NEC

User's Manual

V850/SC1™, V850/SC2™, V850/SC3™

32-Bit Single-Chip Microcontrollers

Hardware

μPD703068Y
μPD703069Y
μPD703088Y
μPD703089Y
μPD70F3089Y

Document No. U15109EJ3V0UD00 (3rd edition)
Date Published June 2002 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.

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

License not needed: μPD70F3089Y
The customer must judge the need for license: μPD703068Y, 703069Y, 703088Y, 703089Y

- **The information in this document is current as of May, 2002. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.
  The quality grade of NEC semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.
  (Note)
  (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.
  (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00.4

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC 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 Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

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

- **Branch Sweden**
  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: 253-8311
Fax: 250-3583

J02.4

Major Revisions in This Edition (1/4)

Page	Description
Throughout	• Deletion of indication “under development” for the following products (developed) μPD703068YGJ-×××-UEN, 703069YGJ-×××-UEN • Addition of watch timer high-speed clock select register (WTNHC), IIC flag registers 0 and 1 (IICF0, IICF1)
p.49	Change of minimum instruction execution time in **1.4.1 Features (V850/SC3)**
p.56	Modification of description in **Table 2-1 Pin I/O Buffer Power Supplies**
p.66	Modification of description in **Table 2-3 Pin Operation States in Various Operating Modes**
pp.109, 113, 115, 116	Modification of **3.4.8 Peripheral I/O registers**
p.119	Addition of **Remarks** in **3.4.9 (2) System status register (SYS)**
p.120	Change of frequency of the V850/SC3 in **4.1 (1) Main clock oscillator**
pp.122, 123	Addition of **Note** and **Caution** in **4.3.1 (1) Processor clock control register (PCC)**
p.123	Modification of description for setting DCLK1 and DCLK0 bits = 01B and addition to **Notes** in **4.3.1 (2) Power save control register (PSC)**
p.128	Modification of description on operation status of A16 to A21 pins in **Table 4-1 Operating Statuses in HALT Mode**
p.129	Modification of description on operation of UART0 to UART3 in **Table 4-2 Operating Statuses in IDLE Mode**
p.131	Addition of description in **4.4.4 (1) Settings and operating states**
p.132	Modification of description on operation status of UART0 to UART3 in **Table 4-3 Operating Statuses in Software STOP Mode**
p.135	Addition of **4.6 (1) When executing an instruction on internal ROM**
p.136	Addition of **Caution** in **4.6 (2) When executing an instruction on external ROM**
p.138	Modification of description in **Table 5-1 Pin I/O Buffer Power Supplies**
p.166	Addition of **Caution** in **5.2.8 (1) Function of P9 pins**
pp.192 to 194	Addition and modification of description in **Table 5-16 Setting When Port Pin Is Used for Alternate Function**
p.198	Addition of **5.4 Operation of Port Function**
p.201	Addition of **Note** and **Caution** in **6.2.2 (1) System control register (SYC) (V850/SC1, V850/SC2)**
p.223	Modification of description in **Figure 7-2 Acknowledging Non-Maskable Interrupt Requests**
p.250	Addition of **7.8.1 Interrupt request valid timing following EI instruction**
p.252	Addition of **7.9 Bit Manipulation Instruction of Interrupt Control Register on DMA Transfer**
p.258	Addition and modification of description in **8.1.3 (2) Capture/compare register n0 (CR00, CR10, CR70 to CR120)**
p.259	Addition and modification of description in **8.1.3 (3) Capture/compare register n1 (CR01, CR11, CR71 to CR121)**
p.261	Addition to **Cautions** in **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)**
p.262	Addition to **Cautions** in **8.1.4 (2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**
p.275	Addition of **Figure 8-6 Configuration of PPG Output** and **Figure 8-7 PPG Output Operation Timing**

Major Revisions in This Edition (2/4)

Page	Description
p.288	Change of description of **Caution** in **8.2.6 (2) One-shot pulse output via external trigger**
p.319	Addition of **Caution** in **10.3 (2) Watchdog timer clock select register (WDCS)**
p.324	Addition of description in **11.2 (2) 3-wire serial I/O mode (fixed as MSB first)**
p.327	Addition to **Cautions** in **11.2.2 (1) Serial clock select register n (CSISn) and serial operation mode register n (CSIMn)**
p.350	Modification of description on manipulatable bits in **11.4.3 (6) Clocked serial interface read-only receive buffer registers L5, L6 (SIRBEL5, SIRBEL6)**
p.351	Modification of description on manipulatable bits in **11.4.3 (8) Clocked serial interface transmit buffer registers L5, L6 (SOTBL6, SOTBL5)**
p.352	Modification of description on manipulatable bits in **11.4.3 (10) Clocked serial interface initial transmit buffer registers L5, L6 (SOTBFL5, SOTBFL6)**
p.353	Modification of description on manipulatable bits in **11.4.3 (12) Serial I/O shift registers L5, L6 (SIOL5, SIOL6)**
pp.378, 379	Modification of description and addition to **Note** in **11.5.2 (1) IIC control register 0, 1 (IICC0, IICC1)**
p.385	Addition of **Caution** in **11.5.2 (4) IIC clock expansion registers 0, 1 (IICCE0, IICCE1), IIC function expansion registers 0, 1 (IICX0, IICX1), IIC clock select registers 0, 1 (IICCL0, IICCL1)**
pp.421, 422	Addition of **11.5.12 (2) When communication reservation function is disabled (IICRSVn of IICFn register = 1)**
p.423	Change of description in **11.5.13 Cautions**
p.424	Change of description in **11.5.14 (1) Master operations (1)**
p.425	Addition of **11.5.14 (2) Master operations (2)**
p.426	Addition of description in **Figure 11-39 Slave Operation Flowchart**
p.437	Addition to **Cautions** in **11.6.2 (1) Asynchronous serial interface mode registers 0 to 3 (ASIM0 to ASIM3)**
p.440	Addition to **Cautions** in **11.6.2 (4) Baud rate generator mode control registers n0, n1 (BRGMCn0, BRGMCn1)**
p.441	Addition to **Cautions** in **Figure 11-43 ASIMn Setting (Operation Stopped Mode)**
p.442	Addition to **Cautions** in **Figure 11-44 ASIMn Setting (Asynchronous Serial Interface Mode)**
p.445	Addition to **Cautions** in **Figure 11-47 BRGMCn0 and BRGMCn1 Settings (Asynchronous Serial Interface Mode)**
p.451	Addition of description in **11.6.3 (3) (d) Reception**
p.452	Deletion of description in **11.6.3 (3) (e) Receive error**
p.452	Modification of **Note** in **Figure 11-52 Receive Error Timing**
p.456	Modification of **Caution** in **12.2 (2) A/D conversion result register (ADCR), A/D conversion result register H (ADCRH)**
p.460	Addition of **Caution** in **12.3 (2) Analog input channel specification register (ADS)**
p.467	Modification of description in **12.6 (3) <3> Conflict between writing of ADCR and writing A/D converter mode register 1 (ADM1) or analog input channel specification register (ADS)**
p.470	Modification of description in **12.6 (8) Reading out A/D converter result register (ADCR)**
p.472	Addition of **13.3 Configuration**
p.477	Addition to **Cautions** in **13.4 (6) Start factor settings**

Major Revisions in This Edition (3/4)

Page	Description
pp.479, 480	Addition of **13.5 Operation**
pp.480 to 482	Addition of **13.6 Cautions**
p.483	Modification of description in **14.1 (3) Internal reset by power-on-clear (POC)**
p.487	Modification of description in **14.3 (3) POC control register (POCC)**
p.495	Addition of **Figure 17-1 Example of Wiring of Adapter for Flash Programming (FA-144GJ-UEN)**
p.496	Addition of **Table 17-1 Table for Wiring of Adapter for μPD70F3089Y Flash Programming (FA-144GJ-UEN)**
p.514	Addition of description in **Table 18-5 Control Field Acknowledge Signal Output Conditions**
p.559	Addition of **19.1 Features**
p.559	Modification of description in **Table 19-1 Overview of Functions**
pp.574, 575	Change of manipulatable bits and reset values in **19.4.2 List of FCAN registers**
p.576	Modification of description in **19.5.1 CAN message data length registers 00 to 31 (M_DLC00 to M_DLC31)**
pp.577, 578	Modification of description in **19.5.2 CAN message control registers 00 to 31 (M_CTRL00 to M_CTRL31)**
p.585	Addition of description in **19.5.6 CAN message configuration registers 00 to 31 (M_CONF00 to M_CONF31)**
p.587	Modification of description in **19.5.7 CAN message status registers 00 to 31 (M_STAT00 to M_STAT31)**
p.593	Modification of description on manipulatable bits and modification of register format and bit description in **19.5.10 CAN global interrupt pending register (CGINTP)**
pp.594, 595	Modification of description on manipulatable bits and modification of register format in **19.5.11 CANn interrupt pending register (CnINTP)**
p.596	Addition to **Cautions** in **19.5.12 CAN stop register (CSTOP)**
pp.597, 598	Modification of description on manipulatable bits and modification of bit description in **19.5.13 CAN global status register (CGST)**
p.600	Modification of description on manipulatable bits and modification of bit description in **19.5.14 CAN global interrupt enable register (CGIE)**
p.601	Addition of description in **19.5.15 CAN main clock select register (CGCS)**
p.602	Deletion of **Caution** in **Figure 19-2 FCAN Clocks**
pp.604, 605	Addition of **Cautions** and bit name, and modification of bit description in **19.5.17 CAN message search start/result register (CGMSS/CGMSR)**
p.606	Addition of description in **19.5.18 CANn address mask a registers L and H (CnMASKLa and CnMASKHa)**
p.610	Addition of description in **19.5.19 CANn control register (CnCTRL)**
pp.612 to 614	Modification of description on manipulatable bits and modification of bit description in **19.5.20 CANn definition register (CnDEF)**
pp.618, 619	Modification of description on manipulatable bits and addition of bit description in **19.5.23 CANn interrupt enable register (CnIE)**
pp.626, 627	Modification of description in **Cautions** and addition of bit description in **19.5.27 CANn synchronization control register (CnSYNC)**

Major Revisions in This Edition (4/4)

Page	Description
p.630	Addition of **Caution** in **19.7 Time Stamp Function**
p.633	Modification of description in **19.8 Message Processing**
p.638	Change of **Figure 19-10 Composition of Layers**
p.653	Addition of **Caution** in **19.11.7 (2) Nominal bit time (8 to 25 time quanta)**
p.654	Addition to **Note** in **Figure 19-25 Nominal Bit Time**
p.656	Addition of description in **Figure 19-28 Initialization Processing**
p.659	Addition of **Note** in **Figure 19-33 Setting of CANn Synchronization Control Register (CnSYNC)**
p.664	Addition of description in **Figure 19-38 Message Buffer Setting**
p.667	Addition of **Figure 19-41 Setting of CAN Message Status Registers 00 to 31 (M_STAT00 to M_STAT31)**
p.670	Addition of **Figure 19-44 Setting Receive Operation Using Reception Polling**
p.671	Addition of **Figure 19-45 Setting of CAN Message Search Start/Result Register (CGMSS/CGMSR)**
p.674	Addition of description in **Figure 19-49 CAN Stop Mode Setting**
p.674	Addition of description in **Figure 19-50 Clearing CAN Stop Mode**
p.675	Modification of description in **19.13 Rules for Correct Setting of Baud Rate**
p.680	Addition to **Cautions** in **19.14.2 Burst read mode**
p.682	Deletion of **Caution 2** in **19.16 How to Shutdown FCAN Controller**
pp.682, 683	Addition of <4> and <5> in **19.17 Cautions on Use**
p.684	Addition of **CHAPTER 20 ELECTRICAL SPECIFICATIONS**
p.712	Addition of **CHAPTER 21 PACKAGE DRAWING**
p.713	Addition of **CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**
p.732	Addition of **APPENDIX C 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/SC1, V850/SC2, and V850/SC3 to design application systems using the V850/SC1, V850/SC2, and V850/SC3.

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

Organization The V850/SC1, V850/SC2, and V850/SC3 User's Manual is divided into two parts: hardware (this manual) and architecture (V850 Series™ Architecture User's Manual).

Hardware	Architecture
• Pin functions	• Data types
• CPU functions	• Register set
• On-chip peripheral functions	• Instruction format and instruction set
• Flash memory programming	• Interrupts and exceptions
• IEBus™ controller	• Pipeline operation
• FCAN controller	
• 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 A REGISTER INDEX**.

To understand the details of an instruction function:

→ Refer to **V850 Series Architecture User's Manual,** available separately.

To know the electrical specifications of the V850/SC1, V850/SC2, and V850/SC3:

→ Refer to **CHAPTER 20 ELECTRICAL SPECIFICATIONS**.

How to read register formats:

→ Names of bits whose numbers are enclosed in < > are defined in the device file under reserved words.

To understand the overall functions of the V850/SC1, V850/SC2, and V850/SC3:

→ 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.

Documents related to V850/SC1, V850/SC2, V850/SC3

Document Name	Document No.
V850 Series Architecture User's Manual	U10243E
V850/SC1, V850/SC2, V850/SC3 Hardware User's Manual	This manual

Documents related to development tools (user's manuals)

Document Name		Document No.
IE-703002-MC (In-Circuit Emulator)		U11595E
IE-703089-MC-EM1 (In-Circuit Emulator Option Board)		To be prepared
CA850 Ver. 2.30 or Later C Compiler Package	Operation	U14568E
	C Language	U14566E
	Project Manager	U14569E
	Assembly Language	U14567E
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	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.13 Real-Time OS	Fundamental	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 (1/8)

LIST OF FIGURES (2/8)

LIST OF FIGURES (3/8)

LIST OF FIGURES (4/8)

LIST OF FIGURES (5/8)

LIST OF FIGURES (6/8)

LIST OF FIGURES (7/8)

LIST OF FIGURES (8/8)

LIST OF TABLES (1/4)

LIST OF TABLES (2/4)

LIST OF TABLES (3/4)

LIST OF TABLES (4/4)

CHAPTER 1 INTRODUCTION

The V850/SC1, V850/SC2, and V850/SC3 are products in NEC's V850 Series of single-chip microcontrollers designed for low-power operation.

1.1 General

The V850/SC1, V850/SC2, and V850/SC3 are 32-bit single-chip microcontrollers that include the V850 Series' CPU core, and peripheral functions such as ROM/RAM, a timer/counter, serial interfaces, an A/D converter, and a DMA controller.

The V850/SC2 is equivalent to the V850/SC1, but with an IEBus (Inter Equipment Bus™) controller added to the peripheral functions.

The V850/SC3 has an FCAN (Full Controller Area Network) controller added to the peripheral functions.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850/SC1, V850/SC2, and V850/SC3 can realize multiply, saturated operation, and bit manipulation instructions by means of a hardware multiplier.

Table 1-1 shows the outline of the V850/SC1, V850/SC2, and V850/SC3 lineup.

Table 1-1. Product Lineup of V850/SC1, V850/SC2, and V850/SC3

Product Name		ROM		RAM Size	FCAN	IEBus
Commercial Name	Part Number	Type	Size			
V850/SC1	μPD703068Y	Mask ROM	512 KB	24 KB	–	–
	μPD70F3089Y	Flash memory	512 KB	24 KB	–	–
V850/SC2	μPD703069Y	Mask ROM	512 KB	24 KB	–	1 channel
	μPD70F3089Y	Flash memory	512 KB	24 KB	–	
V850/SC3	μPD703088Y	Mask ROM	512 KB	24 KB	1 channel	–
	μPD703089Y		512 KB	24 KB	2 channels	–
	μPD70F3089Y	Flash memory	512 KB	24 KB		–

1.2 V850/SC1

1.2.1 Features (V850/SC1)

- 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)
  Saturated 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 expansion: 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 multiplexed)
  Address bus: Separate output possible
  3 V to 5 V interface enabled
  Bus hold function
  External wait function
- Internal memory:
  μPD703068Y (mask ROM: 512 KB/RAM: 24 KB)
  μPD70F3089Y (flash memory: 512 KB/RAM: 24 KB)
- Interrupts and exceptions:
  Non-maskable interrupts: 2 sources
  Maskable interrupts: 49 sources
  Software exceptions: 32 sources
  Exception trap: 1 source
- I/O lines:
  Total: 124 (12 input ports and 112 I/O ports)
  3 V to 5 V interface enabled
- Timer/counter:
  16-bit timer: 8 channels (TM0, TM1, TM7 to TM12)
  16-bit timer: 2 channels (TM5, TM6)
- Watch timer:
  When operating on 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 interfaces (SIO)	Asynchronous serial interface (UART)	
	3-wire serial I/O (CSI)	
	I^2C bus interface (I^2C)	
	8- to 16-bit variable-length serial interface	
	CSI (8-bit)/UART:	1 channel
	CSI (8- to 16-bit variable)/UART:	1 channel
	CSI (8-bit)/I^2C:	2 channels
	CSI (8- or 16-bit):	2 channels
	UART:	2 channels
	Dedicated baud rate generator:	5 channels
○ A/D converter	10-bit resolution: 12 channels	
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels	
○ ROM correction	4 correction addresses specifiable	
○ Regulator	3.5 V to 5.5 V input → internal 3.3 V	
○ Key return function	4 to 8 pins selectable, falling edge fixed	
○ Clock generator	During main clock or subclock operation	
	5-level CPU clock (including sub operations)	
○ Power save functions	HALT/IDLE/STOP modes	
○ Package	144-pin plastic LQFP (20 × 20 mm)	
○ CMOS structure	All static circuits	

1.2.2 Application fields (V850/SC1)

AV equipment such as car audio and home audio

★ 1.2.3 Ordering information (V850/SC1)

Part Number	Package	Internal ROM
μPD703068YGJ-×××-UEN	144-pin plastic LQFP (20 × 20)	Mask ROM
μPD70F3089YGJ-UEN[Note]	144-pin plastic LQFP (20 × 20)	Flash memory

Note Under development

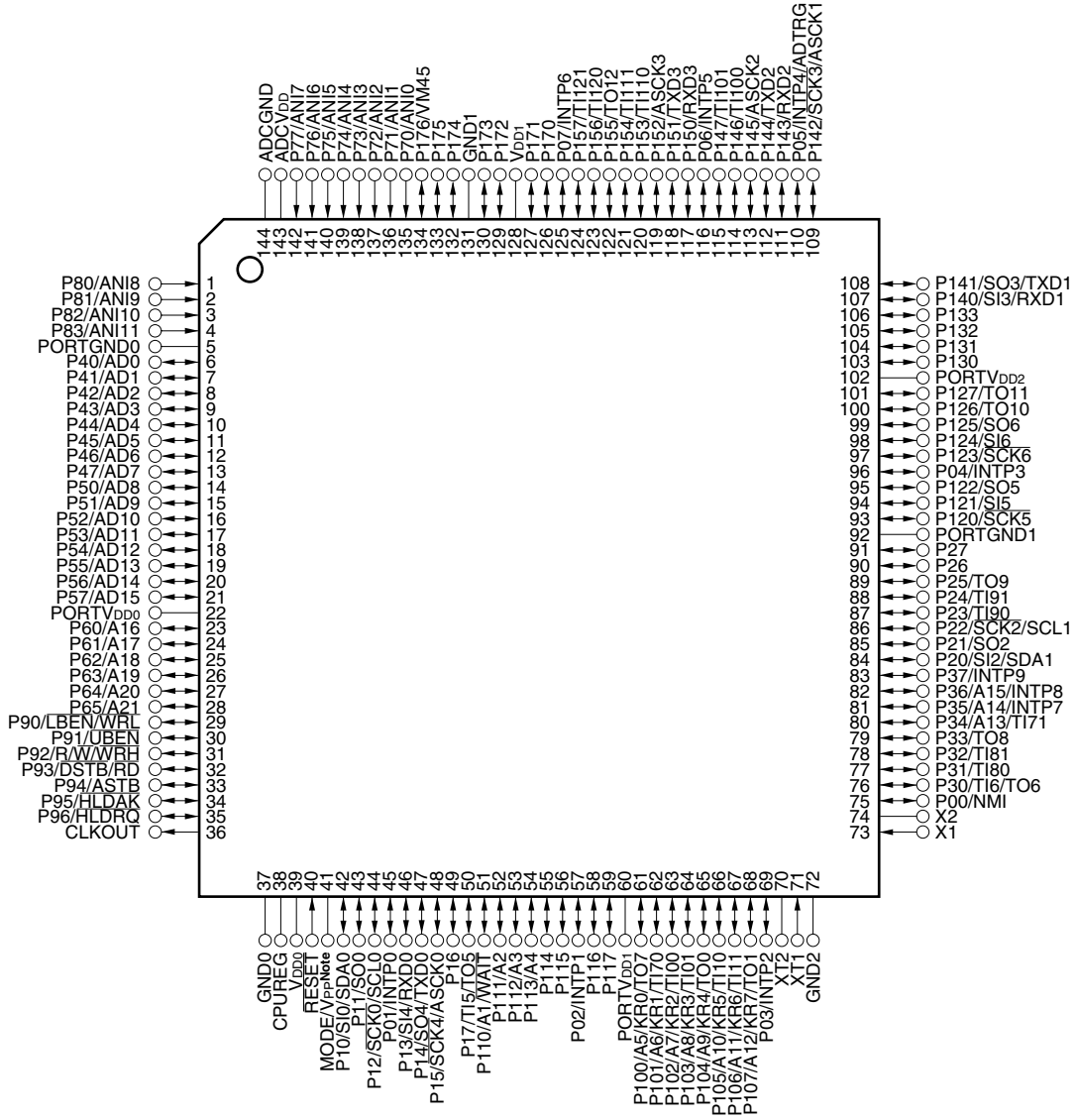
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.2.4 Pin configuration (top view) (V850/SC1)

144-pin plastic LQFP (20 × 20)

- μPD703068YGJ-×××-UEN
- μPD70F3089YGJ-UEN



Note μPD703068Y: MODE

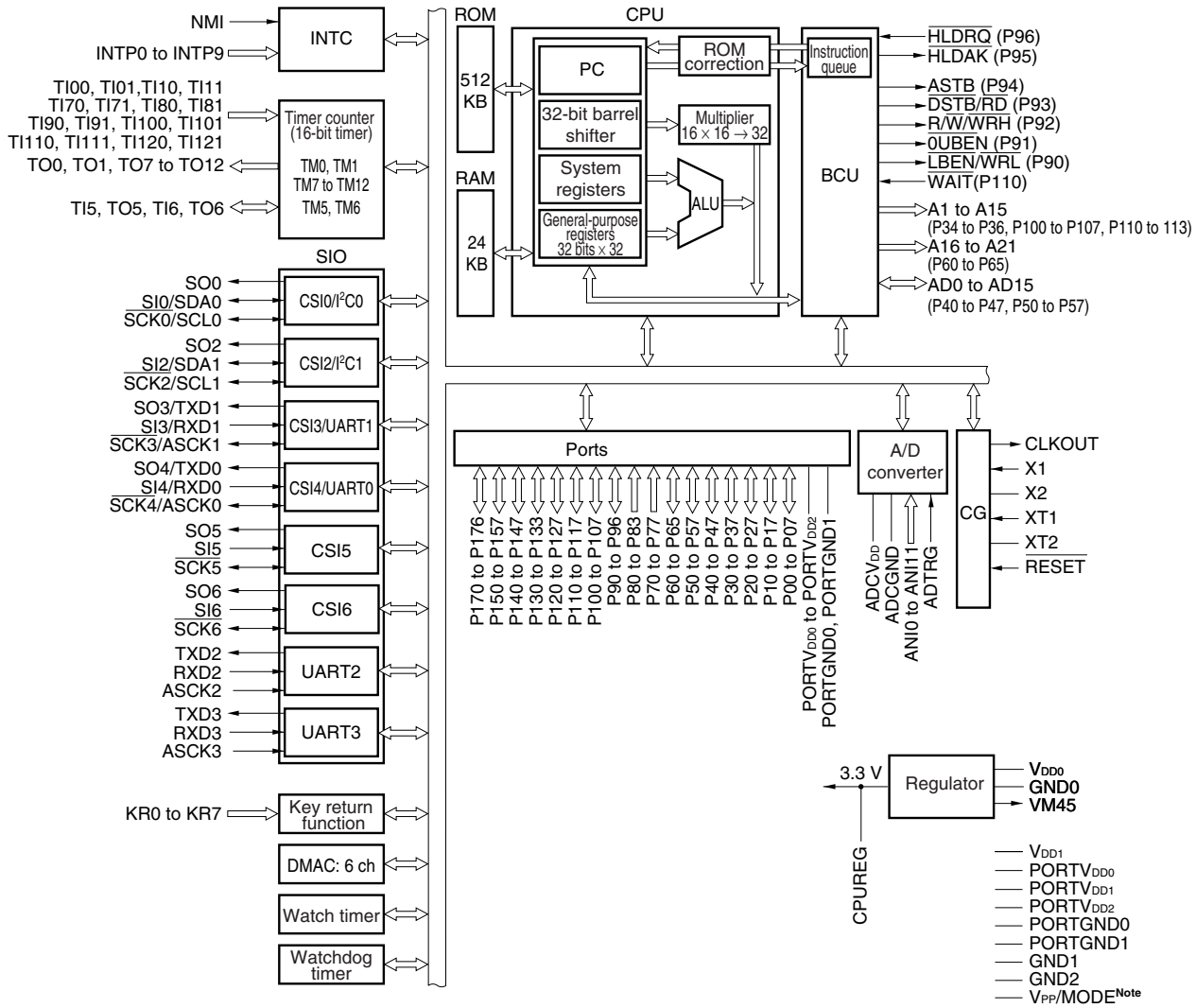
μPD70F3089Y: VPP (connect to either GND0, GND1, or GND2 in normal operation mode)

Pin names (V850/SC1)

A1 to A21:	Address bus	P120 to P127:	Port 12
AD0 to AD15:	Address/data bus	P130 to P133:	Port 13
ADCGND:	Ground for analog	P140 to P147:	Port 14
$ADCV_{DD}$	Power supply for analog	P150 to P157:	Port 15
ADTRG:	A/D trigger input	P170 to P176:	Port 17
ANI0 to ANI11:	Analog input	$\overline{RD}$:	Read strobe
ASCK0 to ASCK3:	Asynchronous serial clock	$\overline{RESET}$:	Reset
ASTB:	Address strobe	R/$\overline{W}$:	Read/write status
CLKOUT:	Clock output	RXD0 to RXD3:	Receive data
CPUREG:	Regulator control	$\overline{SCK0}$, $\overline{SCK2}$ to	
$\overline{DSTB}$:	Data strobe	$\overline{SCK6}$:	Serial clock
GND0 to GND2:	Ground	SCL0, SCL1:	Serial clock
$\overline{HLDAK}$:	Hold acknowledge	SDA0, SDA1:	Serial data
$\overline{HLDRQ}$:	Hold request	SI0, SI2 to SI6:	Serial input
INTP0 to INTP9:	Interrupt request from peripherals	SO0, SO2 to SO6:	Serial output
KR0 to KR7:	Key return	TI00, TI01, TI10,	
$\overline{LBEN}$:	Lower byte enable	TI11, TI100, TI101,	
MODE:	Mode	TI110, TI111, TI120,	
NMI:	Non-maskable interrupt request	TI121, TI5, TI6, TI70,	
PORTGND0,		TI71, TI80, TI81,	
PORTGND1:	Ground for ports	TI90, TI91:	Timer input
$PORTV_{DD0}$ to		TO0, TO1,	
$PORTV_{DD2}$:	Power supply for ports	TO5 to TO12:	Timer output
P00 to P07:	Port 0	TXD0 to TXD3:	Transmit data
P10 to P17:	Port 1	$\overline{UBEN}$:	Upper byte enable
P20 to P27:	Port 2	V_{DD0}, V_{DD1}:	Power supply
P30 to P37:	Port 3	VM45:	V_{DD} = 4.5 V monitor output
P40 to P47:	Port 4	V_{PP}:	Programming power supply
P50 to P57:	Port 5	$\overline{WAIT}$:	Wait
P60 to P65:	Port 6	$\overline{WRH}$:	Write strobe high-level data
P70 to P77:	Port 7	$\overline{WRL}$:	Write strobe low-level data
P80 to P83:	Port 8	X1, X2:	Crystal for main clock
P90 to P96:	Port 9	XT1, XT2:	Crystal for sub-clock
P100 to P107:	Port 10		
P110 to P117:	Port 11		

1.2.5 Function blocks (V850/SC1)

(1) Internal block diagram



Note μPD703068Y: MODE

μPD70F3089Y: V_{PP} (connect to either GND0, GND1, or GND2 in normal operating mode)

(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 a multiplier (16 bits × 16 bits → 32 bits) and a 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 512 KB mask ROM or flash memory mapped to the address space starting at 00000000H.

