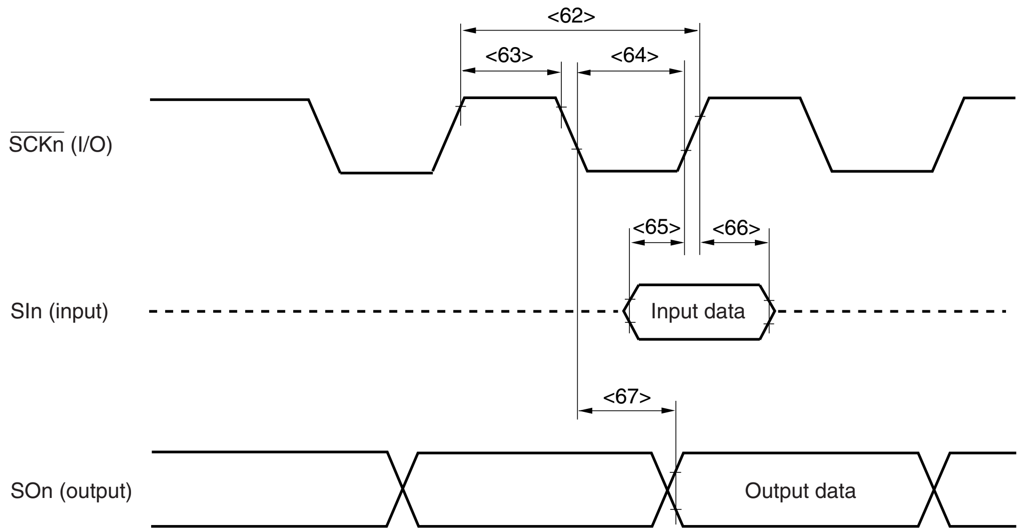
(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = V_{SS} = 0 V)

Parameter	Symbol		Conditions		MIN.	MAX.	Unit
$\overline{SCKn}$ cycle	<62>	t_{KCY2}			400		ns
$\overline{SCKn}$ high-level width	<63>	t_{KH2}			140		ns
$\overline{SCKn}$ low-level width	<64>	t_{KL2}			140		ns
SIn setup time (to $\overline{SCKn}$↑)	<65>	t_{SIK2}			50		ns
SIn hold time (from $\overline{SCKn}$↑)	<66>	t_{KSI2}			50		ns
Delay time from $\overline{SCKn}$↓ to SOn output	<67>	t_{KSO2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	**Note 1**		60	ns
				Note 2		70	ns
			3.0 V ≤ EV_{DD} < 4.0 V			100	ns

Notes 1. μPD703031A, 703031AY, 703031B, 703031BY, 703033A, 703033AY, 703033B, 703033BY, 703034A, 703034AY, 703034B, 703034BY, 703035A, 703035AY, 703035B, 703035BY, 70F3033A, 70F3033AY, 70F3033B, 70F3033BY, 70F3035A, 70F3035AY, 70F3035B, 70F3035BY

2. μPD703030B, 703030BY, 703032A, 703032AY, 703032B, 703032BY, 703036H, 703036HY, 703037A, 703037AY, 703037H, 703037HY, 70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY, 70F3036H, 70F3036HY, 70F3037A, 70F3037AY, 70F3037H, 70F3037HY

Remark n = 0 to 3



Remarks 1. The broken lines indicate high impedance.

2. n = 0 to 3

(9) 3-wire variable length serial interface (CSI4) timing

(a) Master mode

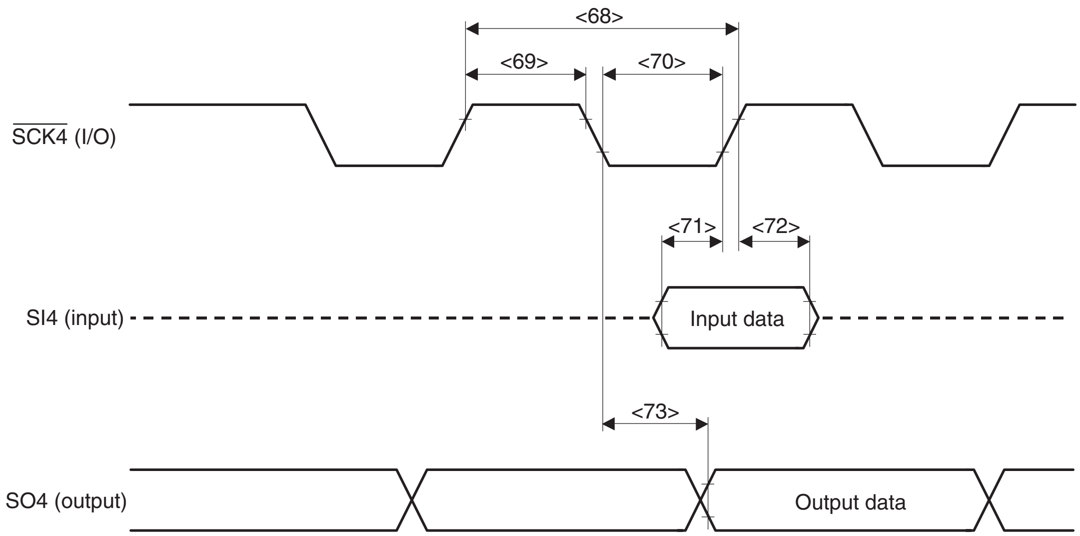
(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCK4}$ cycle	<68>	t_{KCY1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	200		ns
			3.0 V ≤ EV_{DD} < 4.0 V	400		ns
$\overline{SCK4}$ high-level width	<69>	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	60		ns
			3.0 V ≤ EV_{DD} < 4.0 V	140		ns
$\overline{SCK4}$ low-level width	<70>	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	60		ns
			3.0 V ≤ EV_{DD} < 4.0 V	140		ns
SI4 setup time (to $\overline{SCK4}$↑)	<71>	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	25		ns
			3.0 V ≤ EV_{DD} < 4.0 V	50		ns
SI4 hold time (from $\overline{SCK4}$↑)	<72>	t_{KSI1}		20		ns
Delay time from $\overline{SCK4}$↓ to SO4 output	<73>	t_{KSO1}			55	ns

(b) Slave mode

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCK4}$ cycle	<68>	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	200		ns
			3.0 V ≤ EV_{DD} < 4.0 V	400		ns
$\overline{SCK4}$ high-level width	<69>	t_{KH2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	60		ns
			3.0 V ≤ EV_{DD} < 4.0 V	140		ns
$\overline{SCK4}$ low-level width	<70>	t_{KL2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	60		ns
			3.0 V ≤ EV_{DD} < 4.0 V	140		ns
SI4 setup time (to $\overline{SCK4}$↑)	<71>	t_{SIK2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	25		ns
			3.0 V ≤ EV_{DD} < 4.0 V	50		ns
SI4 hold time (from $\overline{SCK4}$↑)	<72>	t_{KSI2}		20		ns
Delay time from $\overline{SCK4}$↓ to SO4 output	<73>	t_{KSO2}	4.0 V ≤ EV_{DD} ≤ 5.5 V		55	ns
			3.0 V ≤ EV_{DD} < 4.0 V		100	ns



Remark The broken lines indicate high impedance.

(10) I²C bus mode (Y versions only) (1/2)

(T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter		Symbol		Normal Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCLn clock frequency		–	f_{CLK}	0	100	0	400	kHz
Bus-free time (between stop/start conditions)		<74>	t_{BUF}	4.7	–	1.3	–	μs
Hold time Note 1		<75>	$t_{HD:STA}$	4.0	–	0.6	–	μs
SCLn clock low-level width		<76>	t_{LOW}	4.7	–	1.3	–	μs
SCLn clock high-level width		<77>	t_{HIGH}	4.0	–	0.6	–	μs
Setup time for start/restart conditions		<78>	$t_{SU:STA}$	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	<79>	$t_{HD:DAT}$	5.0	–	–	–	μs
	I²C mode			0 Note 2	–	0 Note 2	0.9 Note 3	μs
Data setup time		<80>	$t_{SU:DAT}$	250	–	100 Note 4	–	ns
SDAn and SCLn signal rise time		<81>	t_R	–	1000	20 + 0.1Cb Note 5	300	ns
SDAn and SCLn signal fall time		<82>	t_F	–	300	20 + 0.1Cb Note 5	300	ns
Stop condition setup time		<83>	$t_{SU:STO}$	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		<84>	t_{SP}	–	–	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 SDAn signal (at $V_{IHmin.}$ of SCLn signal) in order to occupy the undefined area at the falling edge of SCLn.