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

(d) RAM

This consists of a 24 KB RAM mapped to the address space starting 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 INTP9) 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: one for the main clock (f_{XX}) and one for the 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 ten-channel 16-bit timer/event counter is incorporated, enabling measurement of pulse intervals and frequency as well as programmable pulse output.

(h) Watch timer

This timer counts the reference time period (0.5 second) for counting the clock (the 32.768 kHz subclock or the 8.388 MHz 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 provided to detect inadvertent program loops and 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 interfaces (SIO)

The V850/SC1 includes four kinds of serial interfaces: an asynchronous serial interface (UARTm), a 3-wire serial I/O (CSIn), and an I^2C bus interface (I^2Cx), which can use up to eight channels at the same time. Two of these channels are switchable between the UART and CSI and another two are switchable between CSI and I^2C.

For UARTm, data is transferred via the TXDm and RXDm pins.

For CSIn, data is transferred via the SOn, SIn, and $\overline{SCKn}$ pins.

For I^2Cx, data is transferred via the SDAx and SCLx pins.

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

Remark m = 0 to 3
n = 0, 2 to 6
x = 0, 1

(k) A/D converter

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

(l) DMA controller

A six-channel DMA controller is incorporated. 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) 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
Port 1	8-bit I/O		Serial interface, timer I/O
Port 2	8-bit I/O		Serial interface, timer I/O
Port 3	8-bit I/O		Timer I/O, external address bus, external interrupt
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		Timer I/O, key return input, external address bus
Port 11	8-bit I/O		Wait control, external address bus
Port 12	8-bit I/O		Serial interface, timer output
Port 13	4-bit I/O		–
Port 14	8-bit I/O		Serial interface, timer input
Port 15	8-bit I/O		Serial interface, timer I/O
Port 17	7-bit I/O		V_{DD0} = 4.5 V monitor output

1.3 V850/SC2

1.3.1 Features (V850/SC2)

- Number of instructions: 74
- Minimum instruction execution time
  53 ns (operating at 18.87 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): 106 ns (operating at 18.87 MHz)
  (Able to execute instructions in parallel continuously without creating any register hazards)
  Saturated 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 expansion: 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 multiplexed)
  Address bus: Separate output possible
  3 V to 5 V interface enabled
  Bus hold function
  External wait function
- Internal memory:
  μPD703069Y (mask ROM: 512 KB/RAM: 24 KB)
  μPD70F3089Y (flash memory: 512 KB/RAM: 24 KB)
- Interrupts and exceptions:
  Non-maskable interrupts: 2 sources
  Maskable interrupts: 51 sources
  Software exceptions: 32 sources
  Exception trap: 1 source
- I/O lines:
  Total: 124 (12 input ports and 112 I/O ports)
  3 V to 5 V interface enabled
- Timer/counter:
  16-bit timer: 8 channels (TM0, TM1, TM7 to TM12)
  16-bit timer: 2 channels (TM5, TM6)
- Watch timer:
  When operating on 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 interfaces (SIO)	Asynchronous serial interface (UART)	
	3-wire serial I/O (CSI)	
	I^2C bus interface (I^2C)	
	8- to 16-bit variable-length serial interface	
	CSI (8-bit)/UART:	1 channel
	CSI (8- to 16-bit variable)/UART:	1 channel
	CSI (8-bit)/I^2C:	2 channels
	CSI (8- or 16-bit):	2 channels
	UART:	2 channels
	Dedicated baud rate generator:	5 channels
○ A/D converter	10-bit resolution: 12 channels	
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels	
○ ROM correction	4 correction addresses specifiable	
○ Regulator	3.5 V to 5.5 V input → internal 3.3 V	
○ Key return function	4 to 8 pins selectable, falling edge fixed	
○ Clock generator	During main clock or subclock operation	
	5-level CPU clock (including sub operations)	
○ Power save functions	HALT/IDLE/STOP modes	
○ IEBus controller	1 channel	
○ Package	144-pin plastic LQFP (20 × 20 mm)	
○ CMOS structure	All static circuits	

1.3.2 Application fields (V850/SC2)

Car audio equipment

★ 1.3.3 Ordering information (V850/SC2)

Part Number	Package	Internal ROM
μPD703069YGJ-×××-UEN	144-pin plastic LQFP (20 × 20)	Mask ROM
μPD70F3089YGJ-UEN[Note]	144-pin plastic LQFP (20 × 20)	Flash memory

Note Under development

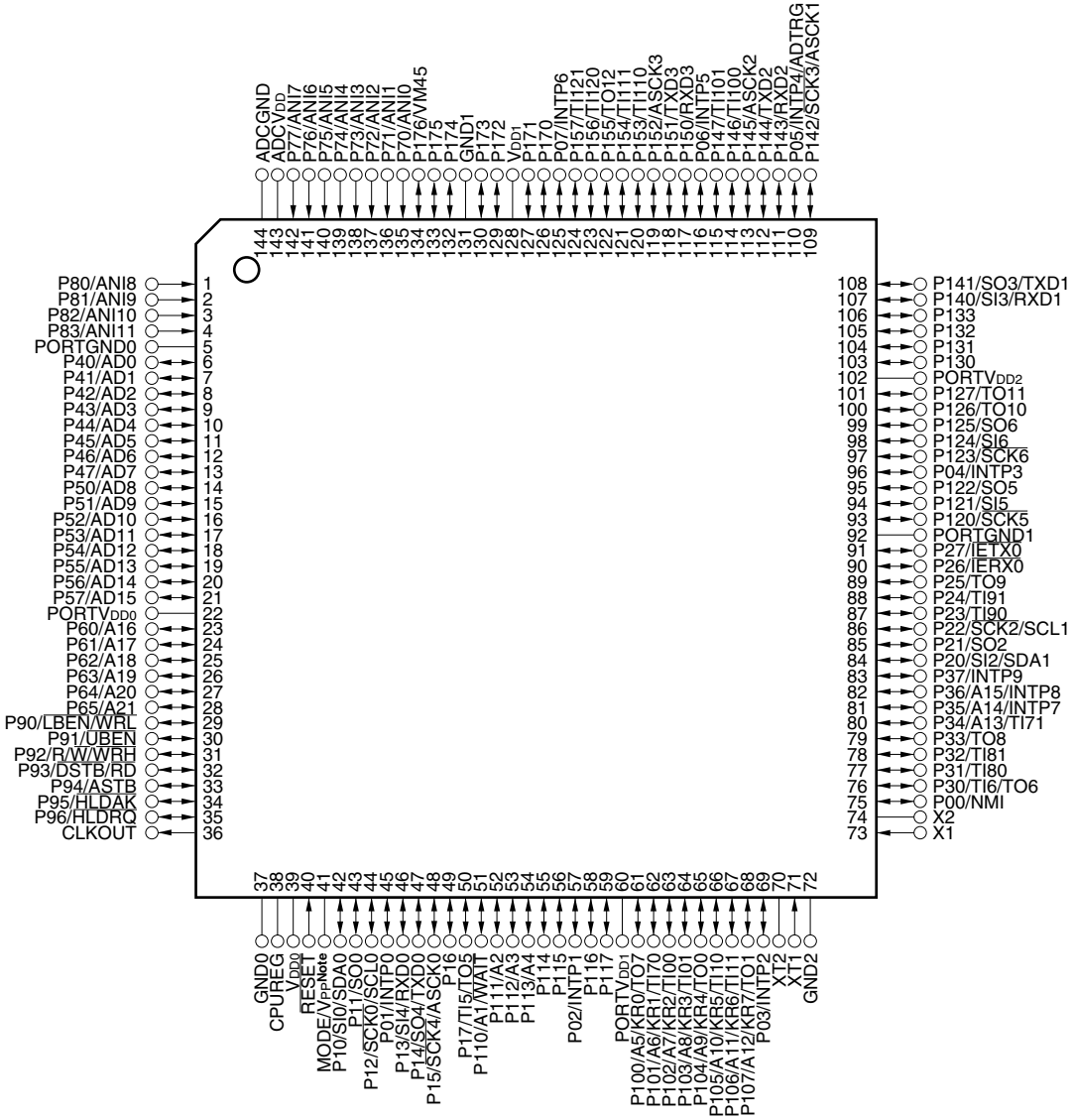
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.3.4 Pin configuration (top view) (V850/SC2)

144-pin plastic LQFP (20 × 20)

- μPD703069YGJ-×××-UEN
- μPD70F3089YGJ-UEN



Note μPD703069Y: MODE

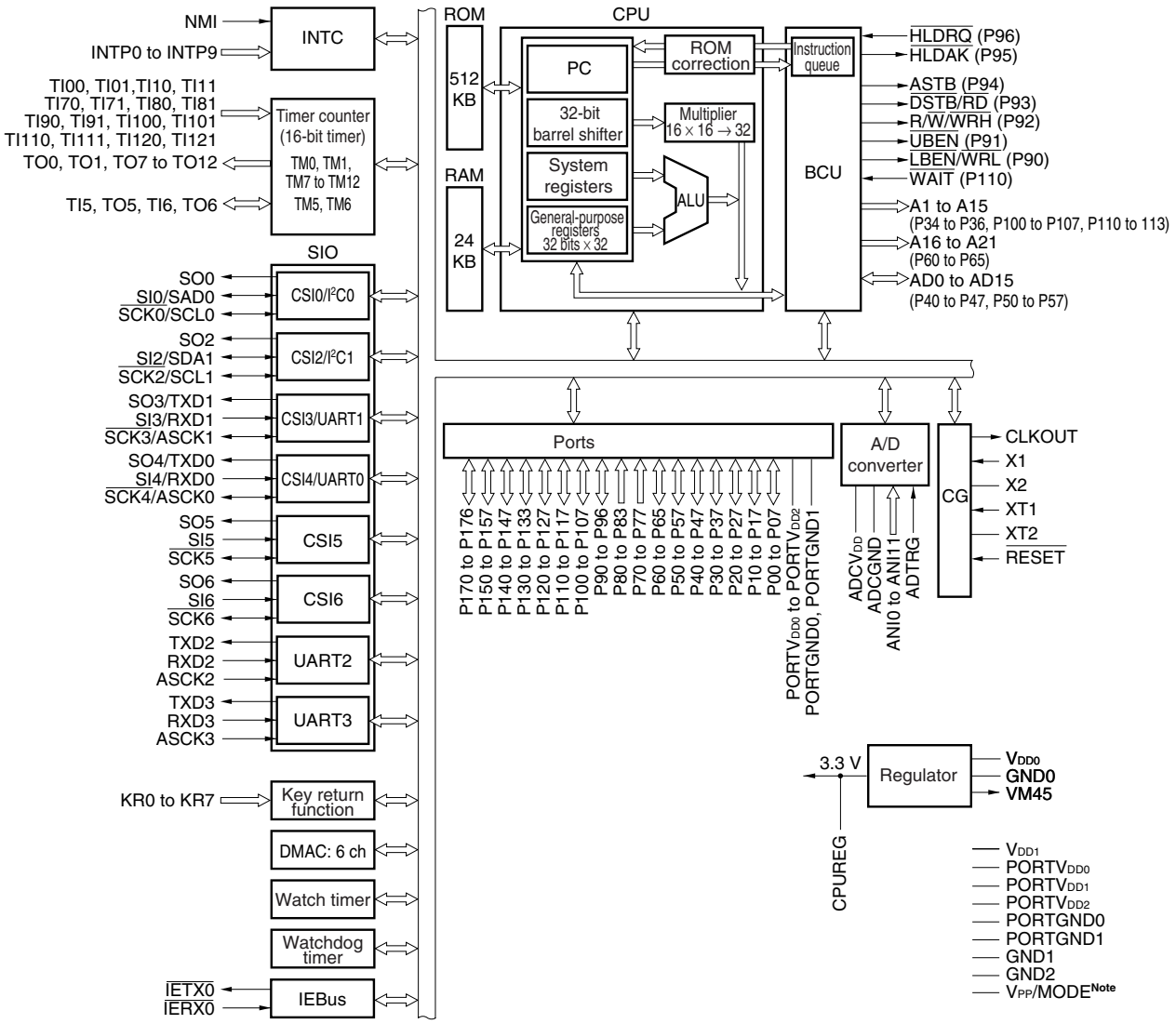
μPD70F3089Y: VPP (connect to either GND0, GND1, or GND2 in normal operating mode)

Pin names (V850/SC2)

A1 to A21:	Address bus	P100 to P107:	Port 10
AD0 to AD15:	Address/data bus	P110 to P117:	Port 11
ADCGND:	Ground for analog	P120 to P127:	Port 12
$ADCV_{DD}$	Power supply for analog	P130 to P133:	Port 13
ADTRG:	A/D trigger input	P140 to P147:	Port 14
ANI0 to ANI11:	Analog input	P150 to P157:	Port 15
ASCK0 to ASCK3:	Asynchronous serial clock	P170 to P176:	Port 17
ASTB:	Address strobe	$\overline{RD}$:	Read strobe
CLKOUT:	Clock output	$\overline{RESET}$:	Reset
CPUREG:	Regulator control	R/$\overline{W}$:	Read/write status
$\overline{DSTB}$:	Data strobe	RXD0 to RXD3:	Receive data
GND0 to GND2:	Ground	$\overline{SCK0}$, $\overline{SCK2}$ to	
$\overline{HLDAK}$:	Hold acknowledge	$\overline{SCK6}$:	Serial clock
$\overline{HLDRQ}$:	Hold request	SCL0, SCL1:	Serial clock
$\overline{IERX0}$:	IEBus receive data	SDA0, SDA1:	Serial data
$\overline{IETX0}$:	IEBus transmit data	SI0, SI2 to SI6:	Serial input
INTP0 to INTP9:	Interrupt request from peripherals	SO0, SO2 to SO6:	Serial output
KR0 to KR7:	Key return	TI00, TI01, TI10,	
$\overline{LBEN}$:	Lower byte enable	TI11, TI100, TI101,	
MODE:	Mode	TI110, TI111, TI120,	
NMI:	Non-maskable interrupt request	TI121, TI5, TI6, TI70,	
PORTGND0,		TI71, TI80, TI81,	
PORTGND1:	Ground for ports	TI90, TI91:	Timer input
$PORTV_{DD0}$ to		TO0, TO1,	
$PORTV_{DD2}$:	Power supply for ports	TO5 to TO12:	Timer output
P00 to P07:	Port 0	TXD0 to TXD3:	Transmit data
P10 to P17:	Port 1	$\overline{UBEN}$:	Upper byte enable
P20 to P27:	Port 2	V_{DD0}, V_{DD1}:	Power supply
P30 to P37:	Port 3	VM45:	V_{DD} = 4.5 V monitor output
P40 to P47:	Port 4	V_{PP}:	Programming power supply
P50 to P57:	Port 5	$\overline{WAIT}$:	Wait
P60 to P65:	Port 6	$\overline{WRH}$:	Write strobe high-level data
P70 to P77:	Port 7	$\overline{WRL}$:	Write strobe low-level data
P80 to P83:	Port 8	X1, X2:	Crystal for main clock
P90 to P96:	Port 9	XT1, XT2:	Crystal for sub-clock

1.3.5 Function blocks (V850/SC2)

(1) Internal block diagram



Note μPD703069Y: MODE

μPD70F3089Y: V_{PP} (connect to either GND0, GND1, or GND2 in normal operating mode)

(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 a multiplier (16 bits × 16 bits → 32 bits) and a 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 512 KB mask ROM or flash memory mapped to the address space starting at 00000000H.

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

(d) RAM

This consists of a 24 KB RAM mapped to the address space starting 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 INTP9) 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: one for the main clock (f_{XX}) and one for the 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 ten-channel 16-bit timer/event counter is incorporated, enabling measurement of pulse intervals and frequency as well as programmable pulse output.

(h) Watch timer

This timer counts the reference time period (0.5 second) for counting the clock (the 32.768 kHz subclock or the 8.388 MHz 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 provided to detect inadvertent program loops and 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 interfaces (SIO)

The V850/SC2 includes four kinds of serial interfaces: an asynchronous serial interface (UARTm), a 3-wire serial I/O (CSIn), and an I^2C bus interface (I^2Cx), which can use up to eight channels at the same time. Two of these channels are switchable between the UART and CSI and another two are switchable between CSI and I^2C.

For UARTm, data is transferred via the TXDm and RXDm pins.

For CSIn, data is transferred via the SOn, SIn, and $\overline{SCKn}$ pins.

For I^2Cx, data is transferred via the SDAx and SCLx pins.

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

Remark m = 0 to 3
n = 0, 2 to 6
x = 0, 1

(k) A/D converter

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

(l) DMA controller

A six-channel DMA controller is incorporated. 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) 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
Port 1	8-bit I/O		Serial interface, timer I/O
Port 2	8-bit I/O		Serial interface, timer I/O, IEBus data I/O
Port 3	8-bit I/O		Timer I/O, external address bus, external interrupt
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		Timer I/O, key return input, external address bus
Port 11	8-bit I/O		Wait control, external address bus
Port 12	8-bit I/O		Serial interface, timer output
Port 13	4-bit I/O		–
Port 14	8-bit I/O		Serial interface, timer input
Port 15	8-bit I/O		Serial interface, timer I/O
Port 17	7-bit I/O		V_{DD0} = 4.5 V monitor output

(n) IEBus controller

The IEBus controller is a small digital data transmission system for transferring data between units and is incorporated in the V850/SC2 only.

1.4 V850/SC3

1.4.1 Features (V850/SC3)

○	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 16 MHz) (Able to execute instructions in parallel continuously without creating any register hazards) Saturated 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 expansion: 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 multiplexed) 3 V to 5 V interface enabled Bus hold function External wait function
○	Internal memory	μPD703088Y, 703089Y (mask ROM: 512 KB/RAM: 24 KB) μPD70F3089Y (flash memory: 512 KB/RAM: 24 KB)
○	Interrupts and exceptions	Non-maskable interrupts: 2 sources Maskable interrupts: 53 sources (μPD703088Y) 56 sources (μPD703089Y, 70F3089Y) Software exceptions: 32 sources Exception trap: 1 source
○	I/O lines	Total: 124 (12 input ports and 112 I/O ports) 3 V to 5 V interface enabled
○	Timer/counter	16-bit timer: 8 channels (TM0, TM1, TM7 to TM12) 16-bit timer: 2 channels (TM5, TM6)
○	Watch timer	When operating on 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 interfaces (SIO)	Asynchronous serial interface (UART)	
	3-wire serial I/O (CSI)	
	I^2C bus interface (I^2C)	
	8- to 16-bit variable-length serial interface	
	CSI (8-bit)/UART:	1 channel
	CSI (8- to 16-bit variable)/UART:	1 channel
	CSI (8-bit)/I^2C:	2 channels
	CSI (8- or 16-bit):	2 channels
	UART:	2 channels
	Dedicated baud rate generator:	5 channels
○ A/D converter	10-bit resolution: 12 channels	
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels	
○ ROM correction	4 correction addresses specifiable	
○ Regulator	3.5 V to 5.5 V input → internal 3.3 V	
○ Key return function	4 to 8 pins selectable, falling edge fixed	
○ Clock generator	During main clock or subclock operation	
	5-level CPU clock (including sub operations)	
○ Power save functions	HALT/IDLE/STOP modes	
○ FCAN controller	2 channels (μPD703089Y, 70F3089Y)	
	1 channel (μPD703088Y)	
○ Package	144-pin plastic LQFP (20 × 20 mm)	
○ CMOS structure	All static circuits	

1.4.2 Application fields (V850/SC3)

Car audio equipment

★ 1.4.3 Ordering information (V850/SC3)

Part Number	Package	Internal ROM
μPD703088YGJ-×××-UEN[Note]	144-pin plastic LQFP (20 × 20)	Mask ROM
μPD703089YGJ-×××-UEN[Note]	144-pin plastic LQFP (20 × 20)	Mask ROM
μPD70F3089YGJ-UEN[Note]	144-pin plastic LQFP (20 × 20)	Flash memory

Note Under development

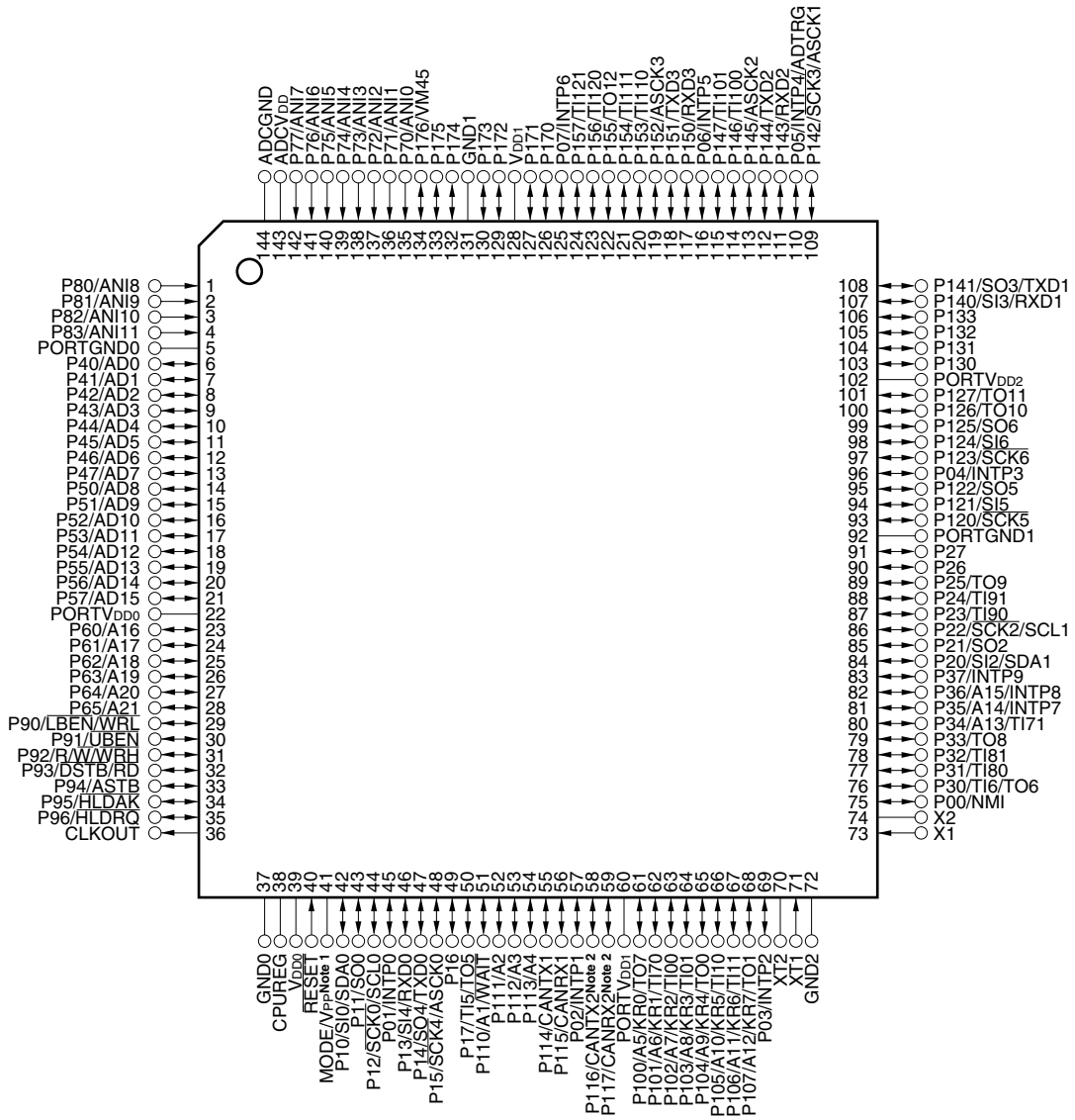
Remarks 1. ××× indicates ROM code suffix.

2. ROMless devices are not provided.

1.4.4 Pin configuration (top view) (V850/SC3)

144-pin plastic LQFP (20 × 20)

- μPD703088YGJ-×××-UEN
- μPD703089YGJ-×××-UEN
- μPD70F3089YGJ-UEN



Notes 1. μPD703088Y, 703089Y: MODE

μPD70F3089Y: VPP (connect to either GND0, GND1, or GND2 in normal operating mode)

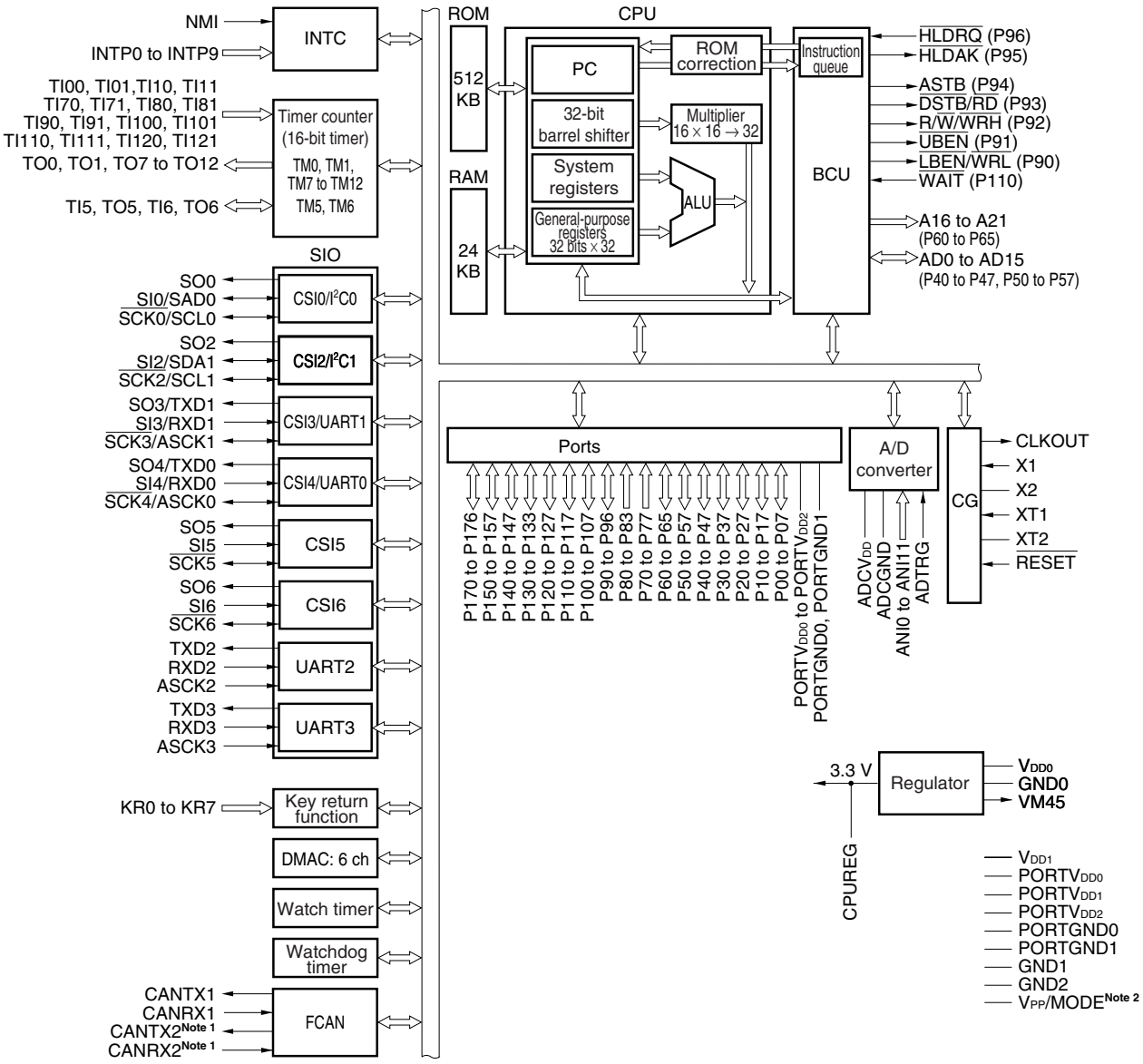
2. CANTX2 and CANRX2 are available only for the μPD703089Y and 70F3089Y.

Pin names (V850/SC3)

A16 to A21:	Address bus
AD0 to AD15:	Address/data bus
ADCGND:	Ground for analog
$ADCV_{DD}$	Power supply for analog
ADTRG:	A/D trigger input
ANI0 to ANI11:	Analog input
ASCK0 to ASCK3:	Asynchronous serial clock
ASTB:	Address strobe
CANRX1, CANRX2:	FCAN receive data
CANTX1, CANTX2:	FCAN transmit data
CLKOUT:	Clock output
CPUREG:	Regulator control
$\overline{DSTB}$:	Data strobe
GND0 to GND2:	Ground
$\overline{HLDAK}$:	Hold acknowledge
$\overline{HLDRQ}$:	Hold request
INTP0 to INTP9:	Interrupt request from peripherals
KR0 to KR7:	Key return
$\overline{LBEN}$:	Lower byte enable
MODE:	Mode
NMI:	Non-maskable interrupt request
PORTGND0, PORTGND1:	Ground for ports
$PORTV_{DD0}$ to $PORTV_{DD2}$:	Power supply for ports
P00 to P07:	Port 0
P10 to P17:	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 P117:	Port 11
P120 to P127:	Port 12
P130 to P133:	Port 13
P140 to P147:	Port 14
P150 to P157:	Port 15
P170 to P176:	Port 17
$\overline{RESET}$:	Reset
R/$\overline{W}$:	Read/write status
RXD0 to RXD3:	Receive data
$\overline{SCK0}$, $\overline{SCK2}$ to $\overline{SCK6}$:	Serial clock
SCL0, SCL1:	Serial clock
SDA0, SDA1:	Serial data
SI0, SI2 to SI6:	Serial input
SO0, SO2 to SO6:	Serial output
TI00, TI01, TI10, TI11, TI100, TI101, TI110, TI111, TI120, TI121, TI5, TI6, TI70, TI71, TI80, TI81, TI90, TI91:	Timer input
TO0, TO1, TO5 to TO12:	Timer output
TXD0 to TXD3:	Transmit data
$\overline{UBEN}$:	Upper byte enable
V_{DD0}, V_{DD1}:	Power supply
VM45:	V_{DD} = 4.5 V monitor output
V_{PP}:	Programming power supply
$\overline{WAIT}$:	Wait
X1, X2:	Crystal for main clock
XT1, XT2:	Crystal for sub-clock

1.4.5 Function blocks (V850/SC3)

(1) Internal block diagram



Notes 1. μPD703089Y, 70F3089Y only

2. μPD703088Y, 703089Y: MODE

μPD70F3089Y: V_{PP} (connect to either GND0, GND1, or GND2 in normal operating mode)

(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 a multiplier (16 bits × 16 bits → 32 bits) and a 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 512 KB mask ROM or flash memory mapped to the address space starting at 00000000H.

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

(d) RAM

This consists of a 24 KB RAM mapped to the address space starting 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 INTP9) 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: one for the main clock (f_{XX}) and one for the 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 ten-channel 16-bit timer/event counter is incorporated, enabling measurement of pulse intervals and frequency as well as programmable pulse output.

(h) Watch timer

This timer counts the reference time period (0.5 second) for counting the clock (the 32.768 kHz subclock or the 8.388 MHz 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 provided to detect inadvertent program loops and 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 interfaces (SIO)

The V850/SC3 includes four kinds of serial interfaces: an asynchronous serial interface (UARTm), a 3-wire serial I/O (CSIn), and an I^2C bus interface (I^2Cx), which can use up to eight channels at the same time. Two of these channels are switchable between the UART and CSI and another two are switchable between CSI and I^2C.

For UARTm, data is transferred via the TXDm and RXDm pins.

For CSIn, data is transferred via the SOn, SIn, and $\overline{SCKn}$ pins.

For I^2Cx, data is transferred via the SDAx and SCLx pins.

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

Remark m = 0 to 3
n = 0, 2 to 6
x = 0, 1

(k) A/D converter

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

(l) DMA controller

A six-channel DMA controller is incorporated. 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) 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
Port 1	8-bit I/O		Serial interface, timer I/O
Port 2	8-bit I/O		Serial interface, timer I/O
Port 3	8-bit I/O		Timer I/O, external address bus, external interrupt
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		Timer I/O, key return input, external address bus
Port 11	8-bit I/O		Wait control, FCAN data I/O, external address bus
Port 12	8-bit I/O		Serial interface, timer output
Port 13	4-bit I/O		–
Port 14	8-bit I/O		Serial interface, timer input
Port 15	8-bit I/O		Serial interface, timer I/O
Port 17	7-bit I/O		V_{DD0} = 4.5 V monitor output

(n) FCAN controller

The FCAN controller is a small digital data transmission system for transferring data between units.

A two-channel FCAN controller is incorporated in the μPD703089Y and 70F3089Y (FCAN1, FCAN2).

A one-channel FCAN controller is incorporated in the μPD703088Y (FCAN1).

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The names and functions of the pins in the V850/SC1, V850/SC2, and V850/SC3 are described below, divided into port pins and non-port pins.

There are six types of pin I/O buffer power supplies: $PORTV_{DD0}$ to $PORTV_{DD2}$, V_{DD0}, V_{DD1}, and $ADCV_{DD}$. The relationship between these power supplies and the pins is described below.

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

(a) μPD70F3089Y

Power Supply	Corresponding Pins	Usable Voltage Range
$PORTV_{DD0}$ Note 1	P40 to P47, P50 to P57, P60 to P65, P90 to P96, CLKOUT	3.0 V ≤ $PORTV_{DD0}$ ≤ 5.5 V
$PORTV_{DD1}$ Note 1	P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117	3.0 V ≤ $PORTV_{DD1}$ ≤ 5.5 V Note 2
$PORTV_{DD2}$ Note 1	P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157	3.0 V ≤ $PORTV_{DD2}$ ≤ 5.5 V Note 2
V_{DD0}	$\overline{RESET}$	When A/D converter not used: 4.0 V ≤ V_{DD0} ≤ 5.5 V When A/D converter used: 4.5 V ≤ V_{DD0} = $ADCV_{DD}$ ≤ 5.5 V
V_{DD1}	P170 to P176	4.0 V ≤ V_{DD1} ≤ 5.5 V
$ADCV_{DD}$	P70 to P77, P80 to P83	When A/D converter not used: 4.0 V ≤ $ADCV_{DD}$ ≤ 5.5 V When A/D converter used: 4.5 V ≤ V_{DD0} = $ADCV_{DD}$ ≤ 5.5 V

Notes 1. The electrical specifications differ between an operating frequency of 4 to 17 MHz and an operating frequency of 4 to 20 MHz.

2. When the FCAN controller is used: $PORTV_{DD1}$ ≤ $PORTV_{DD2}$
(Due to the supply voltage conditions of the in-circuit emulator)

Caution The conditions for the power supplies are as follows.

$PORTV_{DD0}$ ≤ $PORTV_{DD1}$ ≤ $PORTV_{DD2}$ ≤ V_{DD0} = V_{DD1} = $ADCV_{DD}$

(b) Other than μPD70F3089Y

Power Supply	Corresponding Pins	Usable Voltage Range
$PORTV_{DD0}$[Note 1]	P40 to P47, P50 to P57, P60 to P65, P90 to P96, CLKOUT	$3.0\ V \le PORTV_{DD0} \le 5.5\ V$
$PORTV_{DD1}$[Note 1]	P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117	$3.0\ V \le PORTV_{DD1} \le 5.5\ V$[Note 2]
$PORTV_{DD2}$[Note 1]	P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157	$3.0\ V \le PORTV_{DD2} \le 5.5\ V$[Note 2]
V_{DD0}	$\overline{RESET}$	When A/D converter not used: $3.5\ V \le V_{DD0} \le 5.5\ V$ When A/D converter used: $4.5\ V \le V_{DD0} = ADCV_{DD} \le 5.5\ V$
V_{DD1}	P170 to P176	$3.5\ V \le V_{DD1} \le 5.5\ V$
$ADCV_{DD}$	P70 to P77, P80 to P83	When A/D converter not used: $3.5\ V \le ADCV_{DD} \le 5.5\ V$ When A/D converter used: $4.5\ V \le V_{DD0} = ADCV_{DD} \le 5.5\ V$

Notes 1. The electrical specifications differ between an operating frequency of 4 to 17 MHz and an operating frequency of 4 to 20 MHz.

2. When the FCAN controller is used: $PORTV_{DD1} \le PORTV_{DD2}$
(Due to the supply voltage conditions of the in-circuit emulator)

Caution The conditions for the power supplies are as follows.

$$\mathbf{PORTV_{DD0} \le PORTV_{DD1} \le PORTV_{DD2} \le V_{DD0} = V_{DD1} = ADCV_{DD}}$$

The differences in the pins of the V850/SC1, V850/SC2, and V850/SC3 are shown below.

Table 2-2. Differences in Pins of V850/SC1, V850/SC2, and V850/SC3

Pin	V850/SC1		V850/SC2		V850/SC3		
	μPD703068Y	μPD70F3089Y	μPD703069Y	μPD70F3089Y	μPD703088Y	μPD703089Y	μPD70F3089Y
V_{PP}	Not available	Available	Not available	Available	Not available		Available
A1 to A5	Available				Not available		
$\overline{IETX0}$	Not available		Available		Not available		
$\overline{IERX0}$	Not available		Available		Not available		
CANTX1, CANRX1	Not available				Available		
CANTX2, CANRX2	Not available				Not available	Available	

(1) Port pins

(1/4)

Pin Name	I/O	PULL	Function	Alternate Function
P00	I/O	No	Port 0 8-bit I/O port Input/output can be specified in 1-bit units.	NMI
P01				INTP0
P02				INTP1
P03				INTP2
P04				INTP3
P05				INTP4/ADTRG
P06				INTP5
P07				INTP6
P10	I/O	No	Port 1 8-bit I/O port Input/output can be specified in 1-bit units. Only P10 and P12 can be specified as N-ch open-drain pins.	SI0/SDA0
P11				SO0
P12				$\overline{\text{SCK0}}$/SCL0
P13				SI4/RXD0
P14				SO4/TXD0
P15				$\overline{\text{SCK4}}$/ASCK0
P16				–
P17				TI5/TO5
P20	I/O	No	Port 2 8-bit I/O port Input/output can be specified in 1-bit units. Only P20 and P22 can be specified as N-ch open-drain pins.	SI2/SDA1
P21				SO2
P22				$\overline{\text{SCK2}}$/SCL1
P23				TI90
P24				TI91
P25				TO9
P26				$\overline{\text{IERX0}}$[Note 1]
P27				$\overline{\text{IETX0}}$[Note 1]
P30	I/O	No	Port 3 8-bit I/O port Input/output can be specified in 1-bit units.	TI6/TO6
P31				TI80
P32				TI81
P33				TO8
P34				TI71, A13[Note 2]
P35				INTP7/A14[Note 2]
P36				INTP8/A15[Note 2]
P37				INTP9

Notes 1. Only for the V850/SC2

2. Only for the V850/SC1 and V850/SC2

Remark PULL: On-chip pull-up resistor

(2/4)

Pin Name	I/O	PULL	Function	Alternate Function
P40	I/O	No	Port 4 8-bit I/O port Input/output 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 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 can be specified in 1-bit units.	A16
P61				A17
P62				A18
P63				A19
P64				A20
P65				A21
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

Remark PULL: On-chip pull-up resistor

(3/4)

Pin Name	I/O	PULL	Function	Alternate Function
P90	I/O	No	Port 9 7-bit I/O port Input/output can be specified in 1-bit units.	$\overline{LBEN}/\overline{WRL}$ Note 1
P91				$\overline{UBEN}$
P92				R/$\overline{W}$/$\overline{WRH}$ Note 1
P93				$\overline{DSTB}/\overline{RD}$ Note 1
P94				ASTB
P95				$\overline{HLDAK}$
P96				$\overline{HLDRQ}$
P100	I/O	Yes	Port 10 8-bit I/O port Input/output can be specified in 1-bit units.	KR0/TO7/A5 Note 1
P101				KR1/TI70/A6 Note 1
P102				KR2/TI00/A7 Note 1
P103				KR3/TI01/A8 Note 1
P104				KR4/TO0/A9 Note 1
P105				KR5/TI10/A10 Note 1
P106				KR6/TI11/A11 Note 1
P107				KR7/TO1/A12 Note 1
P110	I/O	No	Port 11 8-bit I/O port Input/output can be specified in 1-bit units.	$\overline{WAIT}$/A1 Note 1
P111				A2 Note 1
P112				A3 Note 1
P113				A4 Note 1
P114				CANTX1 Note 2
P115				CANRX1 Note 2
P116				CANTX2 Note 3
P117				CANRX2 Note 3
P120	I/O	No	Port 12 8-bit I/O port Input/output can be specified in 1-bit units.	$\overline{SCK5}$
P121				SI5
P122				SO5
P123				$\overline{SCK6}$
P124				SI6
P125				SO6
P126				TO10
P127				TO11
P130	I/O	No	Port 13 4-bit I/O port Input/output can be specified in 1-bit units.	–
P131				–
P132				–
P133				–

Notes 1. Only for the V850/SC1 and V850/SC2

2. Only for the V850/SC3

3. Only for the μPD703089Y, 70F3089Y

Remark PULL: On-chip pull-up resistor

(4/4)

Pin Name	I/O	PULL	Function	Alternate Function
P140	I/O	No	Port 14 8-bit I/O port Input/output can be specified in 1-bit units.	SI3/RXD1
P141				SO3/TXD1
P142				$\overline{\text{SCK3}}$/ASCK1
P143				RXD2
P144				TXD2
P145				ASCK2
P146				TI100
P147				TI101
P150	I/O	Yes	Port 15 8-bit I/O port Input/output can be specified in 1-bit units.	RXD3
P151				TXD3
P152				ASCK3
P153				TI110
P154				TI111
P155				TO12
P156				TI120
P157				TI121
P170	I/O	No	Port 17 7-bit I/O port Input/output can be specified in 1-bit units.	–
P171				–
P172				–
P173				–
P174				–
P175				–
P176				VM45

Remark PULL: On-chip pull-up resistor

(2) Non-port pins

(1/4)

Pin Name	I/O	PULL	Function	Alternate Function
A1	Output	No	Lower address bus used for external memory expansion (V850/SC1 and V850/SC2 only)	P110/$\overline{\text{WAIT}}$
A2 to A4				P111 to P113
A5		Yes		P100/KR0/TO7
A6				P101/KR1/TI70
A7				P102/KR2/TI00
A8				P103/KR3/TI01
A9				P104/KR4/TO0
A10				P105/KR5/TI10
A11				P106/KR6/TI11
A12				P107/KR7/TO1
A13		No		P34/TI71
A14				P35/INTP7
A15				P36/INTP8
A16 to A21	Output	No	Higher address bus used for external memory expansion	P60 to P65
AD0 to AD15	I/O	No	16-bit multiplexed address/data bus used for external memory expansion	P40 to P47, P50 to P57
ADCGND	–	–	Ground potential for A/D converter	–
$ADCV_{DD}$	–	–	Power supply pin and reference voltage pin for A/D converter	–
ADTRG	Input	No	A/D converter external trigger input	P05/INTP4
ANI0 to ANI11	Input	No	Analog input to A/D converter	P70 to P77, P80 to P83
ASCK0	Input	No	Baud rate clock input for UART0	P15/$\overline{\text{SCK4}}$
ASCK1			Baud rate clock input for UART1	P142/$\overline{\text{SCK3}}$
ASCK2			Baud rate clock input for UART2	P145
ASCK3			Baud rate clock input for UART3	P152
ASTB	Output	No	External address strobe signal output	P94
CANRX1	Input	No	CAN1 receive data input[Note 1]	P115
CANRX2			CAN2 receive data input[Note 2]	P117
CANTX1	Output		CAN1 transmit data output[Note 1]	P114
CANTX2			CAN2 transmit data output[Note 2]	P116
CLKOUT	Output	–	Internal system clock output	–
CPUREG	–	–	Connection of regulator output stabilization capacitance	–
$\overline{\text{DSTB}}$	Output	No	External data strobe signal output	P93/$\overline{\text{RD}}$[Note 3]
GND0 to GND2	–	–	Ground potential	–

Notes 1. Only for the V850/SC3

2. Only for the μPD703089Y and 70F3089Y

3. Only for the V850/SC1 and V850/SC2

Remark PULL: On-chip pull-up resistor

(2/4)

Pin Name	I/O	PULL	Function	Alternate Function
$\overline{\text{HLDAK}}$	Output	No	Bus hold acknowledge output	P95
$\overline{\text{HLDRQ}}$	Input		Bus hold request input	P96
$\overline{\text{IERX0}}$	Input	No	IEBus data input (V850/SC2 only)	P26
$\overline{\text{IETX0}}$	Output		IEBus data output (V850/SC2 only)	P27
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
INTP6			External interrupt request input (digital noise elimination for remote control)	P07
INTP7			External interrupt request input (analog noise elimination)	P35/A14[Note 1]
INTP8				P36/A15[Note 1]
INTP9				P37
KR0	Input	Yes	Key return input	P100/A5[Note 1]/TO7
KR1				P101/A6[Note 1]/TI70
KR2				P102/A7[Note 1]/TI00
KR3				P103/A8[Note 1]/TI01
KR4				P104/A9[Note 1]/TO0
KR5				P105/A10[Note 1]/TI10
KR6				P106/A11[Note 1]/TI11
KR7				P107/A12[Note 1]/T01
$\overline{\text{LBEN}}$	Output	No	External data bus's lower byte enable signal output	P90/$\overline{\text{WRL}}$[Note 1]
MODE	–	–	Specifies operation mode (other than μPD78F38089Y)	V_{PP}[Note 2]
NMI	Input	No	Non-maskable interrupt request input (analog noise elimination)	P00
PORTGND0	–	–	Ground potential for ports	–
PORTGND1				–
$PORTV_{DD0}$	–	–	Positive power supply for ports (P40 to P47, P50 to P57, P60 to P65, P90 to P96, CLKOUT)	–
$PORTV_{DD1}$			Positive power supply for ports (P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117)	–
$PORTV_{DD2}$			Positive power supply for ports (P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157)	–
R/$\overline{\text{W}}$	Output	No	External read/write status output	P92/$\overline{\text{WRH}}$[Note 1]
$\overline{\text{RD}}$	Output	No	Read strobe signal output (V850/SC1 and V850/SC2 only)	P93/$\overline{\text{DSTB}}$
$\overline{\text{RESET}}$	Input	–	System reset input	–
RXD0	Input	No	Serial receive data input for UART0, UART1, UART2, UART3	P13/SI4
RXD1				P140/SI3
RXD2				P143
RXD3				P150
$\overline{\text{SCK0}}$	I/O	No	Serial clock I/O (3-wire type) for CSI0, CSI2, CSI3	P12/SCL0
$\overline{\text{SCK2}}$				P22/SCL1
$\overline{\text{SCK3}}$				P142/ASCK1

Notes 1. Only for the V850/SC1 and V850/SC2

2. Only for the μPD70F3089Y

Remark PULL: On-chip pull-up resistor

(3/4)

Pin Name	I/O	PULL	Function	Alternate Function
$\overline{SCK4}$	I/O	No	Serial clock I/O (3-wire type) for variable-length CSI4	P15/ASCK0
$\overline{SCK5}$			Serial clock I/O (3-wire type) for CSI5, CSI6	P120
$\overline{SCK6}$				P123
SCL0	I/O	No	Serial clock I/O for I^2C0	P12/$\overline{SCK0}$
SCL1			Serial clock I/O for I^2C1	P22/$\overline{SCK2}$
SDA0	I/O	No	Serial transmit/receive data input for I^2C0	P10/SI0
SDA1			Serial transmit/receive data input for I^2C1	P20/SI2
SI0	Input	No	Serial receive data input (3-wire type) for CSI0, CSI2, CSI3	P10/SDA0
SI2				P20/SDA1
SI3				P140/RXD1
SI4	Input	No	Serial receive data input (3-wire type) for variable-length CSI4	P13/RXD0
SI5			Serial receive data input (3-wire type) for CSI5, CSI6	P121
SI6				P124
SO0	Output	No	Serial transmit data output (3-wire type) for CSI0, CSI2, CSI3	P11
SO2				P21
SO3				P141/TXD1
SO4	Output	No	Serial transmit data output (3-wire type) for variable-length CSI4	P14/TXD0
SO5		No	Serial transmit data output (3-wire type) for CSI5, CSI6	P122
SO6		No		P125
TI00	Input	Yes	External count clock input for TM0/ external capture trigger input for TM0	P102/A7[Note]/KR2
TI01			External capture trigger input for TM0	P103/A8[Note]/KR3
TI10			External count clock input for TM1/ external capture trigger input for TM1	P105/A10[Note]/KR5
TI11			External capture trigger input for TM1	P106/A11[Note]/KR6
TI100		No	External count clock input for TM10/ external capture trigger input for TM10	P146
TI101			External capture trigger input for TM10	P147
TI110		No	External count clock input for TM11/ external capture trigger input for TM11	P153
TI111			External capture trigger input for TM11	P154
TI120	Input	No	External count clock input for TM12/ external capture trigger input for TM12	P156
TI121			External capture trigger input for TM12	P157
TI5			External count clock input for TM5, TM6	P17/TO5
TI6				P30/TO6

Note Only for the V850/SC1 and V850/SC2

Remark PULL: On-chip pull-up resistor

(4/4)

Pin Name	I/O	PULL	Function	Alternate Function
TI70	Input	Yes	External count clock input for TM7/ external capture trigger input for TM7	P101/A6[Note 1]/KR1
TI71		No	External capture trigger input for TM7	P34/A13[Note 1]
TI80			External count clock input for TM8/ external capture trigger input for TM8	P31
TI81			External capture trigger input for TM8	P32
TI90		No	External count clock input for TM9/ external capture trigger input for TM9	P23
TI91			External capture trigger input for TM9	P24
TO0	Output	Yes	Pulse signal output for TM0, TM1, TM10 to TM12, TM5	P104/A9[Note 1]/KR4
TO1				P107/A12[Note 1]/KR7
TO10		No		P126
TO11				P127
TO12				P155
TO5				P17/TI5
TO6		No	Pulse signal output for TM6 to TM9	P30/TI6
TO7		Yes		P100/A5[Note 1]/KR0
TO8		No		P33
TO9				P25
TXD0	Output	No	Serial transmit data output for UART0, UART1, UART2, UART3	P14/SO4
TXD1				P141/SO3
TXD2				P144
TXD3				P151
$\overline{\text{UBEN}}$	Output	No	Higher byte enable signal output for external data bus	P91
V_{DD0}	–	–	Positive power supply pin ($\overline{\text{RESET}}$)	–
V_{DD1}			Positive power supply pin (P170 to P176)	–
VM45	Output	No	V_{DD0} = 4.5 V monitor output	P176
V_{PP}	–	–	High-voltage application pin for program write/verify (μPD70F3089Y only)	MODE[Note 2]
$\overline{\text{WAIT}}$	Input	No	Control signal input for inserting wait in bus cycle	P110/A1[Note 1]
$\overline{\text{WRH}}$	Output	No	Higher byte write strobe signal output for external data bus (V850/SC1, V850/SC2 only)	P92/R/$\overline{\text{W}}$
$\overline{\text{WRL}}$			Lower byte write strobe signal output for external data bus (V850/SC1, V850/SC2 only)	P90/$\overline{\text{LBEN}}$
X1	Input	–	Resonator connection for main clock	–
X2	–			–
XT1	Input	–	Resonator connection for subclock	–
XT2	–			–

Notes 1. Only for the V850/SC1 and V850/SC2

2. Other than μPD70F3089Y

Remark PULL: On-chip pull-up resistor

2.2 Pin States

The operation states of pins in various operating modes are described below.