3. If the system does not extend the SCLn signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

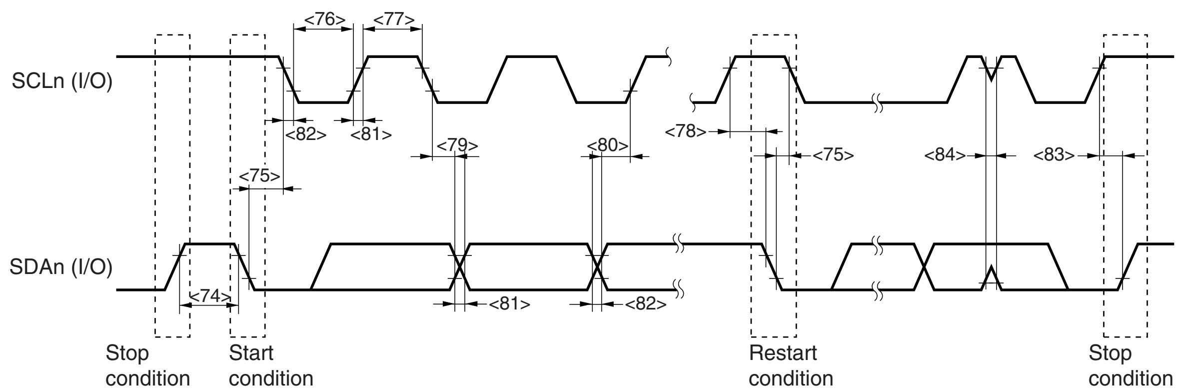
4. The high-speed mode I²C bus can be used in the normal-mode I²C bus system. In this case, set the high-speed mode I²C bus so that it meets the following conditions.

- If the system does not extend the SCLn signal's low state hold time:
  $t_{SU:DAT} \geq 250$ ns
- If the system extends the SCLn signal's low state hold time:
  Transmit the following data bit to the SDAn line prior to the SCLn line release ($t_{Rmax.}$ + $t_{SU:DAT}$ = 1000 + 250 = 1250 ns: Normal mode I²C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

Remark n = 0, 1

(10) I²C bus mode (Y versions only) (2/2)



Remark n = 0, 1

A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = AV_{DD} = AV_{REF} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = 0 V, Output pin load capacitance: C_L = 50 pF)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	–			10	10	10	bit
Overall error[Note 1]	–	ADM2 = 00H				±0.6	%FSR
		ADM2 = 01H				±1.0	%FSR
Conversion time	t_{CONV}			5		10	μs
Zero-scale error[Note 1]						±0.4	%FSR
Full-scale error[Note 1]		ADM2 = 00H				±0.4	%FSR
		ADM2 = 01H				±0.6	%FSR
Integral linearity error[Note 2]		ADM2 = 00H				±4.0	LSB
		ADM2 = 01H				±6.0	LSB
Differential linearity error[Note 2]		ADM2 = 00H				±4.0	LSB
		ADM2 = 01H				±6.0	LSB
Analog reference voltage	AV_{REF}	AV_{REF} = AV_{DD}		4.5		5.5	V
Analog power supply voltage	AV_{DD}			4.5		5.5	V
Analog input voltage	V_{IAN}			AV_{SS}		AV_{REF}	V
AV_{REF} input current	AI_{REF}				1	2	mA
AV_{DD} power supply current	AI_{DD}	Operating current	ADM2 = 00H		3	6	mA
			ADM2 = 01H		4	8	mA

Notes 1. Excluding quantization error (±0.05 %FSR)

2. Excluding quantization error (±0.5 LSB)

Remarks 1. LSB: Least Significant Bit
FSR: Full Scale Range

2. ADM2: A/D converter mode register 2

IEBus Controller Characteristics (V850/SB2 only) (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, BV_{DD} = EV_{DD} = 3.0 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
IEBus system clock frequency	f_S	Communication mode: fixed to mode 1		6.0 Note 1		MHz
				6.29 Notes 1, 2		MHz

Notes 1. 6.0 MHz and 6.29 MHz can not be used together for the IEBus system clock frequency

2. Although the system clock specified in the IEBus specification is 6.0 MHz, operation is guaranteed at 6.29 MHz system clock in the V850/SB2.

Regulator (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output stabilization time	t_{REG}	Stabilization capacitance C = 1 μF (Connected to REGC pin)	1			ms

Cautions 1. Be sure to start inputting supply voltage V_{DD} when $\overline{RESET}$ = V_{SS} = EV_{SS} = BV_{SS} = 0 V (the above state), and make $\overline{RESET}$ high level after the t_{REG} period has elapsed.

2. If supply voltage BV_{DD} or EV_{DD} is input before the t_{REG} period has elapsed following the input of supply voltage V_{DD}, note that data may be driven from the pins until the t_{REG} period has elapsed because the I/O buffers' power supply was turned on while the circuit was in an undefined state.

Flash Memory Programming Mode

(1) μPD70F3030B, 70F3030BY, 70F3032A, 70F3032AY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY

Write/erase characteristics (T_A = –20 to 85°C, V_{DD} = AV_{DD} = BV_{DD} = EV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} power supply voltage	V_{PP2}	During flash memory programming	7.5	7.8	8.1	V
V_{DD} power supply current	I_{DD}	When V_{PP} = V_{PP2}, f_{XX} = 20 MHz			**Note 1**	mA
V_{PP} power supply current	I_{PP}	V_{PP} = V_{PP2}			100	mA
Step erase time	t_{ER}	**Note 2**		0.2		s
Overall erase time per area	t_{ERA}	When the step erase time = 0.2 s, **Note 3**			20	s/area
Writeback time	t_{WB}	**Note 4**		1		ms
Number of writebacks per writeback command	C_{WB}	When the writeback time = 1 ms, **Note 5**			300	Count/ writeback command
Number of erase/writebacks	C_{ERWB}				16	Count
Step writing time	t_{WR}	**Note 6**		20		μs
Overall writing time per word	t_{WRW}	When the step writing time = 20 μs (1 word = 4 bytes), **Note 7**	20		200	μs/word
Number of rewrites per area	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 8**		100		Count/area

Notes 1. I_{DD} = 63 (μPD70F3033B, 70F3033BY), I_{DD} = 68 (μPD70F3030B, 70F3030BY), I_{DD} = 73 (μPD70F3032A, 70F3032AY, 70F3032B, 70F3032BY)

2. The recommended setting value of the step erase time is 0.2 s.

3. The prewrite time prior to erasure and the erase verify time (writeback time) are not included.

4. The recommended setting value of the writeback time is 1 ms.

5. Writeback is executed once by the issuance of the writeback 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, both “erase to write” and “write only” are counted as one rewrite.

Example (P: Write, E: Erase)

Shipped product ———→ P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

Remarks 1. When the PG-FP3 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

2. Area 0 = 000000H to 01FFFFH Area 2 = 040000H to 05FFFFH
Area 1 = 020000H to 03FFFFH Area 3 = 060000H to 07FFFFH
μPD70F3030B, 70F3030BY: Areas 0 to 2
μPD70F3032A, 70F3032AY, 70F3032B, 70F3032BY: Areas 0 to 3
μPD70F3033B, 70F3033BY: Areas 0 and 1

(2) μPD70F3033A, 70F3033AY

Write/erase characteristics (T_A = 10 to 85°C … K rank product,
T_A = –20 to +85°C … E rank product,
V_{DD} = AV_{DD} = BV_{DD} = EV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} power supply voltage	V_{PP2}	During flash memory programming	7.5	7.8	8.1	V
V_{DD} power supply current	I_{DD}	When V_{PP} = V_{PP2}, f_{XX} = 20 MHz			63	mA
V_{PP} power supply current	I_{PP}	V_{PP} = V_{PP2}			100	mA
Step erase time	t_{ER}	**Note 1**		0.2		s
Overall erase time per area	t_{ERA}	When the step erase time = 0.2 s, **Note 2**			20	s/area
Writeback time	t_{WB}	**Note 3**		1		ms
Number of writebacks per writeback command	C_{WB}	When the writeback time = 1 ms, **Note 4**			300	Count/ writeback command
Number of erase/writebacks	C_{ERWB}				16	Count
Step writing time	t_{WR}	**Note 5**		20		μs
Overall writing time per word	t_{WRW}	When the step writing time = 20 μs (1 word = 4 bytes), **Note 6**	20		200	μs/word
Number of rewrites per area	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 7**	**Note 8**			Count/area

Notes 1. The recommended setting value of the step erase time is 0.2 s.

2. The prewrite time prior to erasure and the erase verify time (writeback time) are not included.

3. The recommended setting value of the writeback time is 1 ms.

4. Writeback is executed once by the issuance of the writeback command. Therefore, the retry count must be the maximum value minus the number of commands issued.

5. The recommended setting value of the step writing time is 20 μs.

6. 20 μs is added to the actual writing time per word. The internal verify time during and after the writing is not included.

7. When writing initially to shipped products, both "erase to write" and "write only" are counted as one rewrite for.

Example (P: Write, E: Erase)
Shipped product ———→ P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

8. K rank product: 20 writes/area
E rank product: 100 writes/area

Remarks 1. When the PG-FP3 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

2. Area 0 = 000000H to 01FFFFH
Area 1 = 020000H to 03FFFFH

(3) μPD70F3035A, 70F3035AY

**Write/erase characteristics (T_A = 10 to 85°C … K, E rank product,
T_A = –20 to +85°C … P rank product,
$V_{DD} = AV_{DD} = BV_{DD} = EV_{DD}$ = 4.5 to 5.5 V, $V_{SS} = AV_{SS} = BV_{SS} = EV_{SS}$ = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} power supply voltage	V_{PP2}	During flash memory programming	7.5	7.8	8.1	V
V_{DD} power supply current	I_{DD}	When $V_{PP} = V_{PP2}$, f_{XX} = 13 MHz			51	mA
V_{PP} power supply current	I_{PP}	$V_{PP} = V_{PP2}$			100	mA
Step erase time	t_{ER}	**Note 1**		0.2		s
Overall erase time per area	t_{ERA}	When the step erase time = 0.2 s, **Note 2**			20	s/area
Writeback time	t_{WB}	**Note 3**		1		ms
Number of writebacks per writeback command	C_{WB}	When the writeback time = 1 ms, **Note 4**			300	Count/ writeback command
Number of erase/writebacks	C_{ERWB}				16	Count
Step writing time	t_{WR}	**Note 5**		20		μs
Overall writing time per word	t_{WRW}	When the step writing time = 20 μs (1 word = 4 bytes), **Note 6**	20		200	μs/word
Number of rewrites per area	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 7**	**Note 8**			Count/area

Notes 1. The recommended setting value of the step erase time is 0.2 s.