★ **Table 2-3. Pin Operation States in Various Operating Modes**

Operating Mode / Pin	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 during previously set external bus cycle is held
L: Low-level output
H: High-level output
–: Input without sampling

2.3 Description of Pin Functions

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

P00 to P07 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as an NMI input, external interrupt request inputs, and the external trigger for the A/D converter. The pin's valid edge is 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).

(2) P10 to P17 (port 1) ··· 3-state I/O

P10 to P17 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as I/O pins for the serial interface and timer/counter.

P10 and P12 can be selected as normal output or N-ch open-drain output pins.

(a) Port function

P10 to P17 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(b) Alternate functions

(i) SI0, SI4 (serial input 0, 4) ··· Input

These are the serial receive data input pins for CSI0 and CSI4.

(ii) SO0, SO4 (serial output 0, 4) ··· Output

These are the serial transmit data output pins for CSI0 and CSI4.

(iii) $\overline{\text{SCK0}}$, $\overline{\text{SCK4}}$ (serial clock 0, 4) ··· 3-state I/O

These are the serial clock I/O pins for CSI0 and CSI4.

(iv) SDA0 (serial data 0) ··· I/O

This is the serial transmit/receive data I/O pin for I^2C0.

(v) SCL0 (serial clock 0) ··· I/O

This is the serial clock I/O pin for I^2C0.

(vi) RXD0 (receive data 0) ··· Input

This is the serial receive data input pin for UART0.

(vii) TXD0 (transmit data 0) ··· Output

This is the serial transmit data output pin for UART0.

(viii) ASCK0 (asynchronous serial clock 0) ··· Input

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

(ix) TI5 (timer input 5) ··· Input

This is the external count clock input pin for timer 5.

(x) TO5 (timer output 5) ··· Output

This is the pulse signal output pin for timer 5.

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

P20 to P27 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as I/O pins for the serial interface, timer/counter, and IEBus data.

(a) Port function

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

(b) Alternate functions

(i) SI2 (serial input 2) ··· Input

This is the serial receive data input pin for CSI2.

(ii) SO2 (serial output 2) ··· Output

This is the serial transmit data output pin for CSI2.

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

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

(iv) SDA1 (serial data 1) ... Input

This is the serial transmit/receive data I/O pin for I^2C1.

(v) SCL1 (serial clock 1) ... I/O

This is the serial clock I/O pin for I^2C1.

(vi) TI90 (timer input 90) ... Input

This is the external count clock input and external capture trigger input pin for timer 9.

(vii) TI91 (timer input 91) ... Input

This is the external capture trigger input pin for timer 9.

(viii) TO9 (timer output 9) ... Output

This is the pulse signal output pin for timer 9.

(ix) $\overline{\text{IERX0}}$ (IEBus receive data) ... Input

This is the IEBus data input signal.

$\overline{\text{IERX0}}$ is available only for the V850/SC2.

(x) $\overline{\text{IETX0}}$ (IEBus transmit data) ... Output

This is the IEBus data output signal.

$\overline{\text{IETX0}}$ is available only for the V850/SC2.

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

P30 to P37 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as I/O pins for the timer/counter, an address bus (A13 to A15) for external memory expansion, and external interrupt request inputs.

(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) TI6 (timer input 6) ··· Input

This is the external count clock input pin for timer 6.

(ii) TI71 (timer input 71) ··· Input

This is the external capture trigger input pin for timer 7.

(iii) TI80 (timer input 80) ··· Input

This is the external count clock input and external capture trigger input pin for timer 8.

(iv) TI81 (timer input 81) ··· Input

This is the external capture trigger input pin for timer 8.

(v) TO6, TO8 (timer output 6, 8) ··· Output

These are the pulse signal output pins for timer 6 and timer 8.

(vi) A13 to A15 (address 13 to 15) ··· Output

These pins comprise an address bus for external access and operate as the A13 to A15 (22-bit address) output pins. 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, these pins hold the address of the bus cycle immediately before.

A13 to A15 are available only for the V850/SC1 and V850/SC2.

(vii) INTP7 to INTP9 (interrupt request from peripherals) ··· Input

These are external interrupt request input pins.

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

P40 to P47 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as a time division address/data bus (AD0 to AD7) when memory is expanded externally.

(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 function (external expansion mode)

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

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

These pins comprise a multiplexed address/data bus for external access. At the address timing (T1 state), these pins operate as the AD0 to AD7 (22-bit address) output pins. At the data timing (T2, TW, T3), they operate as 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 in 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 toP57 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as a time division address/data bus (AD8 to AD15) when memory is expanded externally.

(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 function (external expansion mode)

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

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

These pins comprise a multiplexed address/data bus for external access. At the address timing (T1 state), these pins operate as the AD8 to AD15 (22-bit address) output pins. At the data timing (T2, TW, T3), they operate as 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 of 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 function as a 6-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as an address bus (A16 to A21) when memory is expanded externally. During 8-bit access of port 6, the higher two bits are ignored during a write operation and are read as "00" during a read operation.

(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 function (external expansion mode)

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

(i) A16 to A21 (address 16 to 21) ··· Output

These pins comprise an address bus 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, these pins hold the address of the bus cycle immediately before.

(8) P70 to P77 (port 7), P80 to P83 (port 8) ··· Input

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

In addition to input ports, these pins can also be used as analog input pins of the A/D converter for the alternate function. However, they cannot be switched between input ports and analog input pins.

(a) Port function

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

(b) Alternate function

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

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

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

Connect a capacitor between $ADCV_{DD}$ and ADCGND to prevent noise-related operation faults. Also, do not apply voltage that is outside the range for $ADCV_{DD}$ and ADCGND to pins that are being used as inputs for the A/D converter. If it is possible for noise above the $ADCV_{DD}$ range or below the ADCGND range 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 function as a 7-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used 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 “0” during a read operation.

(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 function (external expansion mode)

P90 to P96 can be specified 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 the lower byte enable signal output pin for the external 16-bit data bus. During byte access to the odd-numbered addresses, these pins are set to 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 to inactive, this pin holds the address of the bus cycle immediately before.

(ii) $\overline{\text{UBEN}}$ (upper byte enable) ··· Output

This is the upper byte enable signal output pin for the external 16-bit data bus. During byte access of even-numbered addresses, these pins are set to 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 to inactive, this pin holds the address of the bus cycle immediately before.

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 the output pin for the status signal that indicates whether the bus cycle is a read cycle or a write cycle during external access. High level is set during the read cycle and low level is set during the 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 to inactive.

(iv) $\overline{\text{DSTB}}$ (data strobe) ··· Output

This is the output pin for the access strobe signal for the external data bus. 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 to inactive.

(v) ASTB (address strobe) ··· Output

This is the output pin for the latch strobe signal for the external address bus. 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 to inactive.

(vi) $\overline{\text{HLDAK}}$ (hold acknowledge) ··· Output

This is the output pin for the acknowledge signal that indicates the high impedance status for the address bus, data bus, and control bus when the V850/SC1, V850/SC2, and V850/SC3 receive a bus hold request.

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

(vii) $\overline{\text{HLDRQ}}$ (hold request) ··· Input

This is the input pin by which an external device requests the V850/SC1, V850/SC2, and V850/SC3 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. This occurs either when the V850/SC1, V850/SC2, and V850/SC3 complete execution of the current bus cycle, or immediately if no bus cycle is being executed. The $\overline{\text{HLDAK}}$ signal is then set to active and the bus is released.

(viii) $\overline{\text{WRL}}$ (write strobe low-level data) ··· Output

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

This is output in the same write cycle as $\overline{\text{DSTB}}$.

$\overline{\text{WRL}}$ is available only for the V850/SC1 and V850/SC2.

(ix) $\overline{\text{WRH}}$ (write strobe high-level data) ··· Output

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

This is output in the same write cycle as $\overline{\text{DSTB}}$.

$\overline{\text{WRH}}$ is available only for the V850/SC1 and V850/SC2.

(x) $\overline{\text{RD}}$ (read) ··· Output

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

This is output in the same read cycle as $\overline{\text{DSTB}}$.

$\overline{\text{RD}}$ is available only for the V850/SC1 and V850/SC2.

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

P100 to P107 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as timer/counter I/O pins, key return inputs, and an address bus (A5 to A12) for external memory expansion.

(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) KR0 to KR7 (key return 0 to 7) ... Input

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

(ii) TI00, TI10, TI70 (timer input 00, 10, 70) ... Input

These are external count clock input and external capture trigger input pins for timers 0, 1, and 7.

(iii) TI01, TI11 (timer input 01, 11) ... Input

These are external capture trigger input pins for timers 0 and 1.

(iv) TO0, TO1, TO7 (timer output 0, 1, 7) ... Output

These are pulse signal output pins for timers 0, 1, and 7.

(v) A5 to A12 (address 5 to 12) ... Output

These pins comprise an address bus for external access and operate as the A5 to A12 (22-bit address) output pins. 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, these pins hold the address of the bus cycle immediately before.

A5 to A12 are available only for the V850/SC1 and V850/SC2.

(11) P110 to P117 (port 11) ··· 3-state I/O

P110 to P117 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.
In addition to I/O port pins, these pins can also be used as FCAN data I/O pins, the control signal ($\overline{\text{WAIT}}$) that inserts waits into the bus cycle and an address data bus (A1 to A4) for external memory expansion.

(a) Port function

P110 to P117 can be set to input or output in 1-bit units using the port 11 mode register (PM11).

(b) Alternate functions

(i) $\overline{\text{WAIT}}$ (wait) ··· Input

This is the 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).

(ii) CANRX1, CANRX2 (CAN receive data 1, 2) ··· Input

These are data input signals for CAN1 and CAN2.
CANRX1 is available only for the V850/SC3.
CANRX2 is available only for the μPD703089Y and 70F3089Y.

(iii) CANTX1, CANTX2 (CAN transmit data 1, 2) ··· Output

These are data output signals for CAN1 and CAN2.
CANTX1 is available only for the V850/SC3.
CANTX2 is available only for the μPD703089Y and 70F3089Y.

(iv) A1 to A4 (address 1 to 4)

These pins comprise an address bus for external access and operate as the A1 to A4 (22-bit address) output pins. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets bus cycle to inactive, these pins hold the address of the bus cycle immediately before.
A1 to A4 are available only for the V850/SC1 and V850/SC2.

(12) P120 to P127 (port 12) ··· 3-state I/O

P120 to P127 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as I/O pins for the serial interface, and outputs for the timer/counter.

(a) Port function

P120 to P127 can be set to input or output in 1-bit units using the port 12 mode register (PM12).

(b) Alternate functions

(i) SI5, SI6 (serial input 5, 6) ··· Input

These are the serial receive data input pins for CSI5 and CSI6.

(ii) SO5, SO6 (serial output 5, 6) ··· Output

These are the serial transmit data output pins for CSI5 and CSI6.

(iii) $\overline{\text{SCK5}}$, $\overline{\text{SCK6}}$ (serial clock 5, 6) ··· 3-state I/O

These are the serial clock I/O pins for CSI5 and CSI6.

(iv) TO10, TO11 (timer output 10, 11) ··· Output

These are the pulse signal output pins for timers 10 and 11.

(13) P130 to P133 (port 13) ··· 3-state I/O

P130 to P133 function as a 4-bit I/O port in which input and output can be specified in 1-bit units.

These pins can only be used as I/O port pins.

(14) P140 to P147 (port 14) ··· 3-state I/O

P140 to P147 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as I/O pins for the serial interface, and inputs for the timer/counter.

(a) Port function

P140 to P147 can be set to input or output in 1-bit units using the port 14 mode register (PM14).

(b) Alternate functions

(i) SI3 (serial input 3) ··· Input

This is the serial receive data input pin for CSI3.

(ii) SO3 (serial output 3) ··· Output

This is the serial transmit data output pin for CSI3.

(iii) $\overline{\text{SCK3}}$ (serial clock 3) ··· 3-state I/O

This is the serial clock I/O pin for CSI3.

(iv) RXD1 (receive data 1) ··· Input

This is the serial receive data input pin for UART1.

(v) TXD1 (transmit data 1) ··· Output

This is the serial transmit data output pin for UART1.

(vi) ASCK1 (asynchronous serial clock 1) ... Input

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

(vii) TI100 (timer input 100) ··· Input

This is the external count clock input and external capture trigger input pin for timer 10.

(viii) TI101 (timer input 101) ··· Input

This is the external capture trigger input pin for timer 10.

(15) P150 to P157 (port 15) ··· 3-state I/O

P150 to P157 function as an 8-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as I/O pins for the serial interface and the timer/counter.

(a) Port function

P150 to P157 can be set to input or output in 1-bit units using the port 15 mode register (PM15).

(b) Alternate functions

(i) RXD3 (receive data 3) ... Input

This is the serial receive data input pin for UART3.

(ii) TXD3 (transmit data 3) ... Output

This is the serial transmit data output pin for UART3.

(iii) ASCK3 (asynchronous serial clock 3) ... Input

This is the serial baud rate clock input pin for UART3.

(iv) TI110 (timer input 110) ... Input

This is the external count clock input and external capture trigger input pin for timer 11.

(v) TI111 (timer input 111) ... Input

This is the external capture trigger input pin for timer 11.

(vi) TI120 (timer input 120) ... Input

This is the external count clock input and external capture trigger input pin for timer 12.

(vii) TI121 (timer input 121) ... Input

This is the external capture trigger input pin for timer 12.

(viii) TO12 (timer output 12) ... Output

This is the pulse signal output pin for timer 12.

(16) P170 to P176 (port 17) ··· 3-state I/O

P170 to P176 function as a 7-bit I/O port in which input and output can be specified in 1-bit units.

In addition to I/O port pins, these pins can also be used as V_{DD0} = 4.5 V monitor output. During 8-bit access of port 17, the highest bit is ignored during a write operation and is read as "0" during a read operation.

(a) Port function

P170 to P176 can be set to input or output in 1-bit units using the port 17 mode register (PM17).

(b) Alternate function

(i) VM45 (V_{DD0} = 4.5 V monitor output) … Output

This is the V_{DD0} = 4.5 V monitor output pin.

(17) $\overline{\text{RESET}}$ (reset) ··· Input

$\overline{\text{RESET}}$ is a signal that is input asynchronously and 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 signal can also be used to cancel a standby mode (HALT, IDLE, or STOP mode).

(18) Mode (mode)

This is pin used in other than the μPD78F3089.

Connect to GND0 to GND2 in normal operating mode.

(19) CLKOUT (clock output) ··· Output

This pin outputs internally generated bus clocks.

(20) X1, X2 (crystal)

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

(21) XT1, XT2 (crystal for sub-clock)

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

(22) $ADCV_{DD}$ (power supply for analog)

This is the analog power supply pin for the A/D converter and alternate-function ports.

This pin also functions as a reference voltage pin for the A/D converter.

(23) ADCGND (ground for analog)

This is the ground pin for the A/D converter and alternate-function ports.

(24) CPUREG (regulator control)

This is the regulator pin for the CPU power supply. Connect this pin to GND0 to GND2 via a capacitor of 1 μF (recommended value).

(25) $PORTV_{DD0}$ to $PORTV_{DD2}$ (power supply for port)

These are positive power supply pins for I/O ports and alternate-function pins.

(26) PORTGND0, PORTGND1 (ground for port)

These are ground pins for I/O ports and alternate-function pins (except for the alternate-function ports of the bus interface).

(27) V_{DD0}, V_{DD1} (power supply)

These are positive power supply pins. V_{DD0} and V_{DD1} pins should be connected to a positive power source.

(28) GND0 to GND2 (ground)

These are ground pins. All the GND0 to GND2 pins should be grounded.

(29) V_{PP} (programming power supply)

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

This pin is used in the μPD70F3089Y. Connect to either GND0, GND1, or GND2 in normal operating mode.

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

(1/3)

Pin	Alternate Function	I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection
P00	NMI	8	$PORTV_{DD1}$	Input: Independently connect to $PORTV_{DD1}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P01 to P03	INTP0 to INTP2			
P04	INTP3		$PORTV_{DD2}$	Input: Independently connect to $PORTV_{DD2}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P05	INTP4/ADTRG			
P06, P07	INTP5, INTP6			
P10	SI0/SDA0	10	$PORTV_{DD1}$	Input: Independently connect to $PORTV_{DD1}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P11	SO0	5		
P12	$\overline{SCK0}$/SCL0	10		
P13	SI4/RXD0	8		
P14	SO4/TXD0	5		
P15	$\overline{SCK4}$/ASCK0	8		
P16	–	5		
P17	TI5/TO5	8		
P20	SI2/SDA1	10	$PORTV_{DD2}$	Input: Independently connect to $PORTV_{DD2}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P21	SO2	5		
P22	$\overline{SCK2}$/SCL1	10		
P23, P24	TI90, TI91	8		
P25	TO9	5		
P26	$\overline{IERX0}$ Note 1	8		
P27	$\overline{IETX0}$ Note 1	5		
P30	TI6/TO6	8	$PORTV_{DD1}$	Input: Independently connect to $PORTV_{DD1}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P31, P32	TI80, TI81			
P33	TO8	5		
P34	TI71/A13 Note 2	8		
P35, P36	INTP7/A14 Note 2, INTP8/A15 Note 2			
P37	INTP9			
P40 to P47	AD0 to AD7	5	$PORTV_{DD0}$	Input: Independently connect to $PORTV_{DD0}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P50 to P57	AD8 to AD15		$PORTV_{DD0}$	
P60 to P65	A16 to A21		$PORTV_{DD0}$	

Notes 1. Only for the V850/SC2

2. Only for the V850/SC1 and V850/SC2

(2/3)

Pin	Alternate Function	I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection
P70 to P77	ANI0 to ANI7	9	$ADCV_{DD}$	Independently connect to $ADCV_{DD}$ or ADCGND via a resistor.
P80 to P83	ANI8 to ANI11	9	$ADCV_{DD}$	
P90	$\overline{LBEN}/\overline{WRL}$ Note 1	5	$PORTV_{DD0}$	Input: Independently connect to $PORTV_{DD0}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P91	$\overline{UBEN}$			
P92	$R/\overline{W}/\overline{WRH}$ Note 1			
P93	$\overline{DSTB}/\overline{RD}$ Note 1			
P94	ASTB			
P95	$\overline{HLDAK}$			
P96	$\overline{HLDRQ}$			
P100	KR0/TO7/A5 Note 1	8-A	$PORTV_{DD1}$	Input: Independently connect to $PORTV_{DD1}$, PORTGND0 or PORTGND1 via a resistor. When connecting to PORTGND0 or PORTGND1, disconnect on-chip pull-up resistors by software. Output: Leave open.
P101	KR1/TI70/A6 Note 1			
P102	KR2/TI00/A7 Note 1			
P103	KR3/TI01/A8 Note 1			
P104	KR4/TO0/A9 Note 1			
P105	KR5/TI10/A10 Note 1			
P106	KR6/TI11/A11 Note 1			
P107	KR7/TO1/A12 Note 1			
P110	$\overline{WAIT}$/A1 Note 1	5	$PORTV_{DD1}$	Input: Independently connect to $PORTV_{DD1}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P111 to P113	A2 to A4 Note 1			
P114	CANTX1 Note 2			
P115	CANRX1 Note 2	8		
P116	CANTX2 Note 3	5		
P117	CANRX2 Note 3	8		
P120	$\overline{SCK5}$	8	$PORTV_{DD2}$	Input: Independently connect to $PORTV_{DD2}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P121	SI5			
P122	SO5	5		
P123	$\overline{SCK6}$	8		
P124	SI6			
P125	SO6	5		
P126, P127	TO10, TO11			
P130 to P133	–	5	$PORTV_{DD2}$	Input: Independently connect to $PORTV_{DD2}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.

Notes 1. Only for the V850/SC1 and V850/SC2

2. Only for the V850/SC3

3. Only for the μPD703089Y and 70F3089Y

(3/3)

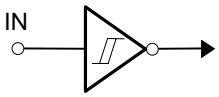
Pin	Alternate Function	I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection
P140	SI3/RXD1	8	$PORTV_{DD2}$	Input: Independently connect to $PORTV_{DD2}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P141	SO3/TXD1	5		
P142	$\overline{SCK3}$/ASCK1	8		
P143	RXD2			
P144	TXD2	5		
P145	ASCK2	8		
P146, P147	TI100, TI101			
P150	RXD3	8	$PORV_{DD2}$	Input: Independently connect to $PORTV_{DD2}$, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P151	TXD3	5		
P152	ASCK3	8		
P153, P154	TI110, TI111			
P155	TO12	5		
P156, P157	TI120, TI121	8		
P170 to P175	–	5	V_{DD1}	Input: Independently connect to V_{DD1}, PORTGND0, or PORTGND1 via a resistor. Output: Leave open.
P176	VM45			
CLKOUT	–	4	$PORTV_{DD0}$	Leave open.
$\overline{RESET}$	–	2	V_{DD0}	–
X1, X2	–	–	CPUREG	–
XT1, XT2	–	–		–
CPUREG	–	–	–	–
V_{PP}[Note 1]	MODE	–	V_{DD0}	Connect to either GND0, GND1, or GND2.
MODE[Note 2]	V_{PP}	–	V_{DD0}	Connect to either GND0, GND1, or GND2.
V_{DD0}, V_{DD1}	–	–	–	–
GND0 to GND2	–	–	–	–
$ADCV_{DD}$	–	–	–	–
ADCGND	–	–	–	–
$PORTV_{DD0}$ to $PORTV_{DD2}$	–	–	–	–
PORTGND0, PORTGND1	–	–	–	–

Notes 1. μPD70F3089Y

2. Other than μPD70F3089Y

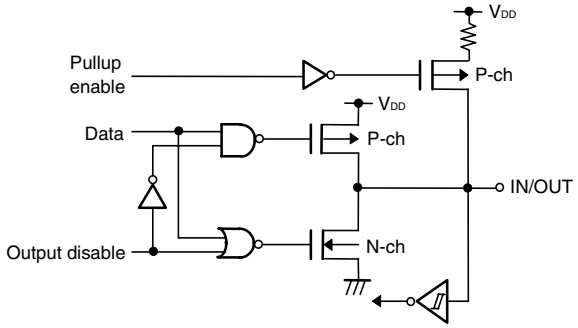
2.5 Pin I/O Circuits

Type 2

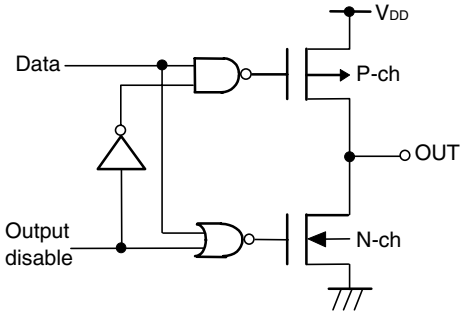


Schmitt-triggered input with hysteresis characteristics

Type 8-A

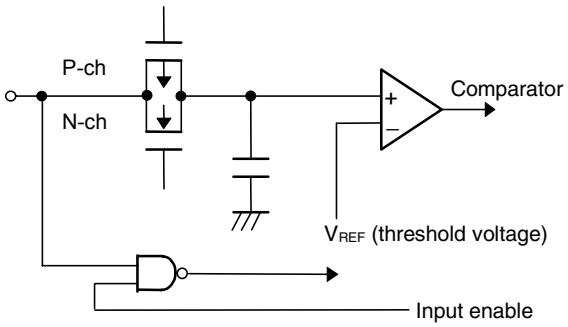


Type 4

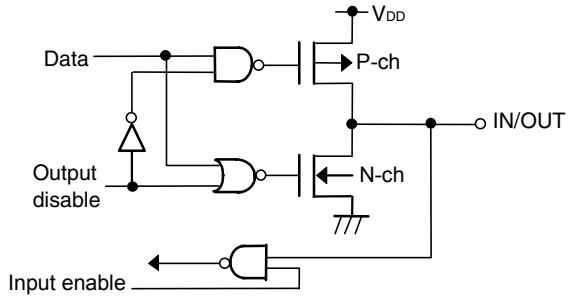


Push-pull output that can be set for high impedance output (both P-ch and N-ch off).