2. The prewrite time prior to erasure and the erase verify time (writeback time) are not included.

3. The recommended setting value of the writeback time is 1 ms.

4. Writeback is executed once by the issuance of the writeback command. Therefore, the retry count must be the maximum value minus the number of commands issued.

5. The recommended setting value of the step writing time is 20 μs.

6. 20 μs is added to the actual writing time per word. The internal verify time during and after the writing is not included.

7. When writing initially to shipped products, both “erase to write” and “write only” are counted as one rewrite for.

Example (P: Write, E: Erase)

Shipped product ——→ P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

8. K, E rank product: 20 writes/area
P rank product: 100 writes/area

Remarks 1. When the PG-FP3 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

2. Area 0 = 000000H to 01FFFFH
Area 1 = 020000H to 03FFFFH

(4) μPD70F3035B, 70F3035BY, 70F3037A, 70F3037AY,

Write/erase characteristics (T_A = –20 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = EV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} power supply voltage	V_{PP2}	During flash memory programming	7.5	7.8	8.1	V
V_{DD} power supply current	I_{DD}	When V_{PP} = V_{PP2}, f_{XX} = 13 MHz			**Note 1**	mA
V_{PP} power supply current	I_{PP}	V_{PP} = V_{PP2}			100	mA
Step erase time	t_{ER}	**Note 2**		0.2		s
Overall erase time per area	t_{ERA}	When the step erase time = 0.2 s, **Note 3**			20	s/area
Writeback time	t_{WB}	**Note 4**		1		ms
Number of writebacks per writeback command	C_{WB}	When the writeback time = 1 ms, **Note 5**			300	Count/ writeback command
Number of erase/writebacks	C_{ERWB}				16	Count
Step writing time	t_{WR}	**Note 6**		20		μs
Overall writing time per word	t_{WRW}	When the step writing time = 20 μs (1 word = 4 bytes), **Note 7**	20		200	μs/word
Number of rewrites per area	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 8**	100			Count/area

Notes 1. I_{DD} = 51 (μPD70F3035B, 70F3035BY),
I_{DD} = 61 (μPD70F3037A, 70F3037AY)

2. The recommended setting value of the step erase time is 0.2 s.

3. The prewrite time prior to erasure and the erase verify time (writeback time) are not included.

4. The recommended setting value of the writeback time is 1 ms.

5. Writeback is executed once by the issuance of the writeback 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, both “erase to write” and “write only” are counted as one rewrite.

Example (P: Write, E: Erase)

Shipped product ———→ P → E → P → E → P: 3 rewrites

Shipped product → E → P → E → P → E → P: 3 rewrites

Remarks 1. When the PG-FP3 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

2. Area 0 = 000000H to 01FFFFH Area 2 = 040000H to 05FFFFH
Area 1 = 020000H to 03FFFFH Area 3 = 060000H to 07FFFFH
μPD70F3035B, 70F3035BY: Areas 0 and 1
μPD70F3037A, 70F3037AY: Areas 0 to 3

(5) μPD70F3036H, 70F3036HY, 70F3037H, 70F3037HY

Write/erase characteristics (T_A = –20 to +85°C, V_{DD} = AV_{DD} = BV_{DD} = EV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = BV_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{PP} power supply voltage	V_{PP2}	During flash memory programming	7.5	7.8	8.1	V
V_{DD} power supply current	I_{DD}	When V_{PP} = V_{PP2}, f_{XX} = 19 MHz			**Note 1**	mA
V_{PP} power supply current	I_{PP}	V_{PP} = V_{PP2}			100	mA
Step erase time	t_{ER}	**Note 2**		0.2		s
Overall erase time per area	t_{ERA}	When the step erase time = 0.2 s, **Note 3**			20	s/area
Writeback time	t_{WB}	**Note 4**		1		ms
Number of writebacks per writeback command	C_{WB}	When the writeback time = 1 ms, **Note 5**			300	Count/ writeback command
Number of erase/writebacks	C_{ERWB}				16	Count
Step writing time	t_{WR}	**Note 6**		20		μs
Overall writing time per word	t_{WRW}	When the step writing time = 20 μs (1 word = 4 bytes), **Note 7**	20		200	μs/word
Number of rewrites per area	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 8**	100			Count/area

Notes 1. I_{DD} = 68 (μPD70F3036H, 70F3036HY),
I_{DD} = 73 (μPD70F3037H, 70F3037HY)

2. The recommended setting value of the step erase time is 0.2 s.

3. The prewrite time prior to erasure and the erase verify time (writeback time) are not included.

4. The recommended setting value of the writeback time is 1 ms.

5. Writeback is executed once by the issuance of the writeback 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, both “erase to write” and “write only” are counted as one rewrite.

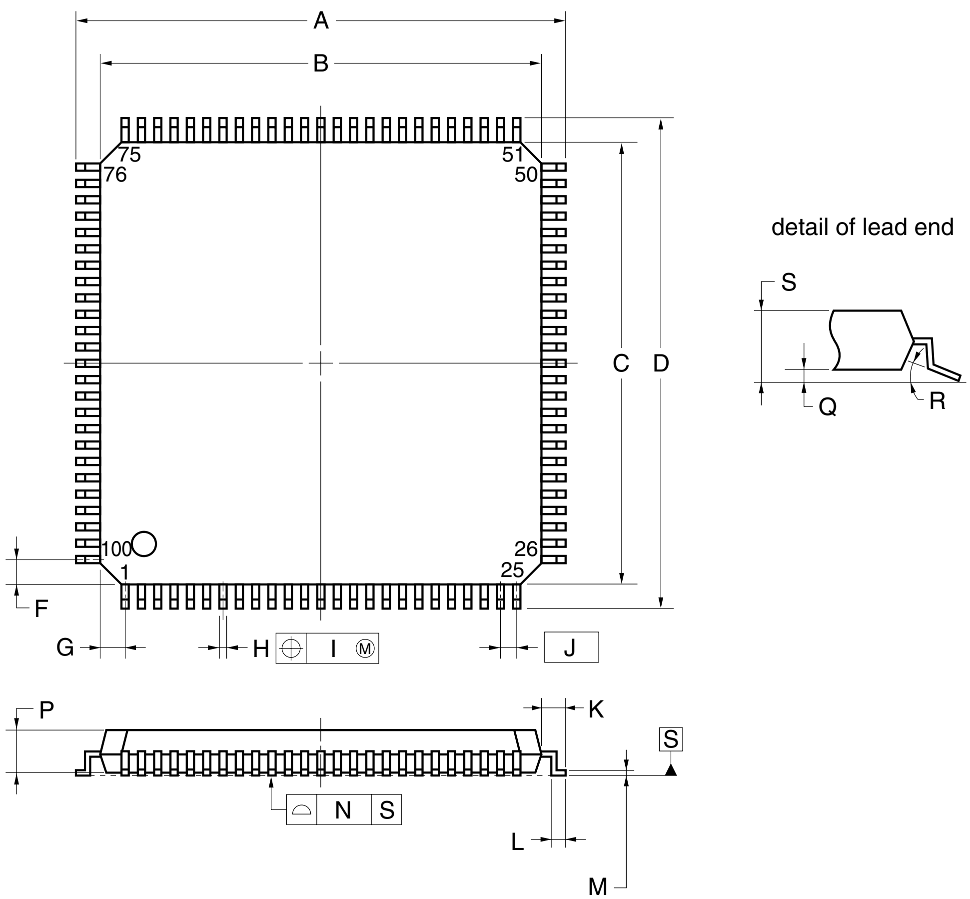
Example (P: Write, E: Erase)

Shipped product ——→ P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

Remarks 1. When the PG-FP3 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings unless otherwise specified.