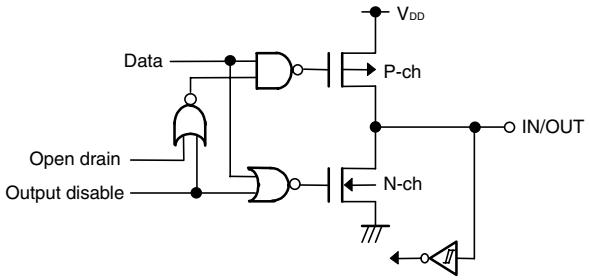
Type 9



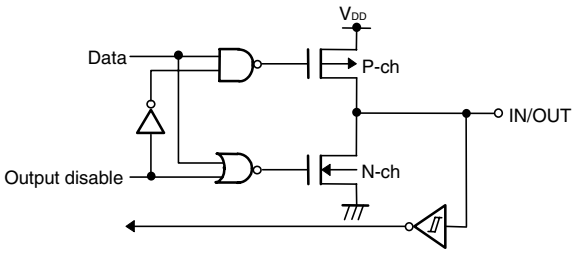
Type 5



Type 10



Type 8



CHAPTER 3 CPU FUNCTIONS

The CPU of the V850/SC1, V850/SC2, and V850/SC3 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/SC1: 50 ns (@ 20 MHz internal operation)
  V850/SC2: 53 ns (@ 18.87 MHz internal operation)
  V850/SC3: 62.5 ns (@ 16 MHz internal operation)
- Address space: 16 MB linear
- General-purpose registers: 32 bits × 32
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instructions 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/SC1, V850/SC2, and V850/SC3 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32-bit width. 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 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used. r2 is sometimes used by a real-time OS. r2 can be used as a variable register when the real-time OS that is used does not use r2.

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 being used)	
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 functions
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 is not possible.

After reset: 00000000H

Symbol	31 24	23 1	0
PC	Fixed to 0	Address of instruction under execution	0

3.2.2 System register set

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

Table 3-2. System Register Numbers

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

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

(1) Interrupt source register (ECR)

After reset: 00000000H

Symbol	31	16	15	0
ECR	FECC		EICC	

FECC	Exception code of NMI (for the exception code, refer to **Table 7-1**.)
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 **7.2.3 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 **7.4.3 EP flag.**

ID	Maskable interrupt servicing specification
0	Maskable interrupt acknowledgement enabled.
1	Maskable interrupt acknowledgement disabled. This flag is set (1) when a maskable interrupt request is acknowledged. For details, refer to **7.3.6 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	Overflow has not occurred.
1	Overflow occurred.

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

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

(2/2)

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

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

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

3.3 Operating Modes

The V850/SC1, V850/SC2, and V850/SC3 have the following operating modes.

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

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

(2) Flash memory programming mode

This mode is provided only in the μPD70F3089Y. The internal flash memory can be programmed or erased when the V_{PP} voltage is applied to the V_{PP} pin.

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

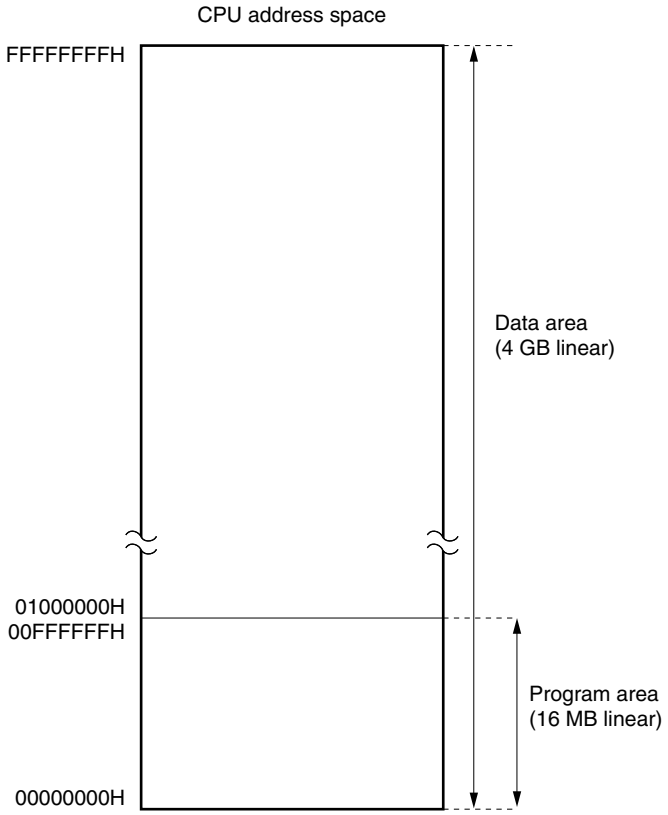
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850/SC1, V850/SC2, and V850/SC3 has 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). When referencing instruction addresses, a 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

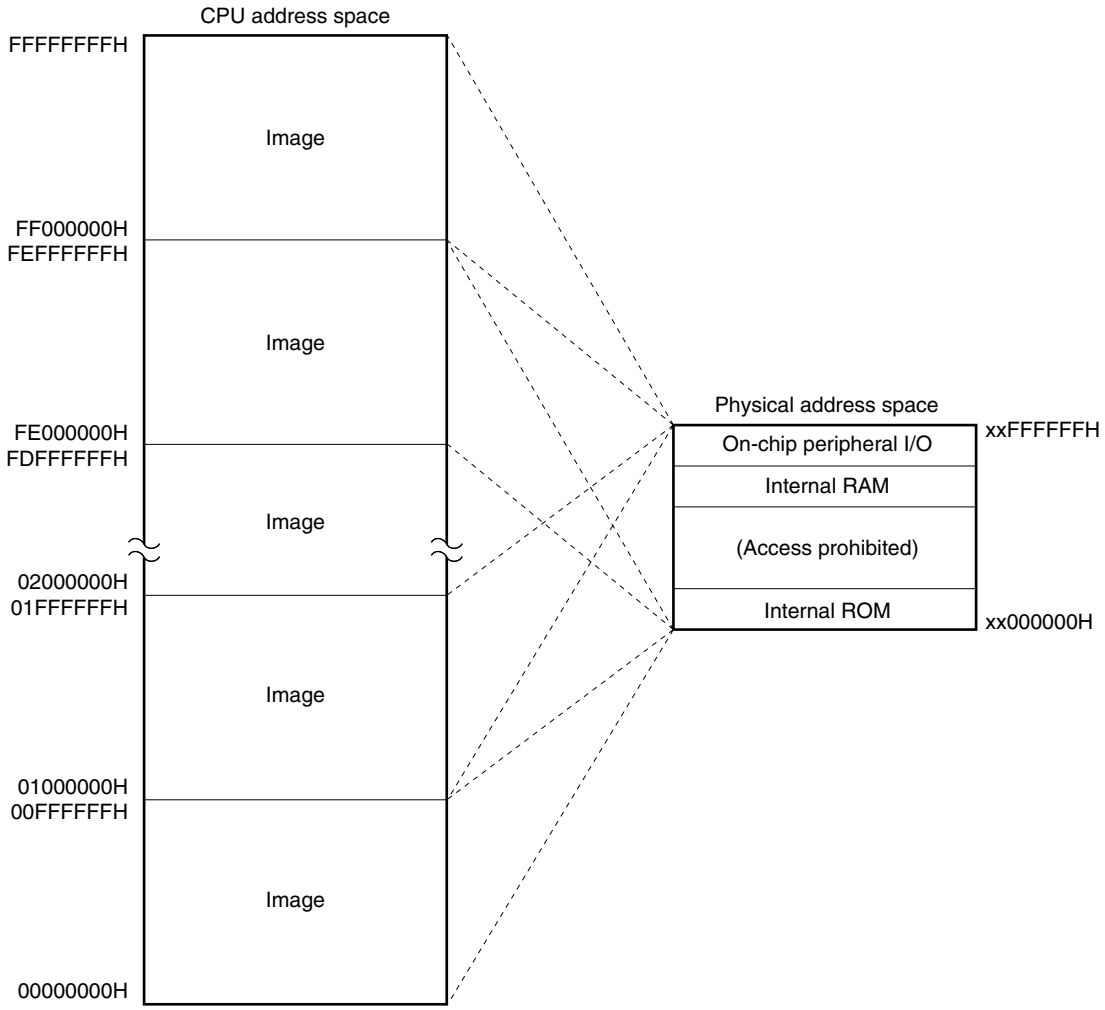


3.4.2 Imaging

The 4 GB CPU address space can be viewed as 256 images of a 16 MB physical address space. In other words, the same 16 MB block is accessed regardless of the values of bits 31 to 24 of the CPU address. The address space imaging 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. Address Space Imaging



3.4.3 Wrap-around of CPU address space

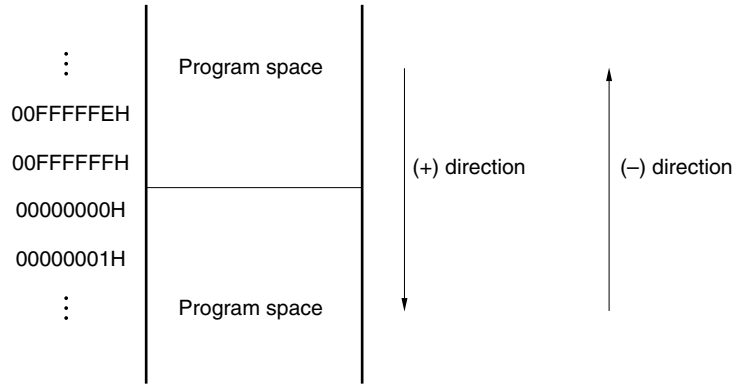
(1) Program space

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

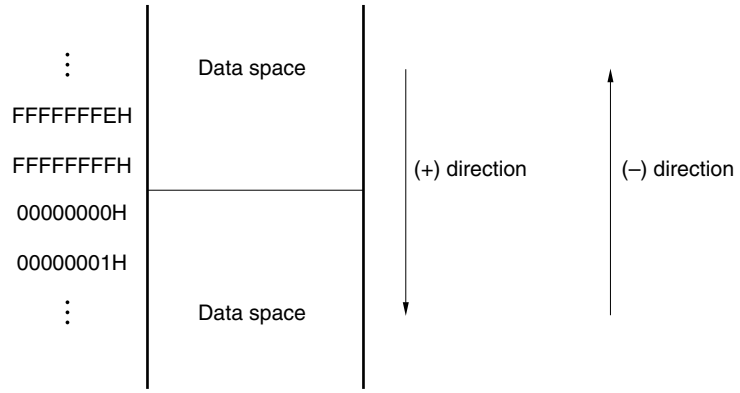


(2) Data space

The result of an 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.

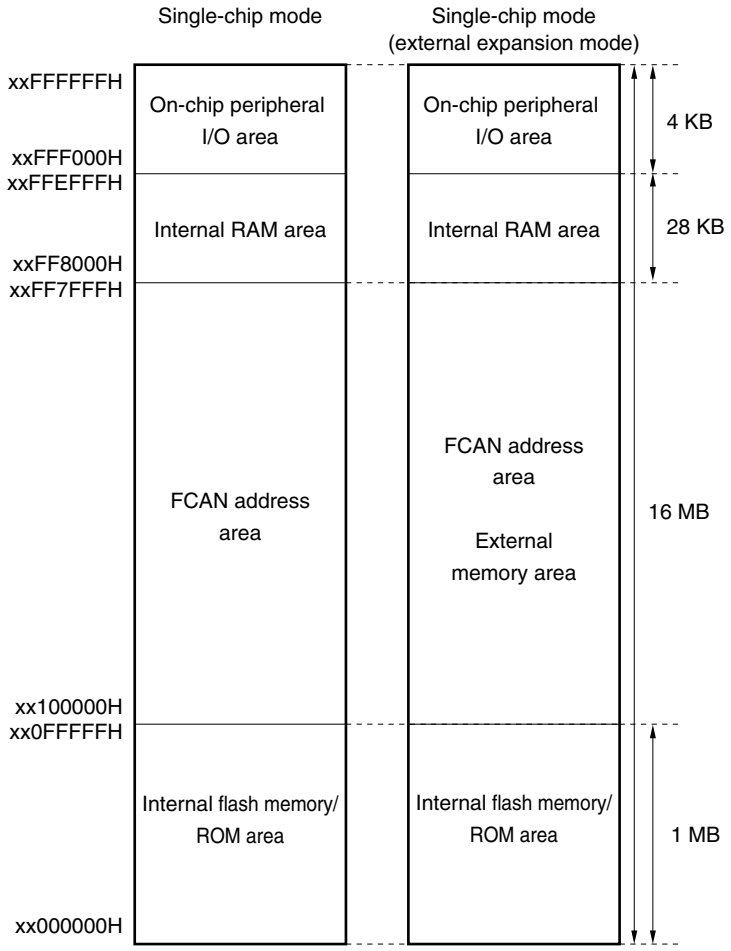
Figure 3-5. Data Space



3.4.4 Memory map

The V850/SC1, V850/SC2, and V850/SC3 reserve areas as shown below.

Figure 3-6. Memory Map



3.4.5 Area

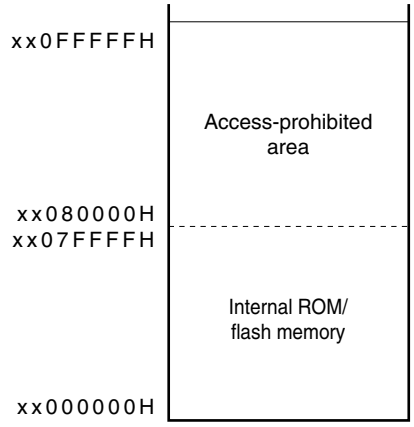
(1) Internal ROM/flash memory area

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

512 KB are available for the addresses xx000000H to xx07FFFFH.

Addresses xx080000H to xx0FFFFFH are an access-prohibited area

Figure 3-7. Internal ROM/Flash Memory Area



Interrupt/exception table

The V850/SC1, V850/SC2, and V850/SC3 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	00000210H	INTSR0/INTCSI4
00000010H	NMI	00000220H	INTST0
00000020H	INTWDT	00000230H	INTKR
00000040H	TRAP0n (n = 0 to F)	00000240H	INTCE1Note 2/INTIE1Note 3
00000050H	TRAP1n (n = 0 to F)	00000250H	INTCR1Note 2/INTIE2Note 3
00000060H	ILGOP	00000260H	INTCT1Note 2
00000080H	INTWDTM	00000270H	INTCMENote 2
00000090H	INTP0	00000280H	INTTM80
000000A0H	INTP1	00000290H	INTTM81
000000B0H	INTP2	000002A0H	INTTM90
000000C0H	INTP3	000002B0H	INTTM91
000000D0H	INTP4	000002C0H	INTSR1/INTCSI3
000000E0H	INTP5	000002D0H	INTST1
000000F0H	INTP6	000002E0H	INTDMA3
00000100H	INTCSI5	000002F0H	INTDMA4
00000110H	INTAD	00000300H	INTDMA5
00000120H	INTDMA0	00000310H	INTCE2Note 4
00000130H	INTDMA1	00000320H	INTCR2Note 4
00000140H	INTDMA2	00000330H	INTCT2Note 4
00000150H	INTTM00	00000340H	INTP7
00000160H	INTTM01	00000350H	INTSR2
00000170H	INTTM10	00000360H	INTST2
00000180H	INTTM11	00000370H	INTSR3
00000190H	INTTM70	00000380H	INTST3
000001A0H	INTTM71	00000390H	INTTM100
000001B0H	INTCSI6	000003A0H	INTTM101
000001C0H	INTTM5/INTP8Note 1	000003B0H	INTTM110
000001D0H	INTWTN	000003C0H	INTTM111
000001E0H	INTWTNI	000003D0H	INTTM120
000001F0H	INTIIC0/INTCSI0	000003E0H	INTTM121
00000200H	INTTM6/INTP9Note 1	000003F0H	INTIIC1/INTCSI2

Notes 1. When using INTP8 or INTP9, stop TM5 and TM6 (TCEm0 bit of TMCm0 register = 0) and do not use them. When using TM5 or TM6, do not specify edges for INTP8 and INTP9 (EGP1n bit of EGP1 register = 0 and EGN1n bit of EGN1 register = 0) and do not use them as external interrupts (they can be used as ports) (n = 6, 7).

2. Only for the V850/SC3

3. Only for the V850/SC2

4. Only for the μPD703089Y and 70F3089Y

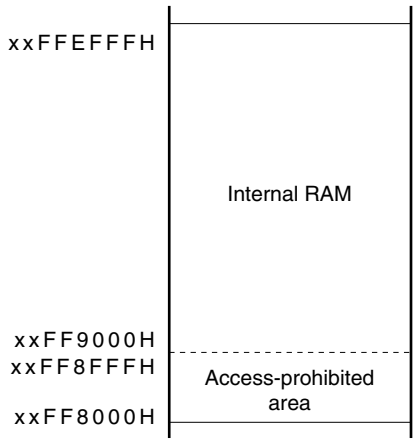
(2) Internal RAM area

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

24 KB are available for the addresses xxFF9000H to xxFFEFFFH.

Addresses xxFF8000H to xxFF8FFFH are an access-prohibited area

Figure 3-8. Internal RAM Area

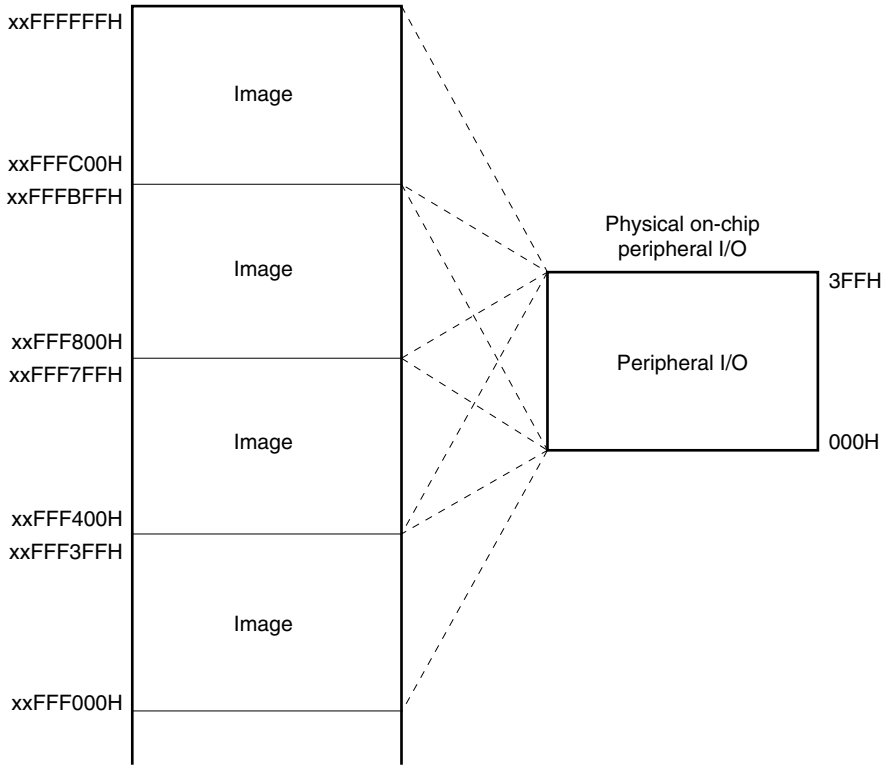


(3) On-chip peripheral I/O area

A 4 KB area of addresses FFF000H to FFFFFFH is reserved as an on-chip peripheral I/O area. The V850/SC1, V850/SC2, and V850/SC3 are provided with a 1 KB area of addresses FFF000H to FFF3FFH as a physical on-chip peripheral I/O area. The rest of the area (FFF400H to FFFFFFH) shows images of these addresses.

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

Figure 3-9. 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 address n that can be accessed only in halfword units is accessed in word units, the operation is replaced with two halfword operations. The first operation (lower 16 bits) accesses the register with address n and the second operation (higher 16 bits) accesses the register with address n + 2.

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

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

(4) External memory

The V850/SC1, V850/SC2, and V850/SC3 can use an area of up to 16 MB (xx100000H to xxFF7FFFH) for external memory accesses (in single-chip mode: external expansion).

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

Caution Addresses xxnFF800H to xxnFFFFFH (n = 3, 7, B) constitute an access-prohibited area because this is an FCAN address area.

Figure 3-10. External Memory Area (When Expanded to 64 KB, 256 KB, or 1 MB)

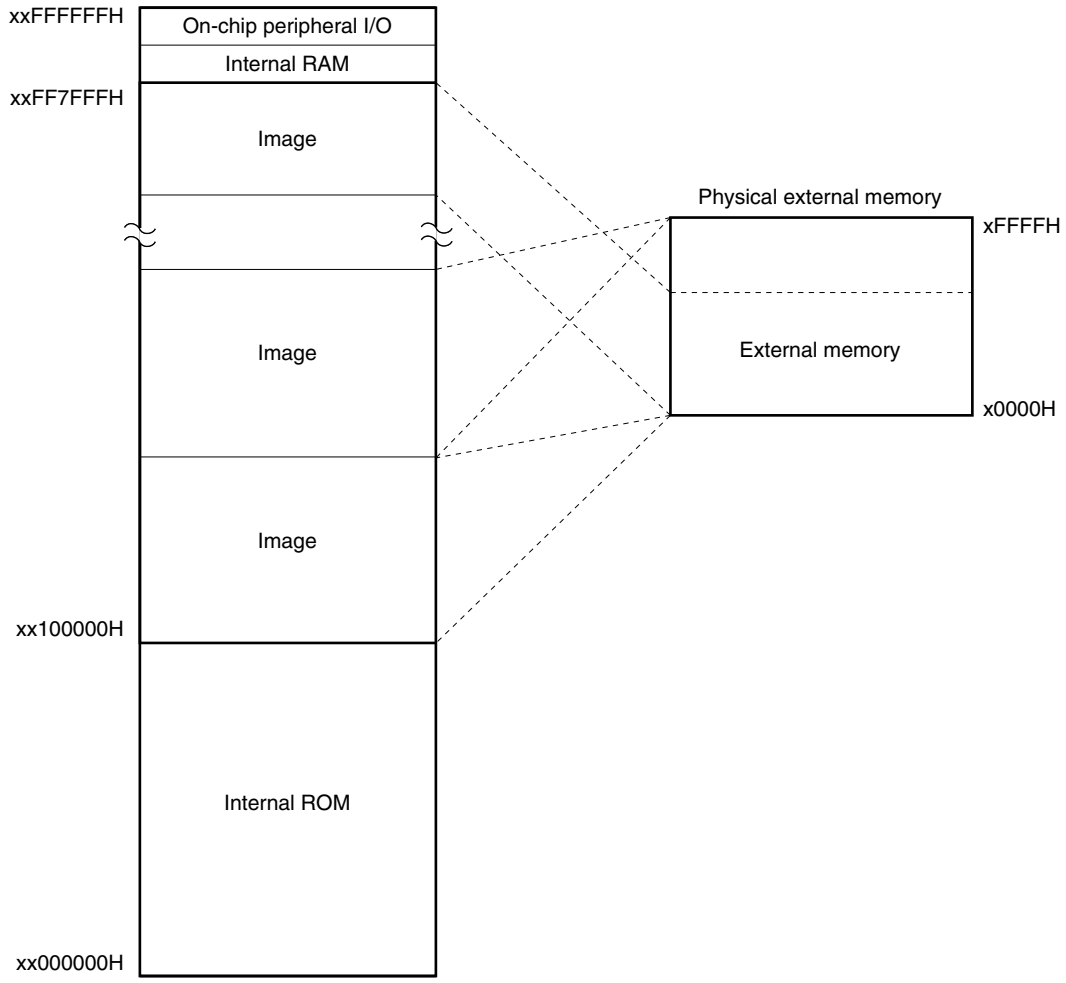
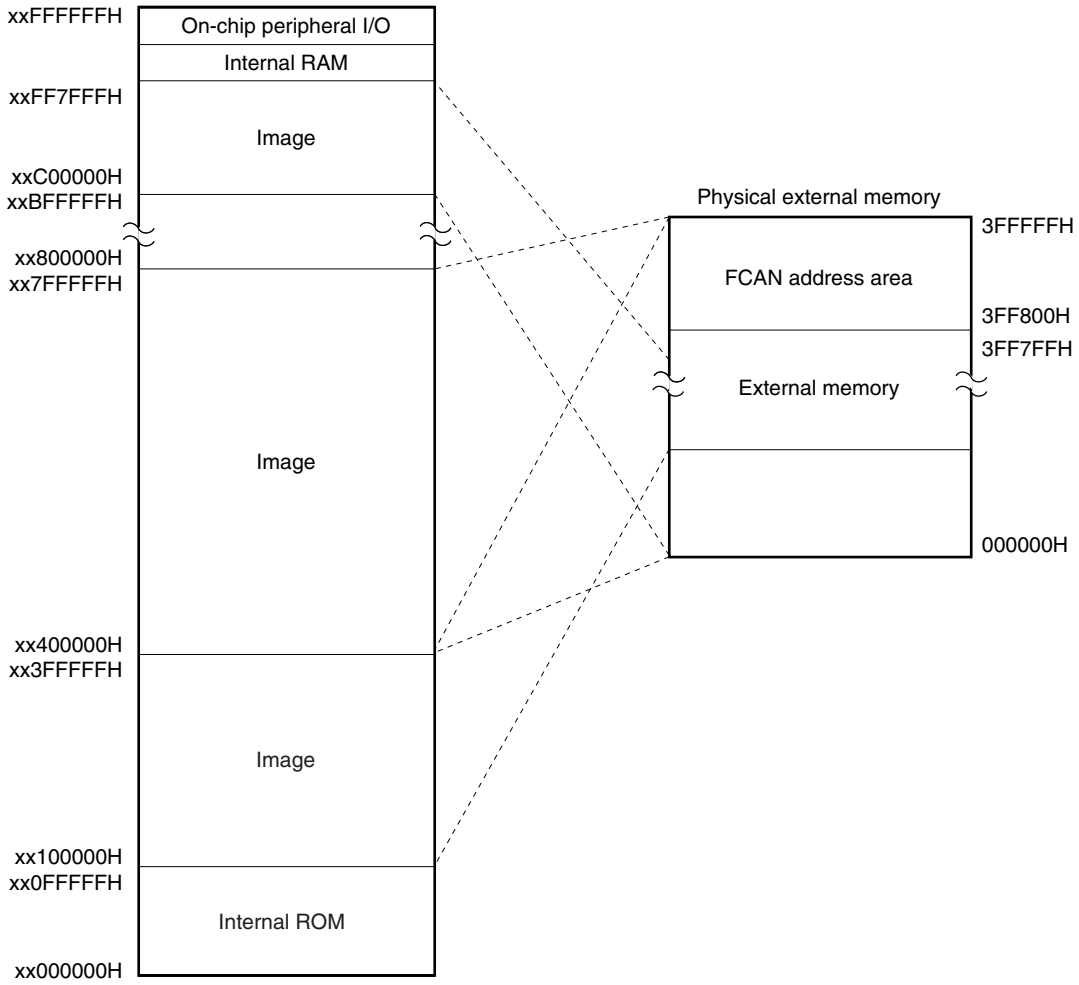


Figure 3-11. External Memory Area (When Expanded to 4 MB)



3.4.6 External expansion mode

The V850/SC1, V850/SC2, and V850/SC3 allow external devices to be connected to the external memory space 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 using the memory expansion mode register (MM).

In the V850/SC1 and V850/SC2, the address bus (A1 to A15) is set to multiplexed output with the data bus (D1 to D15), however, separate output is also available by setting the memory address output mode register (MAM) (see the User's Manual of the 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 for the V850/SC1 and V850/SC2, the $\overline{\text{WAIT}}$ pin based wait function cannot be used when using a separate bus (programmable wait can be used, however). Similarly, a separate bus cannot be used when the $\overline{\text{WAIT}}$ pin based wait function is being used.

Because the V850/SC1, V850/SC2, and V850/SC3 are fixed to single-chip mode in the normal operating mode, the pins related to the bus interface are set to port mode, disabling use of the external memory. When the external memory is used (external expansion mode), specify the MM register or MAM register by program (for the V850/SC1 and V850/SC2) or specify the MM register by program (for the V850/SC3).

(1) Memory expansion mode register (MM)

This register sets the mode of each pin of ports 4, 5, 6, and 9 in the V850/SC1, V850/SC2, and V850/SC3. In the external expansion mode, an external device can be connected to an 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- 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 operating modes
0	Port mode
1	External expansion mode ($\overline{\text{HLDAK}}$: P95, $\overline{\text{HLDRQ}}$: P96)

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

Note Only for the V850/SC1 and V850/SC2

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

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

(2) Memory address output mode register (MAM)

This register sets the mode of ports 3, 10, and 11 in the V850/SC1 and V850/SC2. 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

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

Caution The memory address output mode register (MAM) cannot be debugged by an in-circuit emulator. Also, switching to a separate bus is not possible by setting the MAM register using software.
For details, refer to the User's Manual of the relevant in-circuit emulator.

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

P34 to P36, P100 to P107, and P110 to P113 are used for separate bus output. The procedure for performing separate bus 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 bus is used as an alternate-function pin for other than the separate bus, turn off the function that uses the alternate-function pin.
<4> Set the memory address output mode register (MAM).
<5> Set the memory expansion mode register (MM).

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

3.4.7 Recommended use of address space

The architecture of the V850/SC1, V850/SC2, and V850/SC3 requires that a register that serves as a pointer be secured for address generation when accessing operand data in the data space. Operand data can be accessed directly from an instruction in the ±32 KB above and below the address in this pointer register. However, the general-purpose registers that can be used as a pointer register are 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/SC1, V850/SC2, and V850/SC3, 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 utilizing 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/SC1, V850/SC2, and V850/SC3, a 16 MB physical address space is seen as 256 images in the 4 GB CPU address space. The highest bit (bit 23) of this 24-bit address is assigned as an address sign-extended to 32 bits.

(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 by a sign-extended, 16-bit displacement value. All resources including on-chip hardware can therefore 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 3-12. Application of Wrap-Around

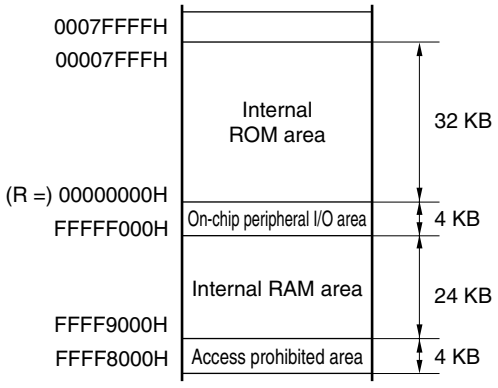
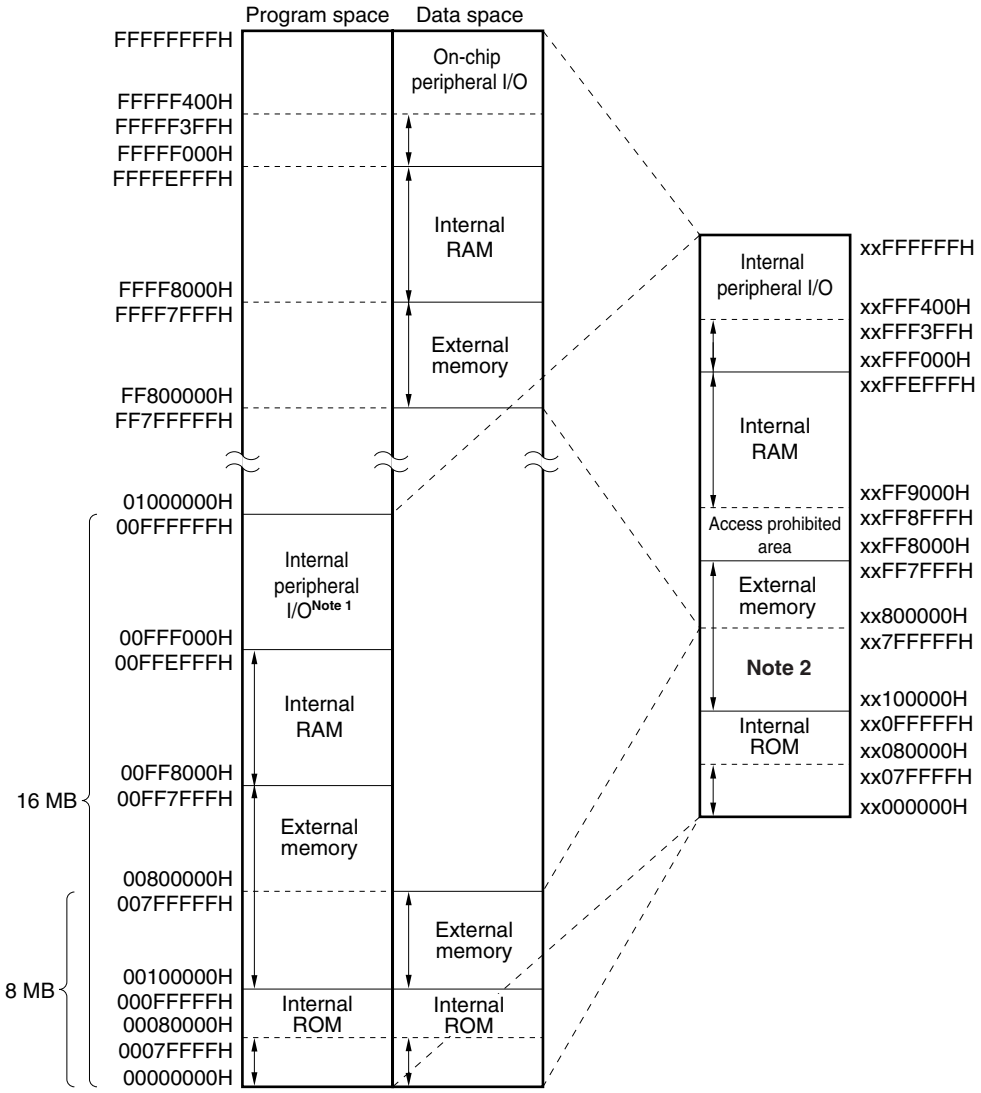


Figure 3-13. Recommended Memory Map (Flash Memory Version)



Notes 1. This area cannot be used as a program area.

2. Addresses xxnF800H to xxnFFFFH (n = 3, 7, B) constitute an FCAN address area and cannot be accessed during external expansion.

Remarks 1. The arrows indicate the recommended area.

2. This is a recommended memory map for the μPD70F3089Y.

3.4.8 Peripheral I/O registers

(1/10)

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		√	√			
FFFFF018H	Port 12	P12		√	√			
FFFFF01AH	Port 13	P13		√	√			
FFFFF01CH	Port 14	P14		√	√			
FFFFF01EH	Port 15	P15		√	√			
FFFFF020H	Port 0 mode register	PM0		√	√			FFH
FFFFF022H	Port 1 mode register	PM1		√	√			FFH
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		√	√			FFH
FFFFF038H	Port 12 mode register	PM12		√	√			
FFFFF03AH	Port 13 mode register	PM13		√	√			0FH
FFFFF03CH	Port 14 mode register	PM14		√	√			FFH
FFFFF03EH	Port 15 mode register	PM15		√	√			
FFFFF040H	Port alternate-function control register	PAC		√	√			00H
FFFFF042H	Port alternate-function control register 2	PAC2		√	√			
FFFFF048H	Port 17	P17		√	√			
FFFFF04CH	Memory expansion mode register	MM		√	√			
FFFFF058H	Port 17 mode register	PM17		√	√			7FH

Note Resetting initializes registers to input mode, so the pin level is read during a read operation. The output latch is initialized to 00H.

(2/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF060H	Data wait control register	DWC	R/W			√		FFFFH
FFFFF062H	Bus cycle control register	BCC				√		AAAAH
FFFFF064H	System control register[Note 1]	SYC		√	√			00H
FFFFF068H	Memory address output mode register[Note 1]	MAM	W		√			
FFFFF070H	Power save control register	PSC	R/W	√	√			C0H
FFFFF074H	Processor clock control register	PCC		√	√			03H
FFFFF076H	POC control register	POCC			√			00H
FFFFF078H	System status register	SYS		√	√			00H
FFFFF07AH	POC status register	POCS	R		√			Held[Note 2]
FFFFF07CH	VM45 control register	VM45C	R/W		√			00H
FFFFF094H	Pull-up resistor option register 10	PU10		√	√			
FFFFF0A2H	Port 1 function register	PF1		√	√			
FFFFF0A4H	Port 2 function register	PF2		√	√			
FFFFF0C0H	Rising edge specification register 0	EGP0		√	√			
FFFFF0C2H	Falling edge specification register 0	EGN0		√	√			
FFFFF0C4H	Rising edge specification register 1	EGP1		√	√			
FFFFF0C6H	Falling edge specification register 1	EGN1		√	√			
FFFFF0D0H	16-bit timer register 10	TM10	R			√		0000H
★ FFFFF0D2H	16-bit capture/compare register 100	CR100	**Note 3**			√		
★ FFFFF0D4H	16-bit capture/compare register 101	CR101				√		
FFFFF0D6H	Prescaler mode register 100	PRM100	R/W		√			00H
FFFFF0D8H	16-bit timer mode control register 10	TMC10		√	√			
FFFFF0DAH	Capture/compare control register 10	CRC10		√	√			
FFFFF0DCH	Timer output control register 10	TOC10		√	√			
FFFFF0DEH	Prescaler mode register 101	PRM101			√			
FFFFF0E0H	16-bit timer register 11	TM11	R			√		0000H
★ FFFFF0E2H	16-bit capture/compare register 110	CR110	**Note 3**			√		
★ FFFFF0E4H	16-bit capture/compare register 111	CE111				√		
FFFFF0E6H	Prescaler mode register 110	PRM110	R/W		√			00H
FFFFF0E8H	16-bit timer mode control register 11	TMC11		√	√			
FFFFF0EAH	Capture/compare control register 11	CRC11		√	√			
FFFFF0ECH	Timer output control register 11	TOC11		√	√			
FFFFF0EEH	Prescaler mode register 111	PRM111			√			
FFFFF0F0H	16-bit timer register 12	TM12	R			√		0000H
★ FFFFF0F2H	16-bit capture/compare register 120	CR120	**Note 3**			√		
★ FFFFF0F4H	16-bit capture/compare register 121	CR121				√		

Notes 1. Only for the V850/SC1 and V850/SC2

2. This value is 03H only after a power-on-clear reset. This cannot be reset by $\overline{\text{RESET}}$ signal input or watchdog timer.

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

(3/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF0F6H	Prescaler mode register 120	PRM120	R/W		√			00H
FFFFF0F8H	16-bit timer mode control register 12	TMC12		√	√			
FFFFF0FAH	Capture/compare control register 12	CRC12		√	√			
FFFFF0FCH	Timer output control register 12	TOC12		√	√			
FFFFF0FEH	Prescaler mode register 121	PRM121			√			
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		√	√			
FFFFF110H	Interrupt control register	CSIC5		√	√			
FFFFF112H	Interrupt control register	ADIC		√	√			
FFFFF114H	Interrupt control register	DMAIC0		√	√			
FFFFF116H	Interrupt control register	DMAIC1		√	√			
FFFFF118H	Interrupt control register	DMAIC2		√	√			
FFFFF11AH	Interrupt control register	TMIC00		√	√			
FFFFF11CH	Interrupt control register	TMIC01		√	√			
FFFFF11EH	Interrupt control register	TMIC10		√	√			
FFFFF120H	Interrupt control register	TMIC11		√	√			
FFFFF122H	Interrupt control register	TMIC70		√	√			
FFFFF124H	Interrupt control register	TMIC71		√	√			
FFFFF126H	Interrupt control register	CSIC6		√	√			
FFFFF128H	Interrupt control register	TMIC5		√	√			
FFFFF12AH	Interrupt control register	WTNIC		√	√			
FFFFF12CH	Interrupt control register	WTNIIC		√	√			
FFFFF12EH	Interrupt control register	CSIC0		√	√			
FFFFF130H	Interrupt control register	TMIC6		√	√			
FFFFF132H	Interrupt control register	CSIC4		√	√			
FFFFF134H	Interrupt control register	STIC0		√	√			
FFFFF136H	Interrupt control register	KRIC		√	√			
FFFFF138H	Interrupt control register	**Note 1**		√	√			
FFFFF13AH	Interrupt control register	**Note 2**		√	√			
FFFFF13CH	Interrupt control register[Note 3]	CANIC3		√	√			
FFFFF13EH	Interrupt control register[Note 3]	CANIC7		√	√			

Notes 1. CANIC1 (V850/SC3)/IEBIC1 (V850/SC2)

2. CANIC2 (V850/SC3)/IEBIC2 (V850/SC2)

3. Only for the V850/SC3

(4/10)

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	TMIC80	R/W	√	√			47H
FFFFF142H	Interrupt control register	TMIC81		√	√			
FFFFF144H	Interrupt control register	TMIC90		√	√			
FFFFF146H	Interrupt control register	TMIC91		√	√			
FFFFF148H	Interrupt control register	CSIC3		√	√			
FFFFF14AH	Interrupt control register	STIC1		√	√			
FFFFF14CH	Interrupt control register	DMAIC3		√	√			
FFFFF14EH	Interrupt control register	DMAIC4		√	√			
FFFFF150H	Interrupt control register	DMAIC5		√	√			
FFFFF152H	Interrupt control register[Note]	CANIC4		√	√			
FFFFF154H	Interrupt control register[Note]	CANIC5		√	√			
FFFFF156H	Interrupt control register[Note]	CANIC6		√	√			
FFFFF158H	Interrupt control register	PIC7		√	√			
FFFFF15AH	Interrupt control register	SRIC2		√	√			
FFFFF15CH	Interrupt control register	STIC2		√	√			
FFFFF15EH	Interrupt control register	SRIC3		√	√			
FFFFF160H	Interrupt control register	STIC3		√	√			
FFFFF162H	Interrupt control register	TMIC100		√	√			
FFFFF164H	Interrupt control register	TMIC101		√	√			
FFFFF166H	In-service priority register	ISPR	R	√	√			00H
FFFFF168H	Interrupt control register	TMIC110	R/W	√	√			47H
FFFFF16AH	Interrupt control register	TMIC111		√	√			
FFFFF16CH	Interrupt control register	TMIC120		√	√			
FFFFF16EH	Interrupt control register	TMCI121		√	√			
FFFFF170H	Command register	PRCMD	W		√			Undefined
FFFFF172H	Interrupt control register	CSIC2	R/W	√	√			47H
FFFFF180H	DMA peripheral I/O address register 0	DIOA0				√		Undefined
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			√			

Note Available only for the μPD703089Y and 70F3089Y.

(5/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF1A6H	DMA channel control register 2	DCHC2	R/W	√	√			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				√		
FFFFF1D4H	DMA byte count register 5	DBC5			√			
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				√		
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				√		
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			√			
FFFFF230H	Asynchronous serial interface mode register 2	ASIM2		√	√			
FFFFF232H	Asynchronous serial interface status register 2	ASIS2	R	√	√			
FFFFF234H	Baud rate generator control register 2	BRGC2	R/W		√			
FFFFF236H	Transmit shift register 2	TXS2	W		√			FFH
FFFFF238H	Receive buffer register 2	RXB2	R		√			
FFFFF23AH	Baud rate generator mode control register 20	BRGMC 20	R/W		√			00H

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

(6/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF23CH	Baud rate generator mode control register 21	BRGMC21	R/W		√			00H
FFFFF240H	Clocked serial interface mode register 5	CSIM5	R/W	√	√			
FFFFF242H	Clocked serial interface clock select register 5	CSICK5		√	√			
FFFFF244H	Clocked serial interface receive buffer register 5	SIRB5	R			√		0000H
FFFFF246H	Clocked serial interface receive buffer register L5	SIRBL5			√			00H
FFFFF248H	Clocked serial interface transmit buffer register 5	SOTB5	R/W			√		0000H
★ FFFFF24AH	Clocked serial interface transmit buffer register L5	SOTBL5		√	√			00H
FFFFF24CH	Clocked serial interface read-only receive buffer register 5	SIRBE5	R			√		0000H
★ FFFFF24EH	Clocked serial interface read-only receive buffer register L5	SIRBEL5		√	√			00H
FFFFF250H	Clocked serial interface initial transmit buffer register 5	SOTBF5	R/W			√		0000H
★ FFFFF252H	Clocked serial interface initial transmit buffer register L5	SOTBFL5		√	√			00H
FFFFF254H	Serial I/O shift register 5	SIO5	R			√		0000H
★ FFFFF256H	Serial I/O shift register L5	SIOL5		√	√			00H
FFFFF260H	Clocked serial interface mode register 6	CSIM6	R/W	√	√			0000H
FFFFF262H	Clocked serial interface clock select register 6	CSICK6		√	√			00H
FFFFF264H	Clocked serial interface receive buffer register 6	SIRB6	R			√		0000H
FFFFF266H	Clocked serial interface receive buffer register L6	SIRBL6			√			00H
FFFFF268H	Clocked serial interface transmit buffer register 6	SOTB6	R/W			√		0000H
★ FFFFF26AH	Clocked serial interface transmit buffer register L6	SOTBL6		√	√			00H
FFFFF26CH	Clocked serial interface read-only receive buffer register 6	SIRBE6	R			√		0000H
★ FFFFF26EH	Clocked serial interface read-only receive buffer register L6	SIRBEL6		√	√			00H
FFFFF270H	Clocked serial interface initial transmit buffer register 6	SOTBF6	R/W			√		0000H
★ FFFFF272H	Clocked serial interface initial transmit buffer register L6	SOTBFL6		√	√			00H
FFFFF274H	Serial I/O shift register 6	SIO6	R			√		000H
★ FFFFF276H	Serial I/O shift register L6	SIOL6		√	√			00H
FFFFF284H	Timer clock select register 60	TCL60	R/W		√			
FFFFF286H	Timer mode control register 60	TMC60		√	√			04H[Note]

Note Although the hardware status is initialized to 04H, 00H is readout if read.

(7/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF28AH	16-bit counter 6	TM6	R			√		0000H
FFFFF28CH	16-bit compare register 6	CR6	R/W			√		
FFFFF28EH	Timer clock select register 61	TCL61			√			00H
FFFFF2A0H	Serial I/O shift register 0	SIO0			√			
FFFFF2A2H	Serial operation mode register 0	CSIM0		√	√			
FFFFF2A4H	Serial clock select register 0	CSIS0			√			
FFFFF2B0H	Asynchronous serial interface mode register 3	ASIM3		√	√			
FFFFF2B2H	Asynchronous serial interface status register 3	ASIS3	R	√	√			
FFFFF2B4H	Baud rate generator control register 3	BRGC3	R/W		√			
FFFFF2B6H	Transmit shift register 3	TXS3	W		√			FFH
FFFFF2B8H	Receive buffer register 3	RXB3	R		√			
FFFFF2BAH	Baud rate generator mode control register 30	BRGMC 30	R/W		√			00H
FFFFF2BCH	Baud rate generator mode control register 31	BRGMC 31			√			
FFFFF2C0H	Serial I/O shift register 2	SIO2			√			
FFFFF2C2H	Serial operation mode register 2	CSIM2		√	√			
FFFFF2C4H	Serial clock select register 2	CSIS2			√			
FFFFF2D0H	Serial I/O shift register 3	SIO3			√			
FFFFF2D2H	Serial operation mode register 3	CSIM3		√	√			
FFFFF2D4H	Serial clock select register 3	CSIS3			√			
FFFFF2E0H	Variable-length serial I/O shift register 4	SIO4				√		0000H
FFFFF2E2H	Variable-length serial control register 4	CSIM4		√	√			00H
FFFFF2E4H	Variable-length serial setting register 4	CSIB4		√	√			
FFFFF2E6H	Baud rate generator source clock select register 4	BRGCN4			√			
FFFFF2E8H	Baud rate output clock select 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	Transmit shift register 1	TXS1	W		√			FFH
FFFFF318H	Receive buffer register 1	RXB1	R		√			

(8/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
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			√			
FFFFF334H	Timer clock select register 50	TCL50			√			
FFFFF336H	Timer mode control register 50	TMC50		√	√			04H[Note]
FFFFF33AH	16-bit counter 5	TM5	R			√		0000H
FFFFF33CH	16-bit compare register 5	CR5	R/W			√		
FFFFF33EH	Timer clock select register 51	TCL51			√			00H
FFFFF340H	IIC control register 0	IICC0		√	√			
FFFFF342H	IIC status register 0	IICS0	R	√	√			
FFFFF344H	IIC clock select register 0	IICCL0	R/W	√	√			
FFFFF346H	Slave address register 0	SVA0			√			
FFFFF348H	IIC shift register 0	IIC0			√			
FFFFF34AH	IIC function expansion register 0	IICX0		√	√			
FFFFF34CH	IIC clock expansion register 0	IICCE0			√			
FFFFF350H	IIC control register 1	IICC1		√	√			
FFFFF352H	IIC status register 1	IICS1	R	√	√			
FFFFF354H	IIC clock select register 1	IICCL1	R/W	√	√			
FFFFF356H	Slave address register 1	SVA1			√			
FFFFF358H	IIC shift register 1	IIC1			√			
FFFFF35AH	IIC function expansion register 1	IICX1		√	√			
FFFFF35CH	IIC clock expansion register 1	IICCE1			√			
FFFFF360H	Watch timer mode register	WTNM		√	√			
FFFFF364H	Watch timer clock select register	WTNCS			√			
★ FFFFF366H	Watch timer high-speed clock select register	WTNHC			√			
★ FFFFF368H	IIC flag register 0	IICF0		√	√			
★ FFFFF36AH	IIC flag register 1	IICF1		√	√			
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 select register	OSTS			√			01H
FFFFF382H	Watchdog timer clock select register	WDCS			√			00H
FFFFF384H	Watchdog timer mode register	WDTM		√	√			
FFFFF38EH	DMA start factor expansion register	DMAS		√	√			

Note Although the hardware status is initialized to 04H, 00H is readout if read.