2. Area 0 = 000000H to 01FFFFH Area 2 = 040000H to 05FFFFH
Area 1 = 020000H to 03FFFFH Area 3 = 060000H to 07FFFFH
μPD70F3036H, 70F3036HY: Areas 0 to 2
μPD70F3037H, 70F3037HY: Areas 0 to 3

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



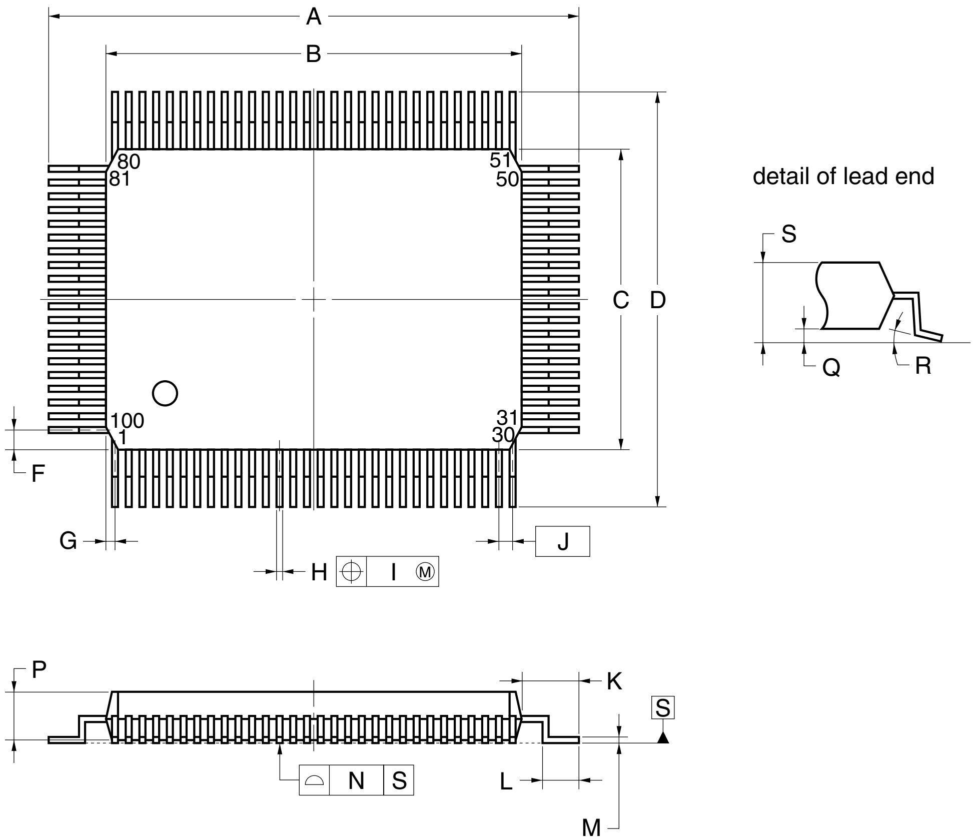
NOTE

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	$3°^{+7°}_{-3°}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

100-PIN PLASTIC QFP (14x20)



NOTE

Each lead centerline is located within 0.15 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	23.6±0.4
B	20.0±0.2
C	14.0±0.2
D	17.6±0.4
F	0.8
G	0.6
H	0.30±0.10
I	0.15
J	0.65 (T.P.)
K	1.8±0.2
L	0.8±0.2
M	$0.15^{+0.10}_{-0.05}$
N	0.10
P	2.7±0.1
Q	0.1±0.1
R	5°±5°
S	3.0 MAX.

P100GF-65-3BA1-4

★ # CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS

The V850/SB1 and V850/SB2 should be soldered and mounted under the following recommended conditions.

For the details of the recommended soldering conditions, refer to the document **Semiconductor Device Mounting Technology Manual (C10535E).**

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

Table 22-1. Surface Mounting Type Soldering Conditions (1/5)

(1) μPD703031AGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703031BGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703031AYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703031BYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703033AGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703033BGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703033AYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703033BYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703034AGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703034BGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703034AYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703034BYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703035AGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703035BGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703035AYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703035BYGC-xxx-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 to 72 hours)	IR35-107-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (at 200°C or higher), Count: Two times or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	VP15-107-2
Partial heating	Pin temperature: 300°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 22-1. Surface Mounting Type Soldering Conditions (2/5)

(2) μPD703030BGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703030BYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703036HGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD703036HYGC-xxx-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3033AGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3033BGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3033AYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3033BYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3035AGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3035BGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3035AYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3035BYGC-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 to 72 hours)	IR35-103-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (at 200°C or higher), Count: Two times or less Exposure limit: 3 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	VP15-103-2
Partial heating	Pin temperature: 300°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 22-1. Surface Mounting Type Soldering Conditions (3/5)

(3)

μPD703030BGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703035AGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703030BYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703035BGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703031AGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703035AYGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703031BGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703035BYGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703031AYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703036HGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703031BYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703036HYGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703032AGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703037AGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703032BGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703037HGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703032AYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703037AYGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703032BYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD703037HYGF-xxx-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703033AGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3033AGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703033BGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3033BGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703033AYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3033AYGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703033BYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3033BYGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703034AGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3035AGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703034BGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3035BGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703034AYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3035AYGF-3BA:**	**100-pin plastic QFP (14 × 20)**
μPD703034BYGF-xxx-3BA:	**100-pin plastic QFP (14 × 20)**	**μPD70F3035BYGF-3BA:**	**100-pin plastic QFP (14 × 20)**

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 20 to 72 hours)	IR35-207-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (at 200°C or higher), Count: Two times or less Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	VP15-207-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once Preheating temperature: 120°C max. (package surface temperature) Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	WS60-207-1
Partial heating	Pin temperature: 300°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 22-1. Surface Mounting Type Soldering Conditions (4/5)

(4) μPD70F3030BGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3030BYGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3032AGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3032BGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3032AYGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3032BYGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3036HGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3036HYGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3037AGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3037HGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3037AYGF-3BA: 100-pin plastic QFP (14 × 20)
μPD70F3037HYGF-3BA: 100-pin plastic QFP (14 × 20)

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 20 to 72 hours)	IR35-203-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (at 200°C or higher), Count: Two times or less Exposure limit: 3 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	VP15-203-2
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: once Preheating temperature: 120°C max. (package surface temperature) Exposure limit: 3 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	WS60-203-1
Partial heating	Pin temperature: 300°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 22-1. Surface Mounting Type Soldering Conditions (5/5)

(5) μPD70F3030BGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3030BYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3036HGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)
μPD70F3036HYGC-8EU: 100-pin plastic LQFP (fine pitch) (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 230°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)	IR30-103-2
VPS	Package peak temperature: 215°C, Time: 25 to 40 seconds (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: 300°C max., Time: 3 seconds max. (per pin row)	–

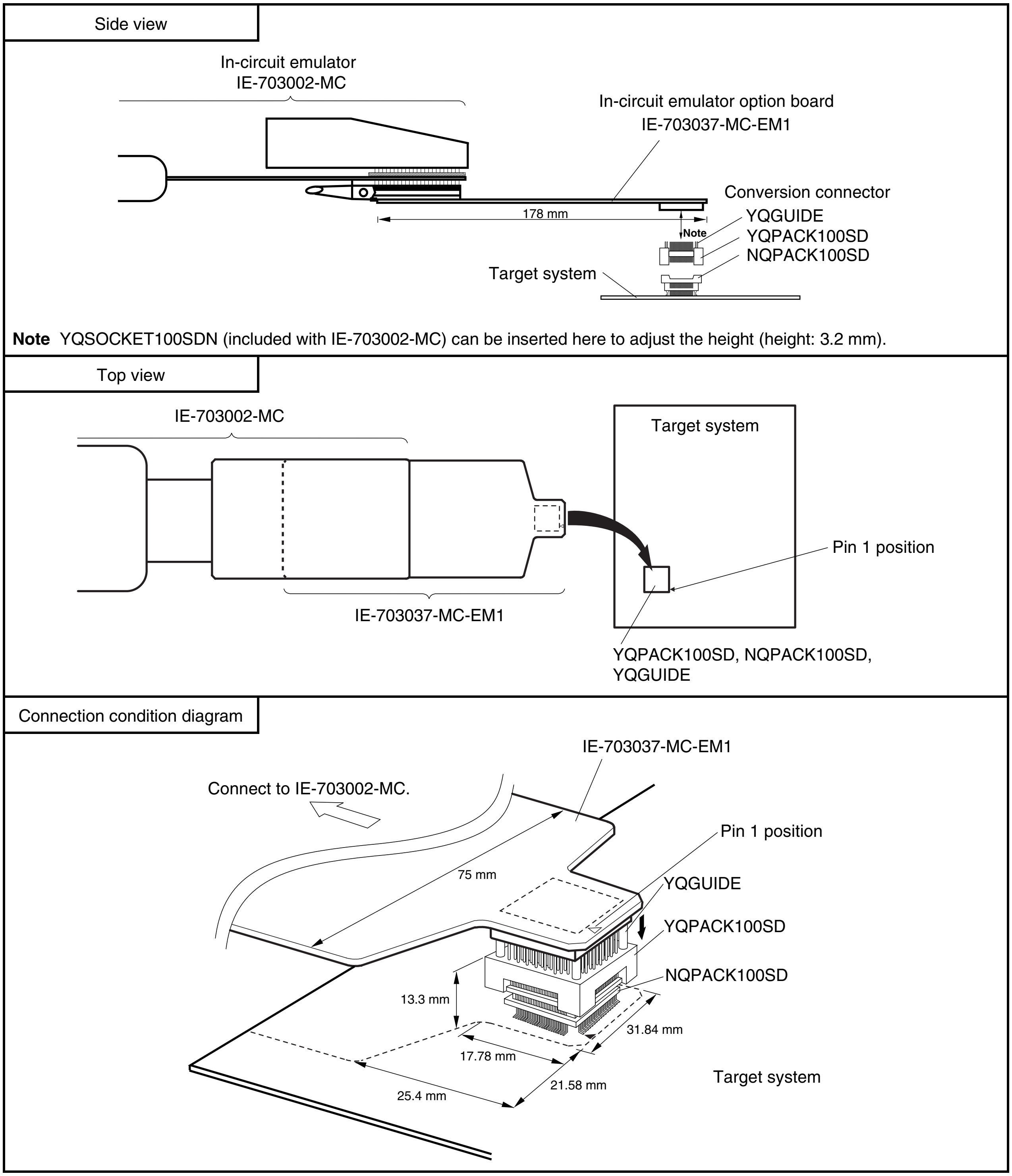
Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

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.

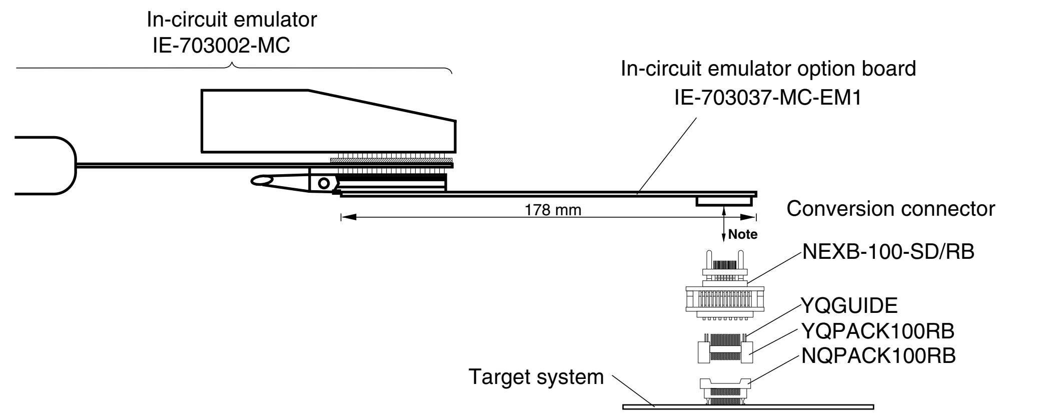
Figure A-1. 100-Pin Plastic LQFP (Fine Pitch) (14 × 14)



Note YQSOCKET100SDN (included with IE-703002-MC) can be inserted here to adjust the height (height: 3.2 mm).

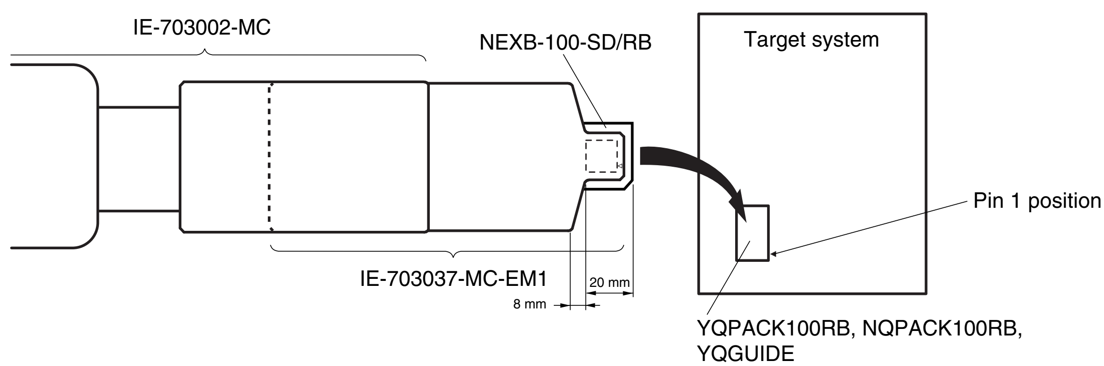
Figure A-2. 100-Pin Plastic QFP (14 × 20)

Side view

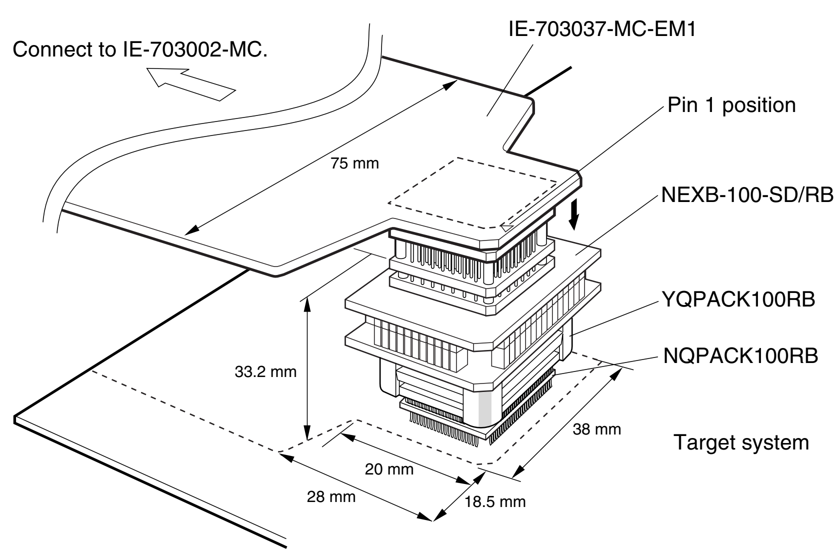


Note YQSOCKET100SDN (included with IE-703002-MC) to this portion for adjusting the height (height: 3.2 mm).

Top view



Connection condition diagram



APPENDIX B REGISTER INDEX

(1/7)

Symbol	Name	Unit	Page
ADCR	A/D conversion result register	ADC	433
ADCRH	A/D conversion result register H	ADC	433
ADIC	Interrupt control register	INTC	162 to 164
ADM1	A/D converter mode register 1	ADC	435
ADM2	A/D converter mode register 2	ADC	437
ADS	Analog input channel specification register	ADC	437
ASIM0	Asynchronous serial interface mode register 0	UART	402
ASIM1	Asynchronous serial interface mode register 1	UART	402
ASIS0	Asynchronous serial interface status register 0	UART	403
ASIS1	Asynchronous serial interface status register 1	UART	403
BCC	Bus cycle control register	BCU	134
BCR	IEBus control register	IEBus	561
BRGC0	Baud rate generator control register 0	UART	404
BRGC1	Baud rate generator control register 1	UART	404
BRGCK4	Baud rate generator output clock selection register 4	CSI	425
BRGCN4	Baud rate generator source clock selection register 4	CSI	424
BRGMC00	Baud rate generator mode control register 00	UART	405
BRGMC01	Baud rate generator mode control register 01	UART	405
BRGMC10	Baud rate generator mode control register 10	UART	405
BRGMC11	Baud rate generator mode control register 11	UART	405
CCR	IEBus communication count register	IEBus	582
CDR	IEBus control data register	IEBus	565
CORAD0	Correction address register 0	CPU	521
CORAD1	Correction address register 1	CPU	521
CORAD2	Correction address register 2	CPU	521
CORAD3	Correction address register 3	CPU	521
CORCN	Correction control register	CPU	519
CORRQ	Correction request register	CPU	520
CR00	16-bit capture/compare register 00	Timer	203
CR01	16-bit capture/compare register 01	Timer	204
CR10	16-bit capture/compare register 10	Timer	203
CR11	16-bit capture/compare register 11	Timer	204
CR20	8-bit compare register 2	Timer	236
CR23	16-bit compare register 23 (when TM2 and TM3 are connected in cascade)	Timer	250
CR30	8-bit compare register 3	Timer	236

(2/7)

Symbol	Name	Unit	Page
CR40	8-bit compare register 4	Timer	236
CR45	16-bit compare register 45 (when TM4 and TM5 are connected in cascade)	Timer	250
CR50	8-bit compare register 5	Timer	236
CR60	8-bit compare register 6	Timer	236
CR67	16-bit compare register 67 (when TM6 and TM7 are connected in cascade)	Timer	250
CR70	8-bit compare register 7	Timer	236
CRC0	Capture/compare control register 0	Timer	207
CRC1	Capture/compare control register 1	Timer	207
CSIB4	Variable-length serial setting register 4	CSI	423
CSIC0	Interrupt control register	INTC	162 to 164
CSIC1	Interrupt control register	INTC	162 to 164
CSIC2	Interrupt control register	INTC	162 to 164
CSIC3	Interrupt control register	INTC	162 to 164
CSIC4	Interrupt control register	INTC	162 to 164
CSIM0	Serial operation mode register 0	CSI	270
CSIM1	Serial operation mode register 1	CSI	270
CSIM2	Serial operation mode register 2	CSI	270
CSIM3	Serial operation mode register 3	CSI	270
CSIM4	Variable-length serial control register 4	CSI	422
CSIS0	Serial clock selection register 0	CSI	270
CSIS1	Serial clock selection register 1	CSI	270
CSIS2	Serial clock selection register 2	CSI	270
CSIS3	Serial clock selection register 3	CSI	270
DBC0	DMA byte counter register 0	DMAC	459
DBC1	DMA byte counter register 1	DMAC	459
DBC2	DMA byte counter register 2	DMAC	459
DBC3	DMA byte counter register 3	DMAC	459
DBC4	DMA byte counter register 4	DMAC	459
DBC5	DMA byte counter register 5	DMAC	459
DCHC0	DMA channel control register 0	DMAC	460
DCHC1	DMA channel control register 1	DMAC	460
DCHC2	DMA channel control register 2	DMAC	460
DCHC3	DMA channel control register 3	DMAC	460
DCHC4	DMA channel control register 4	DMAC	460
DCHC5	DMA channel control register 5	DMAC	460
DIOA0	DMA peripheral I/O address register 0	DMAC	453
DIOA1	DMA peripheral I/O address register 1	DMAC	453
DIOA2	DMA peripheral I/O address register 2	DMAC	453