(9/10)

	Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
					1 Bit	8 Bits	16 Bits	32 Bits	
	FFFFF390H	16-bit timer register 8	TM8	R			√		0000H
★	FFFFF392H	16-bit capture/compare register 80	CR80	**Note 1**			√		
★	FFFFF394H	16-bit capture/compare register 81	CR81				√		
	FFFFF396H	Prescaler mode register 80	PRM80	R/W		√			00H
	FFFFF398H	16-bit timer mode control register 8	TMC8		√	√			
	FFFFF39AH	Capture/compare control register 8	CRC8		√	√			
	FFFFF39CH	Timer output control register 8	TOC8		√	√			
	FFFFF39EH	Prescaler mode register 81	PRM81			√			
	FFFFF3A0H	16-bit timer register 7	TM7	R			√		0000H
★	FFFFF3A2H	16-bit capture/compare register 70	CR70	**Note 1**			√		
★	FFFFF3A4H	16-bit capture/compare register 71	CR71				√		
	FFFFF3A6H	Prescaler mode register 70	PRM70	R/W		√			00H
	FFFFF3A8H	16-bit timer mode control register 7	TMC7		√	√			
	FFFFF3AAH	Capture/compare control register 7	CRC7		√	√			
	FFFFF3ACH	Timer output control register 7	TOC7		√	√			
	FFFFF3AEH	Prescaler mode register 71	PRM71			√			
	FFFFF3B0H	16-bit timer register 9	TM9	R			√		0000H
★	FFFFF3B2H	16-bit capture/compare register 90	CR90	**Note 1**			√		
★	FFFFF3B4H	16-bit capture/compare register 91	CR91				√		
	FFFFF3B6H	Prescaler mode register 90	PRM90	R/W		√			00H
	FFFFF3B8H	16-bit timer mode control register 9	TMC9		√	√			
	FFFFF3BAH	Capture/compare control register 9	CRC9		√	√			
	FFFFF3BCH	Timer output control register 9	TOC9		√	√			
	FFFFF3BEH	Prescaler mode register 91	PRM91			√			
	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			√			
	FFFFF3E0H	IEBus control register[Note 2]	BCR		√	√			
	FFFFF3E2H	IEBus local unit register[Note 2]	UAR				√		0000H
	FFFFF3E4H	IEBus slave address register[Note 2]	SAR				√		
	FFFFF3E6H	IEBus partner address register[Note 2]	PAR	R			√		

★ **Notes 1.** In compare mode: R/W
In capture mode: R

2. Only for the V850/SC2

(10/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 Bit	8 Bits	16 Bits	32 Bits	
FFFFF3E8H	IEBus control data register[Note]	CDR	R/W		√			01H
FFFFF3EAH	IEBus telegraph length register[Note]	DLR			√			
FFFFF3ECH	IEBus data register[Note]	DR			√			00H
FFFFF3EEH	IEBus unit status register[Note]	USR	R	√	√			
FFFFF3F0H	IEBus interrupt status register[Note]	ISR	R/W	√	√			
FFFFF3F2H	IEBus slave status register[Note]	SSR	R	√	√			41H
FFFFF3F4H	IEBus communication success counter[Note]	SCR			√			01H
FFFFF3F6H	IEBus transmit counter[Note]	CCR			√			20H
FFFFF3F8H	IEBus clock select register[Note]	IECLK	R/W		√			00H

Note Only for the V850/SC2

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, the system status register (SYS) is notified. The V850/SC1, V850/SC2, and V850/SC3 have two specific registers, the power save control register (PSC) and processor clock control register (PCC). For details of the PSC register, refer to **4.3.1 (2) Power save control register (PSC),** and for details of the PCC register, refer to **4.3.1 (1) Processor clock control register (PCC).**

The following sequence shows data setting in 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> If necessary, enable DMA operation.

No special sequence is required when reading the specific registers.

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

[Description example]: In case of PCC register

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

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 4.6 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 erroneous program execution.

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

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

After reset: 00H R/W Address: FFFFF078H

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

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

The operating conditions of the PRERR flag are shown below.

(a) Set conditions (PRERR = 1)

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

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

(b) Reset conditions: (PRERR = 0)

(1) When 0 is written to the PRERR flag of the SYS register (however, excluding 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 CLOCK GENERATION FUNCTION

4.1 General

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

(1) Main clock oscillator

★ The V850/SC1 and V850/SC3 have an oscillation frequency of 4 to 20 MHz and the V850/SC2 4 to 18.87 MHz. Oscillation can be stopped by setting the STOP mode 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 clock oscillator is stopped by reset input or STOP mode setting, the oscillation stabilization time is secured after the stop mode is canceled. This oscillation stabilization time is set via the oscillation stabilization time select register (OSTS). The watchdog timer is used to count 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.

3. External clock input is disabled.

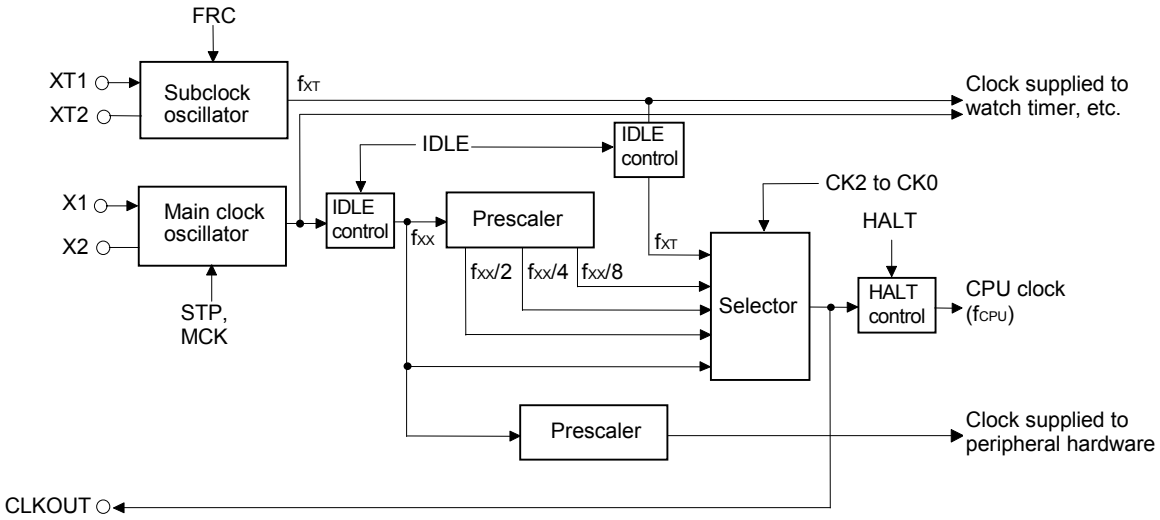
(2) Subclock oscillator

This circuit has an oscillation frequency of 32.768 kHz. Its oscillation is not stopped when the STOP mode is set, nor 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 a reduction in current consumption in the STOP mode.

4.2 Configuration

Figure 4-1. Clock Generator



4.3 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 PSC register.

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

Caution While CLKOUT is being output, changing the CPU clock (CK2 to CK0 bits of PCC register) is disabled.

4.3.1 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- 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
0	Operate
1	Stop

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. It is recommended to manipulate CK2 in 1-bit units. However, when manipulating the PCC register in 8-bit units, be sure not to change the values of CK1 and CK0.

★ **2.** Do not set the STOP mode when the CPU is operating on the subclock (CK2 = 1).

Cautions 1. Do not change the CPU clock (the values of CK2 to CK0 in the PCC register) while CLKOUT is being output.

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

★ **3. Always set bits 5 to 3 to 0.**

Remark X: don't care

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

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

<2> Subclock operation: It takes up to the following number of instructions for the subclock to start operating after the CK2 bit is set.

(CPU clock frequency before setting / subclock frequency) × 2

It is therefore necessary to insert waits equivalent to this number using a program.

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

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

<1> MCK ← 0: Main clock oscillation start

<2> Insert waits 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: It takes up to two instructions to start 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- or 1-bit units.

After reset: C0H R/W Address: FFFFF070H

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

DCLK1	DCLK0	Specification of CLKOUT pin operation
0	0	Output enabled
★ 0	1	Hi-Z output[Note 1]
1	0	Setting prohibited
1	1	Output disabled (after reset)

IDLE	IDLE mode setting
0	Normal mode
1	IDLE mode[Note 2]

STP	STOP mode setting
0	Normal mode
1	STOP mode[Note 3]

★ **Notes 1.** Hi-Z cannot be output from the in-circuit emulator.

2. When IDLE mode is canceled, this bit is automatically reset to 0.

3. When STOP mode is canceled, this bit is automatically reset to 0.

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

(3) Oscillation stabilization time select register (OSTS)

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

After reset: 01H 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	18.87 MHz	16 MHz
0	0	0	$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.10 ms
0	0	1	$2^{18}/F_{XX}$ (after reset)	13.1 ms	13.9 ms	16.4 ms
0	1	0	$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
0	1	1	$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
1	0	0	$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131 ms
Other than above			Setting prohibited			

4.4 Power Save Functions

4.4.1 General

This product provides the following power saving functions.

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

(1) HALT mode

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

A special-purpose instruction (the HALT instruction) is used to switch to HALT mode.

(2) IDLE mode

This mode stops the entire system by stopping the CPU 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 canceled, 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 in the power save control register (PSC) is set (1), the system switches to IDLE mode.

(3) Software STOP mode

This mode stops the entire system by stopping the main clock oscillators. The subclock continues to be supplied to keep on-chip peripheral functions operating. If the 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 (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 (1) to set low-power-consumption mode in which the entire system operates using only the subclock.

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

When IDLE mode has been set, the CPU operating clock and some peripheral functions (DMAC and BCU) are stopped to enable an even greater reduction in power consumption than when in HALT mode.

4.4.2 HALT mode

(1) Settings and operating states

In this mode, the clock oscillator continues to operate but the CPU operating clock is stopped. A clock continues to be supplied for other on-chip peripheral functions to maintain the 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.

When 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 via either the main clock or subclock.

The operating statuses in the HALT mode are listed in Table 4-1.

(2) Cancellation of HALT mode

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

(a) Cancellation by interrupt request

HALT mode is canceled 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) When an interrupt request that has a lower priority level than the interrupt currently being serviced occurs, only HALT mode is canceled 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) Cancellation by $\overline{\text{RESET}}$ pin input

This is the same as for normal reset operations.

Table 4-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item		When CPU Operates on Main Clock		When CPU Operates on Subclock	
		When Subclock Does Not Exist	When Subclock Exists	When Main Clock Oscillation Continues	When Main Clock Oscillation Is Stopped
CPU		Stopped			
ROM correction		Stopped			
Clock generator		Oscillation for main clock and subclock Clock supply to CPU is stopped			
16-bit timer (TM0)		Operating			Operates when INTWTNI is selected as count clock (f_{XT} is selected for watch timer)
16-bit timer (TM1)		Operating			Stopped
16-bit timer (TM5)		Operates when main clock is selected for count clock	Operating		Operates when f_{XT} is selected for count clock
16-bit timer (TM6)		Operating			Stopped
16-bit timer (TM7 to TM12)		Operating			Stopped
Watch timer		Operates when main clock is selected for count clock	Operating		Operates when f_{XT} is selected for count clock
Watchdog timer		Operating (interval timer only)			
Serial interface	CSI0, CSI2 to CSI6	Operating			Operates when an external clock is selected as the serial clock
	I^2C0, I^2C1	Operating			Stopped
	UART0 to UART3	Operating			Operates when an external clock is selected as the baud rate clock
IEBus (V850/SC2 only)		Operating			Stopped
FCAN1, FCAN2[Note] (V850/SC3 only)		Operating			Stopped
A/D converter		Operating			Stopped
DMA0 to DMA5		Operating			
Port function		Held			
External bus interface		Only bus hold function operates			

Note Available only for the μPD703089Y and 70F3089Y.

Table 4-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item		When CPU Operates on Main Clock		When CPU Operates on Subclock	
		When Subclock Does Not Exist	When Subclock Exists	When Main Clock Oscillation Continues	When Main Clock Oscillation Is Stopped
External interrupt request	NMI	Operating			
	INTP0 to INTP3, INTP7 to INTP9	Operating			
	INTP4 and INTP5	Operating			Stopped
	INTP6	Operates when main clock is selected for noise eliminator	Operating		Operates when f_{XT} is selected for noise eliminator
Key return function		Operating			
★ In external expansion mode	AD0 to AD15	High impedance[Note 1]			
	A1 to A15[Note 2]	Held[Note 1] (high impedance when $\overline{\text{HLDAK}}$ = 0)			
	A16 to A21	Held[Note 1] (high impedance when $\overline{\text{HLDAK}}$ = 0)			
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	Held[Note 1] (high impedance when $\overline{\text{HLDAK}}$ = 0)			
	R/$\overline{\text{W}}$	High-level output[Note 1] (high impedance when $\overline{\text{HLDAK}}$ = 0)			
	$\overline{\text{DSTB}}$, $\overline{\text{WRL}}$[Note 2], $\overline{\text{WRH}}$[Note 2], $\overline{\text{RD}}$[Note 2]				
	ASTB				
	$\overline{\text{HLDAK}}$	Operating			

Notes 1. Even when the HALT instruction has been executed, the instruction fetch operation continues until the on-chip instruction prefetch queue becomes full. Once it is full, operation stops in the state shown in Table 4-1.

2. Only for the V850/SC1 and V850/SC2

4.4.3 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 canceled, there is no need for the oscillator to wait for the oscillation stabilization time, so normal operation can be resumed quickly.

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

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

The operating statuses in IDLE mode are listed in Table 4-2.

(2) Cancellation of IDLE mode

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

Table 4-2. Operating Statuses in IDLE Mode (1/2)

IDLE Mode Settings / Item		When Subclock Exists	When Subclock Does Not Exist
CPU		Stopped	
ROM correction		Stopped	
Clock generator		Both main clock and subclock oscillating Clock supply to CPU and on-chip peripheral functions is stopped	
16-bit timer (TM0)		Operates when INTWTNI is selected as the count clock (f_{XT} is selected for watch timer)	Stopped
16-bit timer (TM1)		Stopped	
16-bit timer (TM5)		Operates when f_{XT} is selected for the count clock	Stopped
16-bit timer (TM6)		Stopped	
16-bit timer (TM7 to TM12)		Stopped	
Watch timer		Operating	
Watchdog timer		Stopped	
Serial interfaces	CSI0, CSI2 to CSI6	Operates when an external clock is selected as the serial clock	
	I^2C0, I^2C1	Stopped	
	UART0 to UART3	Operates only for transmission when an external clock is selected as the baud rate clock	
IEBus (V850/SC2 only)		Stopped	
FCAN1, FCAN2[Note] (V850/SC3 only)		Stopped	
A/D converter		Stopped	
DMA0 to DMA5		Stopped	
Port function		Held	

Note Available only for the μPD703089Y, 70F3089Y.

Table 4-2. Operating Statuses in IDLE Mode (2/2)

IDLE Mode Settings / Item		When Subclock Exists	When Subclock Does Not Exist
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3, INTP7 to INTP9	Operating	
	INTP4 and INTP5	Stopped	
	INTP6	Operates when f_{XT} is selected for the sampling clock	Stopped
Key return function		Operating	
In external expansion mode	AD0 to AD15	High impedance	
	A1 to A15[Note]	Held	
	A16 to A21	High impedance	
	$\overline{LBEN}$, $\overline{UBEN}$		
	R/$\overline{W}$		
	$\overline{DSTB}$, $\overline{WRL}$[Note], $\overline{WRH}$[Note], $\overline{RD}$[Note]		
	ASTB		
	$\overline{HLDAK}$		

Note Only for the V850/SC1 and V850/SC2

4.4.4 Software STOP mode

(1) Settings and operating states

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

When the subclock oscillator is not used, if the FRC bit in the processor clock control register (PCC) is set (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 are also stopped (but peripheral functions operating on the subclock are not stopped). The external bus hold request ($\overline{\text{HLDRQ}}$) is 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 4-3.

Caution To reduce the current consumption in software STOP mode, make the FCAN default settings as follows regardless of whether FCAN is used or not.

<1> Set the GOM bit of the CGST register to 1 (set GOM = 1, clear GOM = 0).
<2> Set bits SMNO1 and SMNO0 of the CGMSS register to 01.
<3> Set the GOM bit of the CGST register to 0 (set GOM = 0, clear GOM = 1).

For details of FCAN settings, refer to CHAPTER 19 FCAN CONTROLLER.

(2) Cancellation of software STOP mode

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

One oscillation stabilization time is secured when the STOP mode is canceled.

Table 4-3. Operating Statuses in Software STOP Mode (1/2)

Item \ 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 the count clock (f_{XT} is selected as the count clock for the watch timer)	Stopped
16-bit timer (TM1)		Stopped	
16-bit timer (TM5)		Operates when f_{XT} is selected for the count clock	Stopped
16-bit timer (TM6)		Stopped	
16-bit timer (TM7 to TM12)		Stopped	
Watch timer		Operates when f_{XT} is selected for the count clock	Stopped (operation disabled)
Watchdog timer		Stopped	
Serial interfaces	CSI0, CSI2 to CSI6	Operates when an external clock is selected as the serial clock	
	I^2C0, I^2C1	Stopped	
★	UART0 to UART3	Operates only for transmission when an external clock is selected as the baud rate clock	
IEBus (V850/SC2 only)		Stopped	
FCAN1, FCAN2[Note] (V850/SC3 only)		Stopped	
A/D converter		Stopped	

Note Available only for the μPD703089Y and 70F3089Y.

Table 4-3. Operating Statuses in Software STOP Mode (2/2)

Item \ Mode Settings		When Subclock Exists	When Subclock Does Not Exist
DMA0 to DMA5		Stopped	
Port function		Held	
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3, INTP7 to INTP9	Operating	
	INTP4 and INTP5	Stopped	
	INTP6	Operates when f_{XT} is selected for the sampling clock	Stopped
Key return function		Operating	
In external expansion mode	AD0 to AD15	High impedance	
	A1 to A15[Note]	Held	
	A16 to A21	High impedance	
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$		
	R/$\overline{\text{W}}$		
	$\overline{\text{DSTB}}$, $\overline{\text{WRL}}$[Note], $\overline{\text{WRH}}$[Note], $\overline{\text{RD}}$[Note]		
	ASTB		
	$\overline{\text{HLDAK}}$		

Note Only for the V850/SC1 and V850/SC2

4.5 Oscillation Stabilization Time

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

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

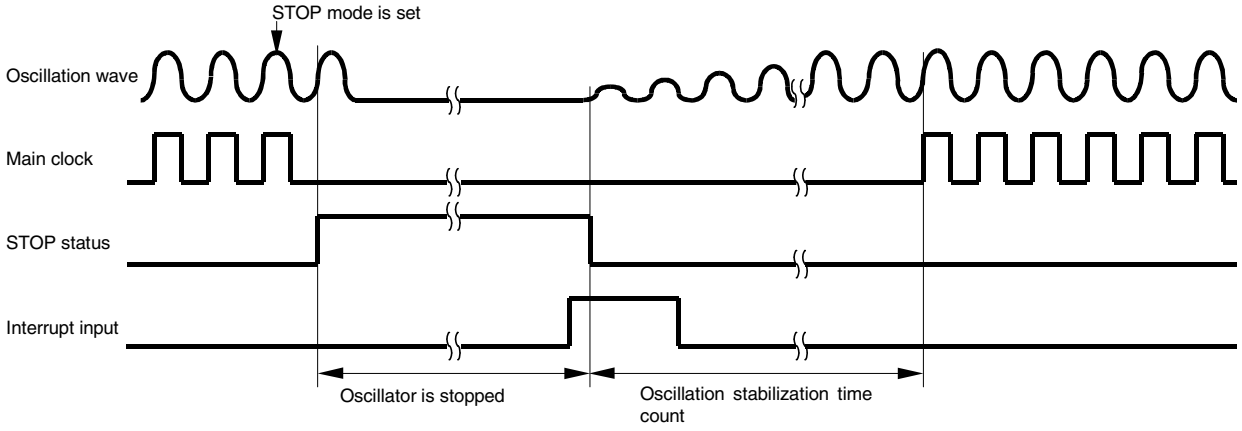
STOP mode is canceled by a non-maskable interrupt or an unmasked maskable interrupt request. When an interrupt is input, the counter (watchdog timer) starts counting and the count time is the length of time that must elapse until the oscillator's clock output stabilizes.

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 4-2. Oscillation Stabilization Time



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

For allocating time with $\overline{\text{RESET}}$ pin, refer to **CHAPTER 14 RESET FUNCTION**.

The oscillation stabilization time is 2^{18}/fxx, in accordance with the value of the OSTS register after reset.

4.6 Cautions on Power Save Function

★ **(1) When executing an instruction on the internal ROM**

To set the power save mode (IDLE or STOP mode) during execution of an instruction on the internal ROM, NOP instructions must be inserted as dummy instructions to execute the routine after the power save mode is released.

The sequence for setting the power save mode is as follows.

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

<5> Enable interrupts (clear NP bit of PSW to 0).
<6> Insert NOP instructions (two or five instructions).
<7> If DMA operation is needed, enable DMA operation.

Cautions 1. Insert two NOP instructions if the value of the ID bit of the PSW is not changed by executing the instruction that clears the NP bit to 0 (<5>), and if changed, insert five NOP instructions (<6>).
The following shows an example of description.

[Description example]

```
LDSR   rX,5              ;NP bit = 1
ST.B   r0,PRCMD[r0]      ;write to PRCMD
ST.B   rD,PSC[r0]        ;set PSC register
LDSR   rY,5              ;NP bit = 0
NOP                      ;Dummy instructions(2 or 5 instructions)
        ⋮
NOP
(next instruction)       ;execution routine after IDLE/STOP mode released
        ⋮
```

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.

To save the PSW value, transfer the PSW value before setting the NP bit to the rY register.

2. The instructions (<5> enable interrupt, <6> NOP instruction) following the store instruction (<4>) for the PSC register that is used to set IDLE mode or STOP mode are executed before the power save mode is entered.

(2) When executing an instruction on the external ROM

If the V850/SC1, V850/SC2, and V850/SC3 are used under the following conditions, a discrepancy occurs between the address indicated by the program counter (PC) and the address at which an instruction is actually read after the power save mode is released.

This may result in the CPU ignoring a 4- or 8-byte instruction from between 4 bytes and 16 bytes after an instruction is executed to write to the PSC register, which could in turn result in the execution of an erroneous instruction.

★ **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) Setting of power save mode (IDLE mode or STOP mode) while an instruction is being executed on external ROM

(ii) Cancellation of power save mode as the result of an interrupt request

(iii) Execution of the next instruction when an interrupt request is held pending following cancellation of the power save mode

Conditions for interrupt request to be held pending:

- When NP flag of PSW register is “1” (NMI servicing in progress/set by software)
- When ID flag of PSW register is “1” (interrupt request servicing in progress/DI instruction/set by software)
- When an interrupt enable (EI) state occurs during interrupt request servicing, but this state is cleared by an interrupt request with the same or lower priority

Therefore, use the V850/SC1, V850/SC2, and V850/SC3 under the following conditions.

[Usage Conditions]

(i) Do not use a power save mode (IDLE mode or STOP mode) during instruction execution on external ROM.

(ii) If it is necessary to use a power save mode during instruction execution on external ROM, implement the following software measures.

- Insert 6 NOP instructions 4 bytes after an instruction that writes to the PSC register.
- After the NOP instructions, insert a BR$+2 instruction to cancel the PC discrepancy.

[Workaround program example]

```
LDSR  rX,5              ;Sets rX value to PSW
ST.B  r0,PRCMD[r0]      ;Writes to PRCMD
ST.B  rD,PSC[r0]        ;Sets PSC register
LDSR  rY,5              ;Returns PSW value
NOP                     ;6 or more NOP instructions
NOP
NOP
NOP
NOP
NOP
BR    $+2               ;Cancels PC discrepancy
```

Remark It is assumed that rD (PSC setting value), rX (value written to PSW), and rY (value written back to PSW) have been set.

CHAPTER 5 PORT FUNCTIONS

5.1 Port Configuration

The V850/SC1, V850/SC2, and V850/SC3 include 124 I/O port pins from ports 0 to 15 and 17 (12 port pins are input only).

There are six pin I/O buffer power supplies; $PORTV_{DD0}$ to $PORTV_{DD2}$, V_{DD0}, V_{DD1}, and $ADCV_{DD}$, which are described below.

★ **Table 5-1. Pin I/O Buffer Power Supplies**

(a) μPD70F3089Y

Power Supply	Corresponding Pins	Usable Voltage Range
$PORTV_{DD0}$ Note 1	P40 to P47, P50 to P57, P60 to P65, P90 to P96, CLKOUT	3.0 V ≤ $PORTV_{DD0}$ ≤ 5.5 V
$PORTV_{DD1}$ Note 1	P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117	3.0 V ≤ $PORTV_{DD1}$ ≤ 5.5 V Note 2
$PORTV_{DD2}$ Note 1	P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157	3.0 V ≤ $PORTV_{DD2}$ ≤ 5.5 V Note 2
V_{DD0}	$\overline{\text{RESET}}$	When A/D converter not used: 4.0 V ≤ V_{DD0} ≤ 5.5 V When A/D converter used: 4.5 V ≤ V_{DD0} = $ADCV_{DD}$ ≤ 5.5 V
V_{DD1}	P170 to P176	4.0 V ≤ V_{DD1} ≤ 5.5 V
$ADCV_{DD}$	P70 to P77, P80 to P83	When A/D converter not used: 4.0 V ≤ $ADCV_{DD}$ ≤ 5.5 V When A/D converter used: 4.5 V ≤ V_{DD0} = $ADCV_{DD}$ ≤ 5.5 V

Notes 1. The electrical specifications differ between an operating frequency of 4 to 17 MHz and an operating frequency of 4 to 20 MHz.