(3/7)

Symbol	Name	Unit	Page
DIOA3	DMA peripheral I/O address register 3	DMAC	453
DIOA4	DMA peripheral I/O address register 4	DMAC	453
DIOA5	DMA peripheral I/O address register 5	DMAC	453
DLR	IEBus telegraph length register	IEBus	569
DMAIC0	Interrupt control register	INTC	162 to 164
DMAIC1	Interrupt control register	INTC	162 to 164
DMAIC2	Interrupt control register	INTC	162 to 164
DMAIC3	Interrupt control register	INTC	162 to 164
DMAIC4	Interrupt control register	INTC	162 to 164
DMAIC5	Interrupt control register	INTC	162 to 164
DMAS	DMA start factor expansion register	DMAC	459
DR	IEBus data register	IEBus	571
DRA0	DMA internal RAM address register 0	DMAC	454
DRA1	DMA internal RAM address register 1	DMAC	454
DRA2	DMA internal RAM address register 2	DMAC	454
DRA3	DMA internal RAM address register 3	DMAC	454
DRA4	DMA internal RAM address register 4	DMAC	454
DRA5	DMA internal RAM address register 5	DMAC	454
DWC	Data wait control register	BCU	132
ECR	Interrupt source register	CPU	99
EGN0	Falling edge specification register 0	INTC	154, 477
EGP0	Rising edge specification register 0	INTC	154, 477
EIPC	Status saving register during interrupt	CPU	99
EIPSW	Status saving register during interrupt	CPU	99
FEPC	Status saving registers for NMI	CPU	99
FEPSW	Status saving registers for NMI	CPU	99
IEBIC1	Interrupt control register	IEBus	162 to 164
IEBIC2	Interrupt control register	IEBus	162 to 164
IECLK	IEBus clock selection register	IEBus	582
IEHCLK	IEBus high-speed clock selection register	IEBus	583
IIC0	IIC shift register 0	I^2C	278, 291, 352
IIC1	IIC shift register 1	I^2C	278, 291, 352
IICC0	IIC control register 0	I^2C	280, 339
IICC1	IIC control register 1	I^2C	280, 339
IICCE0	IIC clock expansion register 0	I^2C	289, 350
IICCE1	IIC clock expansion register 1	I^2C	289, 350
IICCL0	IIC clock selection register 0	I^2C	288, 349
IICCL1	IIC clock selection register 1	I^2C	288, 349

(4/7)

Symbol	Name	Unit	Page
IICF0	IIC flag register 0	I^2C	347
IICF1	IIC flag register 1	I^2C	347
IICIC1	Interrupt control register	I^2C	160 to 164
IICS0	IIC status register 0	I^2C	285, 344
IICS1	IIC status register 1	I^2C	285, 344
IICX0	IIC function expansion register 0	I^2C	289, 350
IICX1	IIC function expansion register 1	I^2C	289, 350
ISPR	In-service priority register	INTC	165
ISR	IEBus interrupt status register	IEBus	575
KRIC	Interrupt control register	KR	162 to 164
KRM	Key return mode register	KR	180
MAM	Memory address output mode register	Port	116
MM	Memory expansion mode register	Port	115
NCC	Noise elimination control register	INTC	167
OSTS	Oscillation stabilization time selection register	WDT	187, 262, 267
P0	Port 0	Port	474
P1	Port 1	Port	479
P2	Port 2	Port	483
P3	Port 3	Port	488
P4	Port 4	Port	492
P5	Port 5	Port	492
P6	Port 6	Port	495
P7	Port 7	Port	498
P8	Port 8	Port	498
P9	Port 9	Port	500
P10	Port 10	Port	503
P11	Port 11	Port	507
PAC	Port alternate function control register	Port	509
PAR	IEBus partner address register	IEBus	565
PCC	Processor clock control register	CG	184
PF1	Port 1 function register	Port	481
PF2	Port 2 function register	Port	485
PF3	Port 3 function register	Port	490
PF10	Port 10 function register	Port	505
PIC0	Interrupt control register	INTC	162 to 164
PIC1	Interrupt control register	INTC	162 to 164
PIC2	Interrupt control register	INTC	162 to 164
PIC3	Interrupt control register	INTC	162 to 164

(5/7)

Symbol	Name	Unit	Page
PIC4	Interrupt control register	INYC	162 to 164
PIC5	Interrupt control register	INTC	162 to 164
PIC6	Interrupt control register	INTC	162 to 164
PM0	Port 0 mode register	Port	476
PM1	Port 1 mode register	Port	480
PM2	Port 2 mode register	Port	484
PM3	Port 3 mode register	Port	489
PM4	Port 4 mode register	Port	493
PM5	Port 5 mode register	Port	493
PM6	Port 6 mode register	Port	496
PM9	Port 9 mode register	Port	501
PM10	Port 10 mode register	Port	504
PM11	Port 11 mode register	Port	508
PRCMD	Command register	CG	127
PRM00	Prescaler mode register 00	RPU	209
PRM01	Prescaler mode register 01	RPU	209
PRM10	Prescaler mode register 10	RPU	210
PRM11	Prescaler mode register 11	RPU	210
PSC	Power save control register	CG	186
PSW	Program status word	CPU	100
PU0	Pull-up resistor option register 0	Port	476
PU1	Pull-up resistor option register 1	Port	480
PU2	Pull-up resistor option register 2	Port	485
PU3	Pull-up resistor option register 3	Port	489
PU10	Pull-up resistor option register 10	Port	505
PU11	Pull-up resistor option register 11	Port	509
RTBH	Real-time output buffer register H	RTO	468
RTBL	Real-time output buffer register L	RTO	468
RTPC	Real-time output port control register	RTO	470
RTPM	Real-time output port mode register	RTO	469
RX0	Receive shift register 0	UART	400
RX1	Receive shift register 1	UART	400
RXB0	Receive buffer register 0	UART	400
RXB1	Receive buffer register 1	UART	400
SAR	IEBus slave address register	IEBus	433, 564
SCR	IEBus success count register	IEBus	581
SERIC0	Interrupt control register	INTC	162 to 164
SERIC1	Interrupt control register	INTC	162 to 164

(6/7)

Symbol	Name	Unit	Page
SIO0	Serial I/O shift register 0	CSI	269
SIO1	Serial I/O shift register 1	CSI	269
SIO2	Serial I/O shift register 2	CSI	269
SIO3	Serial I/O shift register 3	CSI	269
SIO4	Variable-length serial I/O shift register 4	CSI	420
SSR	IEBus slave status register	IEBus	580
STIC0	Interrupt control register	INTC	162 to 164
STIC1	Interrupt control register	INTC	162 to 164
SVA0	Slave address register 0	I^2C	278, 291, 352
SVA1	Slave address register 1	I^2C	278, 291, 352
SYC	System control register	CG	129
SYS	System status register	CG	127
TCL20	Timer clock selection register 20	Timer	237
TCL21	Timer clock selection register 21	Timer	237
TCL30	Timer clock selection register 30	Timer	237
TCL31	Timer clock selection register 31	Timer	237
TCL40	Timer clock selection register 40	Timer	237
TCL41	Timer clock selection register 41	Timer	237
TCL50	Timer clock selection register 50	Timer	237
TCL51	Timer clock selection register 51	Timer	237
TCL60	Timer clock selection register 60	Timer	237
TCL61	Timer clock selection register 61	Timer	237
TCL70	Timer clock selection register 70	Timer	237
TCL71	Timer clock selection register 71	Timer	237
TM0	16-bit timer register 0	Timer	202
TM1	16-bit timer register 1	Timer	202
TM2	8-bit counter 2	Timer	236
TM23	16-bit counter 23 (when TM2 and TM3 are connected in cascade)	Timer	250
TM3	8-bit counter 3	Timer	236
TM4	8-bit counter 4	Timer	236
TM45	16-bit counter 45 (when TM4 and TM5 are connected in cascade)	Timer	250
TM5	8-bit counter 5	Timer	236
TM6	8-bit counter 6	Timer	236
TM67	16-bit counter 67 (when TM6 and TM7 are connected in cascade)	Timer	250
TM7	8-bit counter 7	Timer	236
TMC0	16-bit timer mode control register 0	Timer	205
TMC1	16-bit timer mode control register 1	Timer	205
TMC2	8-bit timer mode control register 2	Timer	240

(7/7)

Symbol	Name	Unit	Page
TMC3	8-bit timer mode control register 3	Timer	240
TMC4	8-bit timer mode control register 4	Timer	240
TMC5	8-bit timer mode control register 5	Timer	240
TMC6	8-bit timer mode control register 6	Timer	240
TMC7	8-bit timer mode control register 7	Timer	240
TMIC00	Interrupt control register	INTC	162 to 164
TMIC01	Interrupt control register	INTC	162 to 164
TMIC10	Interrupt control register	INTC	162 to 164
TMIC11	Interrupt control register	INTC	162 to 164
TMIC2	Interrupt control register	INTC	162 to 164
TMIC3	Interrupt control register	INTC	162 to 164
TMIC4	Interrupt control register	INTC	162 to 164
TMIC5	Interrupt control register	INTC	162 to 164
TMIC6	Interrupt control register	INTC	162 to 164
TMIC7	Interrupt control register	INTC	162 to 164
TOC0	16-bit timer output control register 0	RPU	208
TOC1	16-bit timer output control register 1	RPU	208
TXS0	Transmit shift register 0	UART	400
TXS1	Transmit shift register 1	UART	400
UAR	IEBus unit address register	IEBus	564
USR	IEBus unit status register	IEBus	572
WDCS	Watchdog timer clock selection register	WDT	263
WDTIC	Interrupt control register	INTC	162 to 164
WDTM	Watchdog timer mode register	WDT	166, 264
WTNCS	Watch timer clock selection register	WT	257
WTNHC	Watch timer high-speed clock selection register	WT	256
WTNIC	Interrupt control register	INTC	162 to 164
WTNIIC	Interrupt control register	INTC	162 to 164
WTNM	Watch timer mode control register	WT	255

• How to Read Instruction Set List

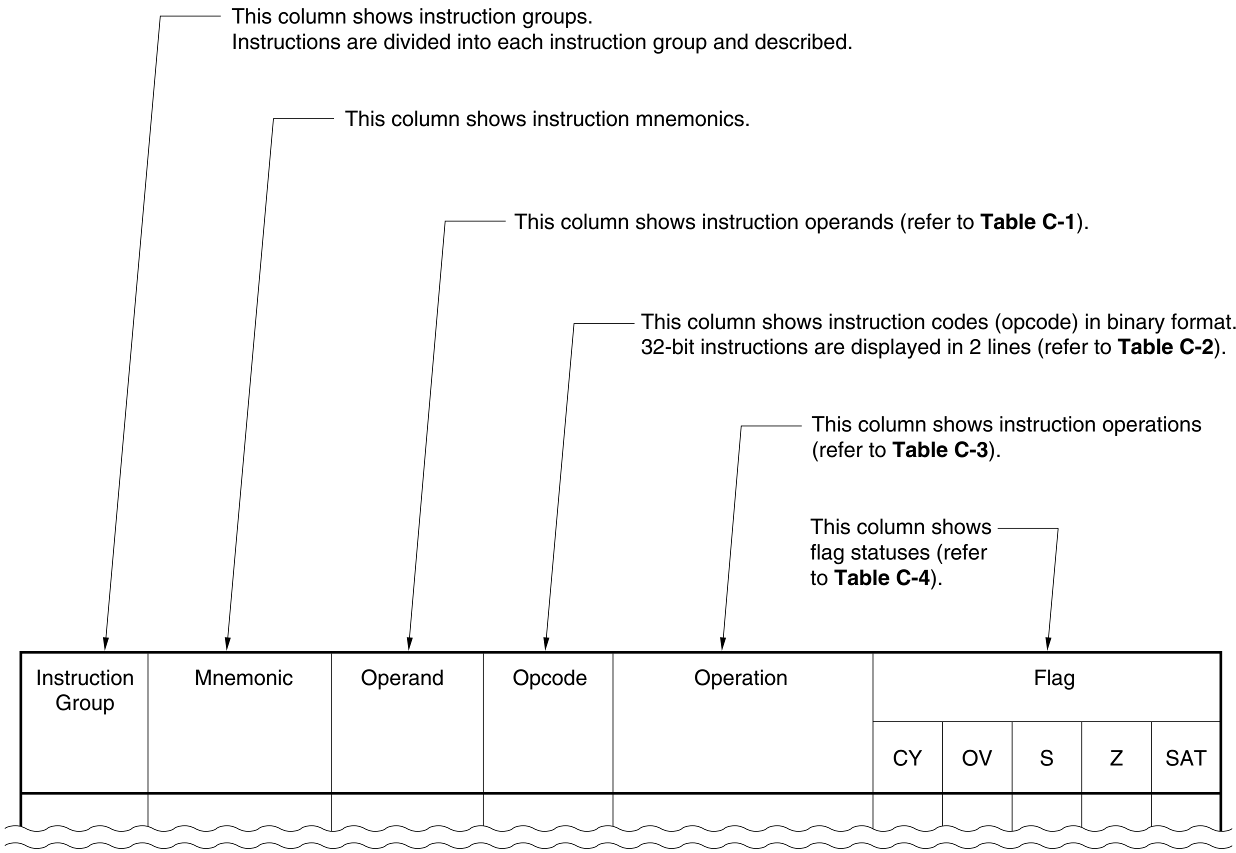


Table C-1. Symbols in Operand Description

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[ ]	Genera-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 complements). 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 of 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	rrrrr1010dddddd0 (**Note 2**)	adr ← ep + zero-extend (disp8) GR [reg2] ← Load-memory (adr, Word)					
	LD.B	disp16 [reg1], reg2	rrrrr111000RRRRR dddddddddddddddd	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]	rrrrr1010dddddd1 (**Note 2**)	adr ← ep + zero-extend (disp8) Store-memory (adr, GR [reg2], Word)					
	ST.B	reg2, disp16 [reg1]	rrrrr111010RRRRR dddddddddddddddd	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 [reg2] 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	rrrrr1111111cccc 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	0000011110dddddd ddddddddddddddd0 (**Note 1**)	PC ← PC + sign-extend (disp22)					
	JARL	disp22, reg2	rrrrr11110dddddd ddddddddddddddd0 (**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. ddddddddddddddddddddd 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 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and opcode is different from that of the other instructions.
rrr = regID specification
RRRRR = reg2 specification

[K]

[L]

[M]

[N]

[O]

★

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.

(1/5)

Edition	Major Revisions from Previous Edition	Applied to:
4th	Modification of **1.2.3 Ordering information (V850/SB1)**	**CHAPTER 1 INTRODUCTION**
	Modification of **1.3.3 Ordering information (V850/SB2)**	
	Modification of description in **2.3 (5) P40 to P47 (Port 4)**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of description in **2.3 (6) P50 to P57 (Port 5)**	
	Modification of description in **2.3 (7) P60 to P65 (Port 6)**	
	Modification of description in **2.3 (9) P90 to P96 (Port 9)**	
	Modification of **Caution** in **2.3 (11) (b) (ii) $\overline{\text{WAIT}}$ (Wait)**	
	Addition of **2.3 (14) CLKOUT (Clock Out)**	
	Addition of **5.8 (1) Acknowledging interrupt servicing after execution of EI instruction**	**CHAPTER 5 INTERRUPT/ EXCEPTION PROCESSING FUNCTION**
	Addition of **6.6 Notes on Power Save Function**	**CHAPTER 6 CLOCK GENERATION FUNCTION**
	Modification of **Caution** in **7.1.3 (2) Capture/compare registers n0 (CR00, CR10)**	**CHAPTER 7 TIMER/COUNTER FUNCTION**
	Modification of **Caution** in **7.1.3 (3) Capture/compare registers n1 (CR01, CR11)**	
	Modification of **Figure 7-34 Data Hold Timing of Capture Register**	
	Addition of **7.2.7 (6) (c) One-shot pulse output function**	
	Modification of **Figure 11-2 A/D Converter Mode Register 1 (ADM1)**	**CHAPTER 11 A/D CONVERTER**
	Addition of description in **11.5 Low Power Consumption Mode**	
	Addition of **Caution** in **CHAPTER 18 FLASH MEMORY**	**CHAPTER 18 FLASH MEMORY**
	Addition of **Table 19-5 Acknowledge Signal Output Condition of Control Field**	**CHAPTER 19 IEBus CONTROLLER (V850/SB2)**
	Addition of description **19.1.8 Bit format**	
	Modification of **Caution** in **19.3.2 (1) (a) Communication enable flag (ENIEBUS)**	
	Addition of **Note** in **Figure 19-18 Timing of INTIE2 Interrupt Generation in Locked State (for (4) and (5))**	
	Addition of **Remark** in **19.3.2 (6) IEBus telegraph length register (DLR)**	
	Addition of **Remark** in **19.3.2 (7) IEBus data register (DR)**	
	Addition of description in **19.3.2 (7) (a) When transmission unit**	
	Modification of description in **19.3.2 (8) (a) Slave request flag (SLVRQ)**	
	Addition of **Caution** in **19.3.2 (8) (b) Arbitration result flag (ARBIT)**	
	Addition of description for **Caution** in **19.3.2 (8) (e) Lock status flag (LOCK)**	
	Addition of **Table 19-8 Reset Condition of Each Flag of ISR Register**	
	Addition of **19.4.3 Communication error source processing list**	
	Modification of **Figure 19-34 Master Transmission (Interval of Interrupt Occurrence)**	
	Modification of **Figure 19-35 Master Reception (Interval of Interrupt Occurrence)**	
	Modification of **Figure 19-36 Slave Transmission (Interval of Interrupt Occurrence)**	
	Modification of **Figure 19-37 Slave Reception (Interval of Interrupt Occurrence)**	

(2/5)

Edition	Major Revisions from Previous Edition	Applied to:
6th	• Addition of the following products. μPD703030B, 703030BY, 703031B, 703031BY, 703032B, 703032BY, 703033B, 703033BY, 703034B, 703034BY, 703035B, 703035BY, 703036H, 703036HY, 703037H, 703037HY, 70F3030B, 70F3030BY, 70F3032B, 70F3032BY, 70F3033B, 70F3033BY, 70F3035B, 70F3035BY, 70F3036H, 70F3036HY, 70F3037H, 70F3037HY • Deletion of the following products. μPD703030A, 703030AY, 703036A, 703036AY	Throughout
	Addition of description on minimum instruction execution time in **1.5.1**	**CHAPTER 1 INTRODUCTION**
	Addition of description on instruction set in **1.5.1**	
	Addition of description in **Table 2-1 Pin I/O Buffer Power Supplies**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of description and addition of **Notes** in **Table 2-3 Operating States of Pins in Each Operating Mode**	
	Addition of description in **2.3 (9) (b) (i) $\overline{\text{LBEN}}$**	
	Modification of P23 I/O circuit type and description on P33 in **2.4 Pin I/O Circuit Types, I/O Buffer Power Supplies and Connection of Unused Pins**	
	Addition of description on minimum instruction execution time in **3.1**	**CHAPTER 3 CPU FUNCTIONS**
	Modification of description and addition of **Note** in **3.2.2 (2) Program status word (PSW)**	
	Addition of **3.4.5 (2) (a) V850/SB1 (uPD703031B, 703031BY), V850/SB2 (μPD703034B, 703034BY)**	
	Modification of **Note** and addition of registers in **3.4.8 Peripheral I/O registers**	
	Addition of description in **3.4.9 Specific registers**	
	Modification of **[Description example]** in **3.4.9 Specific registers**	
	Modification of **Caution 2** in **3.4.9 Specific registers**	
	Addition of **Remarks** in **3.4.9 (2) (b) Reset conditions (PRERR = 1)**	
	Addition of **Note** and **Caution** in **4.2.2 (1) System control register (SYC)**	**CHAPTER 4 BUS CONTROL FUNCTION**
	Addition of **Remark** in **5.3.3 Priorities of maskable interrupts**	**CHAPTER 5 INTERRUPT/ EXCEPTION PROCESSING FUNCTION**
	Addition of **Caution 2** in **5.3.4 Interrupt control register (xxICn)**	
	Addition of **Caution** in **5.3.5 In-service priority register (ISPR)**	
	Addition of **Remark** in **5.3.6 ID flag**	
	Addition of **Remark** in **5.6.2 (2) To generate exception in service program**	
	Addition of **5.8.1 Interrupt request valid timing after EI instruction**	
	Addition of **5.9 Interrupt Control Register Bit Manipulation Instructions During DMA Transfer**	
	Addition of description in **Cautions** in **6.3.1 (1) Processor clock control register (PCC)**	**CHAPTER 6 CLOCK GENERATION FUNCTION**
	Modification of description in **6.3.1 (1) (b) Example of subclock operation → main clock operation setup**	
	Modification of description in **6.3.1 (2) Power save control register (PSC)**	
	Addition and deletion of description in **Table 6-1 Operating Statuses in HALT Mode**	
	Modification of description in **Table 6-2 Operating Statuses in IDLE Mode**	
	Addition of description in **6.4.4 (1) Settings and operating states**	

(3/5)

Edition	Major Revisions from Previous Edition	Applied to:
6th	Modification of description in **Table 6-3 Operating Statuses in Software STOP Mode**	**CHAPTER 6 CLOCK GENERATION FUNCTION**
	Addition of **6.6 (1) While an instruction is being executed on internal ROM**	
	Addition of **6.6 (2) While an instruction is being executed on external ROM**	
	Addition of description in **Caution** in **7.1.4 (1) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**	**CHAPTER 7 TIMER/COUNTER FUNCTION**
	Addition of description in **Caution** in **7.1.4 (2) Capture/compare control registers 0, 1 (CRC0, CRC1)**	
	Modification of description in **Figure 7-5 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**	
	Addition of **Figure 7-6 Configuration of PPG Output**	
	Addition of **Figure 7-7 PPG Output Operation Timing**	
	Modification of description in **Figure 7-8 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**	
	Modification of description in **Figure 7-11 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**	
	Modification of description in **Figure 7-14 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**	
	Modification of description in **Figure 7-17 Timing of Pulse Width Measurement by Restarting (with Rising Edge Specified)**	
	Modification of description in **Figure 7-18 (a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)**	
	Modification of description in **Caution** in **7.2.6 (2) One-shot pulse output with external trigger**	
	Modification of description in **7.2.7 (6) (a) One-shot pulse output by software**	
	Modification of description in **7.2.7 (6) (b) One-shot pulse output with external trigger**	
	Addition of **7.3.1 Outline**	
	Change of **Figure 7-32 Timing of Interval Timer Operation (3/3)**	
	Addition of description to **Remarks** in **Figure 7-34 Square Wave Output Operation Timing**	
	Addition of description to **Remarks** in **Figure 7-35 Timing of PWM Output**	
	Addition of registers and **Caution** in **Figure 8-1 Block Diagram of Watch Timer**	**CHAPTER 8 WATCH TIMER**
	Addition of registers and **Note** in **Table 8-2 Configuration of Watch Timer**	
	Addition of description and **Caution** in **8.3 Watch Timer Control Register**	
	Addition of **8.3 (2) Watch timer high-speed clock selection register (WTNHC)**	
	Addition of description in **8.3 (3) Watch timer clock selection register (WTNCS)**	
	Addition of **Caution** in **9.3 (2) Watchdog timer clock selection register (WDCS)**	**CHAPTER 9 WATCHDOG TIMER**
	Addition of description in **10.2 (2) 3-wire serial I/O mode (fixed to MSB first)**	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Modification of **Caution** in **10.3.2 (1) IIC control registers 0, 1 (IICC0, IICC1)**	
	Addition of **Caution** in **10.3.2 (3) IIC clock selection registers 0, 1 (IICCL0, IICCL1)**	
	Addition of **10.4 I^{2}C Bus (B and H Versions)**	

(4/5)

Edition	Major Revisions from Previous Edition	Applied to:
6th	Addition of description to **Cautions** in **10.5.2 (1) Asynchronous serial interface mode registers 0, 1 (ASIM0, ASIM1)**	**CHAPTER 10 SERIAL INTERFACE FUNCTION**
	Addition to **Cautions** in **10.5.2 (4) Baud rate generator mode control registers n0, n1 (BRGMCn0, BRGMCn1)**	
	Addition of description to **Cautions** in **Figure 10-45 ASIMn Setting (Operation Stop Mode)**	
	Addition of description to **Cautions** in **Figure 10-46 ASIMn Setting (Asynchronous Serial Interface Mode)**	
	Addition of description to **Cautions** in **Figure 10-49 BRGMCn0 and BRGMCn1 Settings (Asynchronous Serial Interface Mode)**	
	Addition of **Caution** in **11.3 (2) Analog input channel specification register (ADS)**	**CHAPTER 11 A/D CONVERTER**
	Addition of **11.7 How to Read A/D Converter Characteristics Table**	
	Addition of **12.3 Configuration**	**CHAPTER 12 DMA FUNCTIONS**
	Addition of **12.4 (2) (a) V850/SB1 (μPD703031B, 703031BY), V850/SB2 (μPD703034B, 703034BY)**	
	Addition of **Caution** in **12.4 (5) DMA channel control registers 0 to 5 (DCHC0 to DCHC5)**	
	Addition of **12.5 Operation**	
	Addition of **12.6 Cautions**	
	Addition of **13.2 Features**	**CHAPTER 13 REAL-TIME OUTPUT FUNCTION (RTO)**
	Addition of **13.3 (2) Output latch**	
	Addition of description in **13.5 Usage**	
	Addition of **13.7 (3)**	
	Addition of description in **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 **Caution** in **14.2.9 (1) Function of P10 pins**	
	Addition of description in **Table 14-12 Setting When Port Pin is Used as Alternate Function**	
	Addition of **14.4 Port Function Operation**	
	Addition of description in **16.1 Outline**	**CHAPTER 16 REGULATOR**
	Addition of description in **Figure 16-1 Regulator**	
	Addition of **18.1.1 (2) V850/SB1 (μPD70F3030B, 70F3030BY), V850/SB2 (uPD70F3036H, 70F3036HY)**	**CHAPTER 18 FLASH MEMORY**
	Addition of **Figure 18-1 Wiring Example of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GC-8EU)**	
	Addition of **Table 18-1 Table for Wiring of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GC-8EU)**	
	Addition of **Figure 18-2 Wiring Example of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GF-3BA)**	
	Addition of **Figure 18-2 Table for Wiring of V850/SB1 and V850/SB2 Flash Writing Adapter (FA-100GF-3BA)**	

(5/5)

Edition	Major Revisions from Previous Edition	Applied to:
6th	Modification of description in **Table 19-5 Acknowledge Signal Output Condition of Control Field**	**CHAPTER 19 IEBus CONTROLLER (V850/SB2)**
	Addition of register to **Table 19-7 Internal Registers of IEBus Controller**	
	Addition of **Remark** in **19.3.2 (13) IEBus clock selection register (IECLK)**	
	Addition of **19.3.2 (14) IEBus high-speed clock selection register (IEHCLK)**	
	Addition of **CHAPTER 20 ELECTRICAL SPECIFICATIONS**	**CHAPTER 20 ELECTRICAL SPECIFICATIONS**
	Addition of **CHAPTER 21 PACKAGE DRAWINGS**	**CHAPTER 21 PACKAGE DRAWINGS**
	Addition of **CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**
	Addition of **APPENDIX A NOTES ON TARGET SYSTEM DESIGN**	**APPENDIX A NOTES ON TARGET SYSTEM DESIGN**
	Modification of **APPENDIX D INDEX**	**APPENDIX D INDEX**
	Addition of **APPENDIX E REVISION HISTORY**	**APPENDIX E REVISION HISTORY**