2. When the FCAN controller is used: $PORTV_{DD1}$ ≤ $PORTV_{DD2}$
(Due to the supply voltage conditions of the in-circuit emulator)

Caution The conditions for the power supplies are as follows.

$PORTV_{DD0}$ ≤ $PORTV_{DD1}$ ≤ $PORTV_{DD2}$ ≤ V_{DD0} = V_{DD1} = $ADCV_{DD}$

(b) Other than μPD70F3089Y

Power Supply	Corresponding Pins	Usable Voltage Range
$PORTV_{DD0}$[Note 1]	P40 to P47, P50 to P57, P60 to P65, P90 to P96, CLKOUT	$3.0\ V \le PORTV_{DD0} \le 5.5\ V$
$PORTV_{DD1}$[Note 1]	P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117	$3.0\ V \le PORTV_{DD1} \le 5.5\ V$[Note 2]
$PORTV_{DD2}$[Note 1]	P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157	$3.0\ V \le PORTV_{DD2} \le 5.5\ V$[Note 2]
V_{DD0}	$\overline{\text{RESET}}$	When A/D converter not used: $3.5\ V \le V_{DD0} \le 5.5\ V$ When A/D converter used: $4.5\ V \le V_{DD0} = ADCV_{DD} \le 5.5\ V$
V_{DD1}	P170 to P176	$3.5\ V \le V_{DD1} \le 5.5\ V$
$ADCV_{DD}$	P70 to P77, P80 to P83	When A/D converter not used: $3.5\ V \le ADCV_{DD} \le 5.5\ V$ When A/D converter used: $4.5\ V \le V_{DD0} = ADCV_{DD} \le 5.5\ V$

Notes 1. Electrical specifications differ between the cases with operating frequency of 4 to 17 MHz and with operating frequency of 4 to 20 MHz.

2. When FCAN controller is used: $PORTV_{DD1} \le PORTV_{DD2}$
(Due to the supply voltage conditions of the in-circuit emulator)

Caution The condition for power supplies is as follows.

$$\mathbf{PORTV_{DD0} \le PORTV_{DD1} \le PORTV_{DD2} \le V_{DD0} = V_{DD1} = ADCV_{DD}}$$

5.2 Port Pin Functions

5.2.1 Port 0

Port 0 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

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, 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 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P0 is read, the pin levels at that time are read. Writing to P0 writes the values to that port. This does not affect the input pins.

In output mode: When P0 is read, the values of P0 are read. Writing to P0 writes the values to that port, and those values are immediately output.

Port 0 includes the following alternate functions.

Table 5-2. Port 0 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 0	P00	NMI	I/O	No	Analog noise elimination
	P01	INTP0			
	P02	INTP1			
	P03	INTP2			
	P04	INTP3			
	P05	INTP4/ADTRG			Digital noise elimination
	P06	INTP5			
	P07	INTP6			

Note Software pull-up function

(1) Function of P0 pins

Port 0 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 0 mode register (PM0).

In output mode, the values set to each bit are output to port 0 (P0). When using this port in output mode, either the valid edge of each interrupt request should be made invalid or each interrupt request should be masked (except for NMI requests).

When using this port in input mode, the pin statuses can be read by reading P0. Also, the values of P0 (output latch) can be read by reading P0 while in output mode.

The valid edges of NMI and INTP0 to INTP6 are specified via the rising edge specification register 0 (EGP0) and the falling edge specification register 0 (EGN0).

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 is provided that 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. Edge detection occurs only after the specified amount of time has elapsed.

(b) Elimination of noise from INTP4 to INTP6 and ADTRG 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, and ADTRG pins if the same level is not detected three times consecutively. The noise-elimination width can be changed for the INTP6 pin (see **7.3.8 (3) Noise elimination of INTP6 pin**).

Cautions 1. If the input pulse width is 2 or 3 clocks, whether it will be detected as a valid edge or eliminated as noise is undetermined.
To ensure correct detection of a valid edge, constant-level input is required for 3 clocks or more.
2. 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.
3. Noise elimination is not performed when these pins are used as an ordinary input port pins.

(3) Control registers

(a) Port 0 mode register (PM0)

PM0 can be read/written in 8- 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) Rising edge specification register 0 (EGP0)

EGP0 can be read/written in 8- 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 does not occur at rising edge
1	Interrupt request signal occurs at rising edge

Remark n = 0: Control of NMI pin
n = 1 to 7: Control of INTP0 to INTP6 pins

(c) Falling edge specification register 0 (EGN0)

EGN0 can be read/written in 8- or 1-bit units.

After reset: 00H R/W Address: FFFFF0C2H

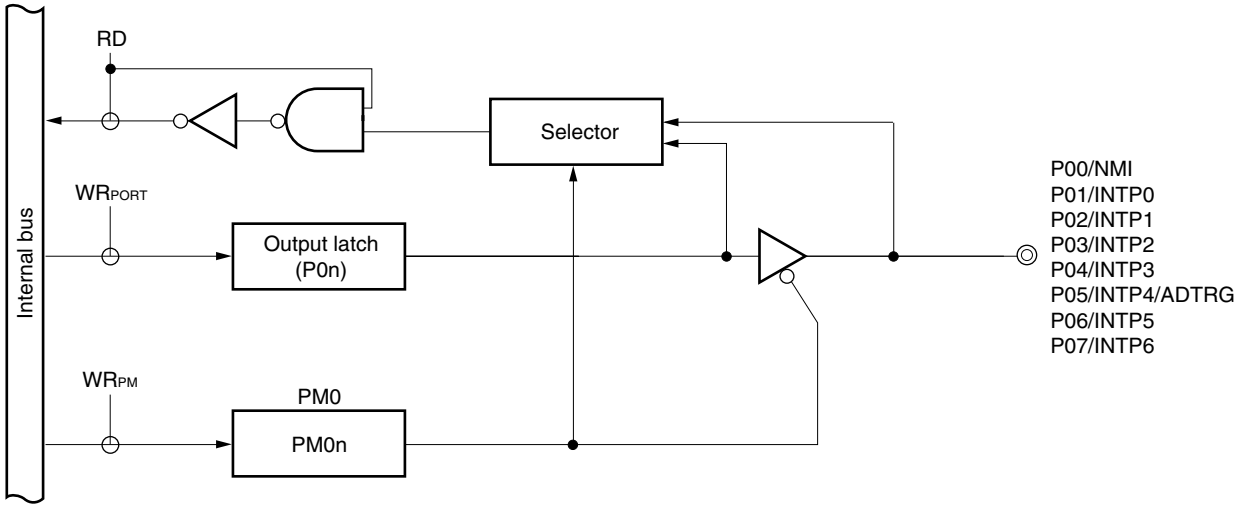
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
EGN0	EGN07	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00

EGN0n	Control of falling edge detection (n = 0 to 7)
0	Interrupt request signal does not occur at falling edge
1	Interrupt request signal occurs 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 5-1. Block Diagram of P00 to P07



Remarks 1. PM0: Port 0 mode register
RD: Port 0 read signal
WR: Port 0 write signal

2. n = 0 to 7

5.2.2 Port 1

Port 1 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

Bits 0 and 2 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	P17	P16	P15	P14	P13	P12	P11	P10

P1n	Control of output data (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P1 is read, the pin levels at that time are read. Writing to P1 writes the values to that port. This does not affect the input pins.

In output mode: When P1 is read, the values of P1 are read. Writing to P1 writes the values to that port, and those values are immediately output.

Port 1 includes the following alternate functions.

Table 5-3. Port 1 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 1	P10	SI0/SDA0	I/O	No	Selectable as N-ch open-drain output
	P11	SO0			–
	P12	$\overline{\text{SCK0}}$/SCL0			Selectable as N-ch open-drain output
	P13	SI4/RXD0			–
	P14	SO4/TXD0			
	P15	$\overline{\text{SCK4}}$/ASCK0			
	P16	–			
	P17	TI5/TO5			

Note Software pull-up function

(1) Function of P1 pins

Port 1 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 1 mode register (PM1).

In output mode, the values set to each bit are output to port 1 (P1). The port 1 function register (PF1) can be used to specify whether P10 and P12 are normal outputs or N-ch open-drain outputs.

When using this port in input mode, the pin statuses can be read by reading P1. Also, the values of P1 (output latch) can be read by reading P1 while in output mode.

Clear P1 and the PM1 register to 0 when using alternate-function pins as outputs. The logical sum (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- or 1-bit units.

After reset: FFH R/W Address: FFFFF022H

	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

(b) Port 1 function register (PF1)

PF1 can be read/written in 8- or 1-bit units.

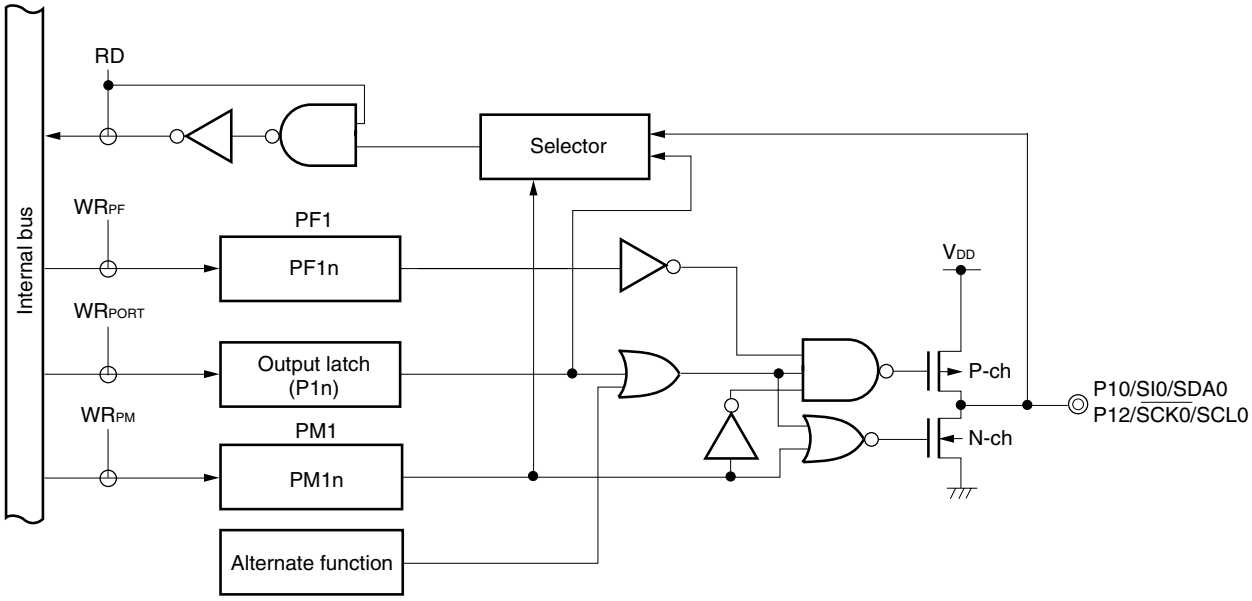
After reset: 00H R/W Address: FFFFF0A2H

	7	6	5	4	3	2	1	0
PF1	0	0	0	0	0	PF12	0	PF10

PF1n	Control of normal output/N-ch open-drain output (n = 0, 2)
0	Normal output
1	N-ch open-drain output

(3) Block diagram (port 1)

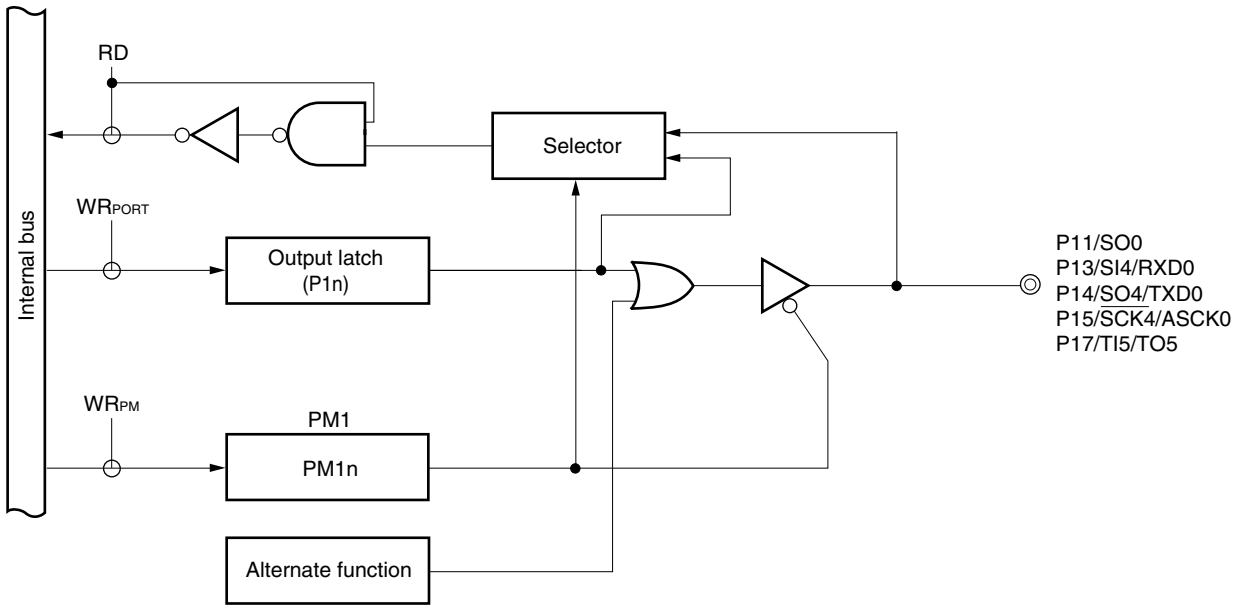
Figure 5-2. Block Diagram of P10 and P12



Remarks 1. PF1: Port 1 function register
PM1: Port 1 mode register
RD: Port 1 read signal
WR: Port 1 write signal

2. n = 0, 2

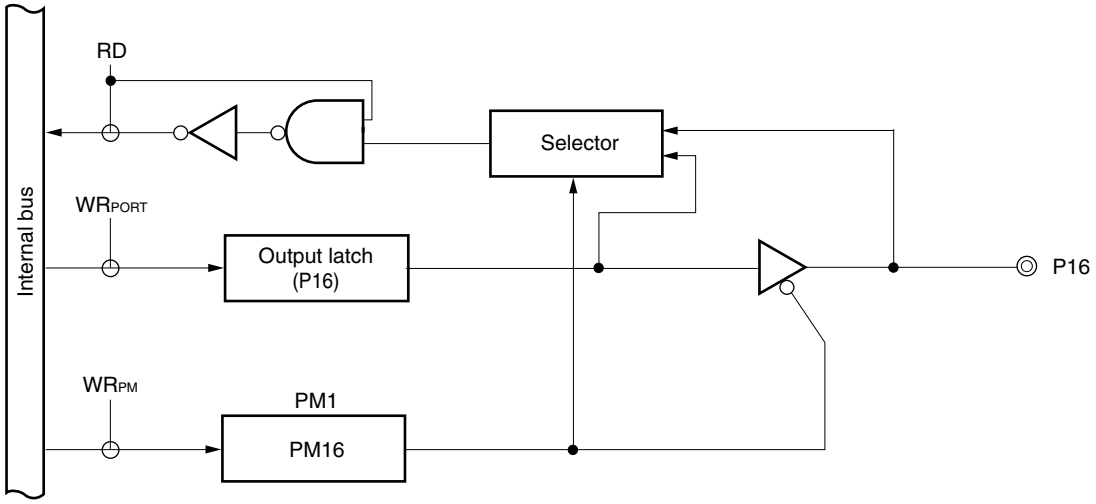
Figure 5-3. Block Diagram of P11, P13 to P15, and P17



Remarks 1. PM1: Port 1 mode register
RD: Port 1 read signal
WR: Port 1 write signal

2. n = 1, 3 to 5, 7

Figure 5-4. Block Diagram of P16



Remark PM1: Port 1 mode register
RD: Port 1 read signal
WR: Port 1 write signal

5.2.3 Port 2

Port 2 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

Bits 0 and 2 are selectable as normal outputs or N-ch open-drain outputs.

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 P2 is read, the pin levels at that time are read. Writing to P2 writes the values to that port. This does not affect the input pins.

In output mode: When P2 is read, the values of P2 are read. Writing to P2 writes the values to that port, and those values are immediately output.

Port 2 includes the following alternate functions.

Table 5-4. Port 2 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note 1]	Remark
Port 2	P20	SI2/SDA1	I/O	No	Selectable as N-ch open-drain output
	P21	SO2			–
	P22	$\overline{\text{SCK2}}$/SCK1			Selectable as N-ch open-drain output
	P23	TI90			–
	P24	TI91			
	P25	TO90			
	P26	$\overline{\text{IERX0}}$[Note 2]			
	P27	$\overline{\text{IETX0}}$[Note 2]			

Notes 1. Software pull-up function

2. Only for the V850/SC2

(1) Function of P2 pins

Port 2 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 2 mode register (PM2).

In output mode, the values set to each bit are output to port 2 (P2). Also, P20 and P22 are selectable as normal outputs or N-ch open-drain outputs using the port 2 function register (PF2).

When using this port in input mode, the pin statuses can be read by reading P2. Also, the values of P2 (output latch) can be read by reading P2 while in output mode.

Clear P2 and the PM2 register to 0 when using alternate-function pins as outputs. The logical sum (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- 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) Port 2 function register (PF2)

PF2 can be read/written in 8- or 1-bit units.

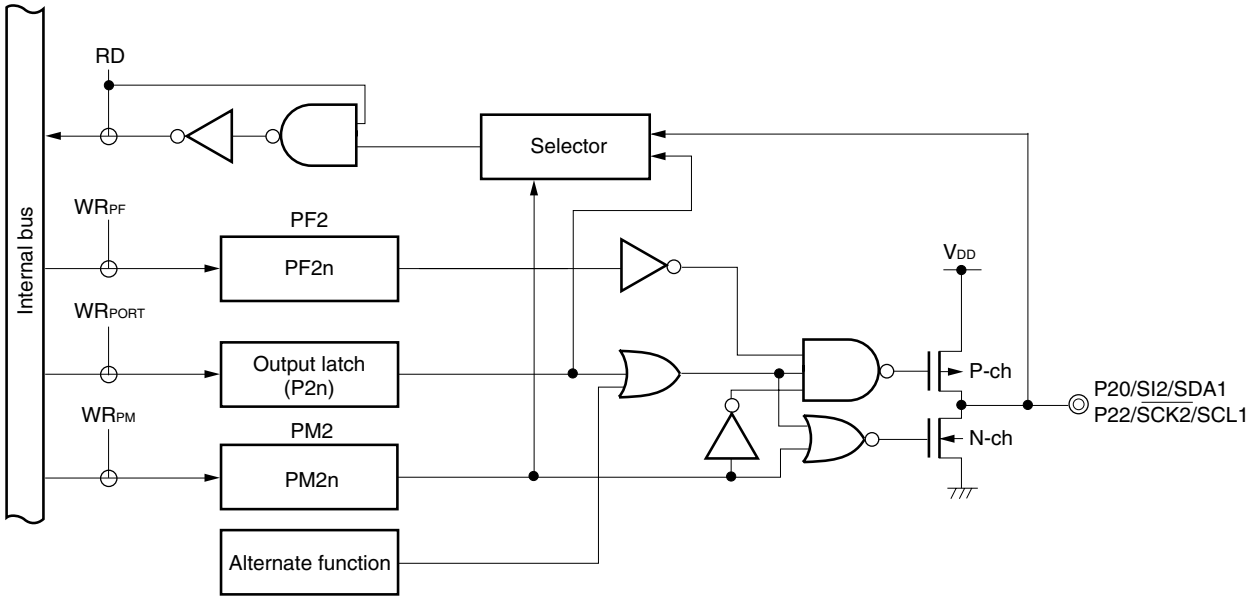
After reset: 00H R/W Address: FFFFF0A4H

	7	6	5	4	3	2	1	0
PF2	0	0	0	0	0	PF22	0	PF20

PF2n	Control of normal output/N-ch open-drain output (n = 0, 2)
0	Normal output
1	N-ch open-drain output

(3) Block diagram (port 2)

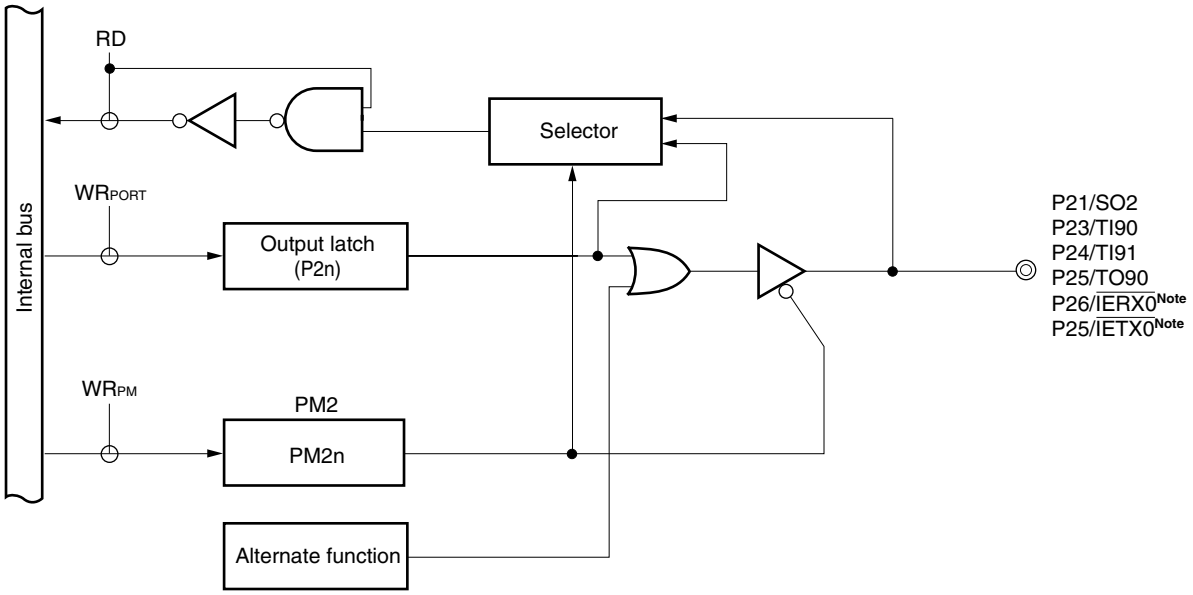
Figure 5-5. Block Diagram of P20 and P22



Remarks 1. PF2: Port 2 function register
PM2: Port 2 mode register
RD: Port 2 read signal
WR: Port 2 write signal

2. n = 0, 2

Figure 5-6. Block Diagram of P21 and P23 to P27



Note Only for the V850/SC2

Remarks 1. PM2: Port 2 mode register
RD: Port 2 read signal
WR: Port 2 write signal

2. n = 1, 3 to 7

5.2.4 Port 3

Port 3 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

When using P35 to P37 as the INTP7 to INTP9 pins, noise is eliminated by an analog 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	Outputs 0
1	Outputs 1

Remark In input mode: When P3 is read, the pin levels at that time are read. Writing to P3 writes the values to that port. This does not affect the input pins.

In output mode: When P3 is read, the values of P3 are read. Writing to P3 writes the values to that port, and those values are immediately output.

Port 3 includes the following alternate functions.

Table 5-5. Port 3 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note 1]	Remark
Port 3	P30	TI6/TO6	I/O	No	–
	P31	TI80			
	P32	TI81			
	P33	TO8			
	P34	TI71/A13[Note 2]			
	P35	INTP7/A14[Note 2]			Analog noise elimination
	P36	INTP8/A15[Note 2]			
	P37	INTP9			

Notes 1. Software pull-up function

2. Only for the V850/SC1 and V850/SC2

(1) Function of P3 pins

Port 3 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 3 mode register (PM3).

In output mode, the values set to each bit are output to port 3 (P3).

When using this port in input mode, the pin statuses can be read by reading P3. Also, the values of P3 (output latch) can be read by reading P3 while in output mode.

When using the alternate-function INTP7 to INTP9 pins, noise is eliminated by the analog noise eliminator (the same as the analog noise eliminator for port 0).

The valid edge for the INTP7 to INTP9 pins is specified by the rising edge specification register 1 (EGP1) and falling edge specification register 1 (EGN1).

When using the alternate-function A13 to A15 pins, set this port via the memory address output mode register (MAM).

Clear P3 and the PM3 register to 0 when using alternate-function pins as outputs. The logical sum (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. Also, the valid edge of each interrupt request becomes invalid (INTP7 to INTP9 do not function immediately after reset).

(2) Control registers

(a) Port 3 mode register (PM3)

PM3 can be read/written in 8- 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) Rising edge specification register 1 (EGP1)

EGP1 can be read/written in 8- or 1-bit units.

After reset: 00H R/W Address: FFFFF0C4H

	<7>	<6>	<5>	4	3	2	1	0
EGP1	EGP17	EGP16	EGP15	0	0	0	0	0

EGP1n	Control of rising edge detection (n = 5 to 7)
0	Interrupt request signal does not occur at rising edge
1	Interrupt request signal occurs at rising edge

Remark n = 5 to 7: Control of INTP7 to INTP9 pins

(c) Falling edge specification register 1 (EGN1)

EGN1 can be read/written in 8- or 1-bit units.

After reset: 00H R/W Address: FFFFF0C6H

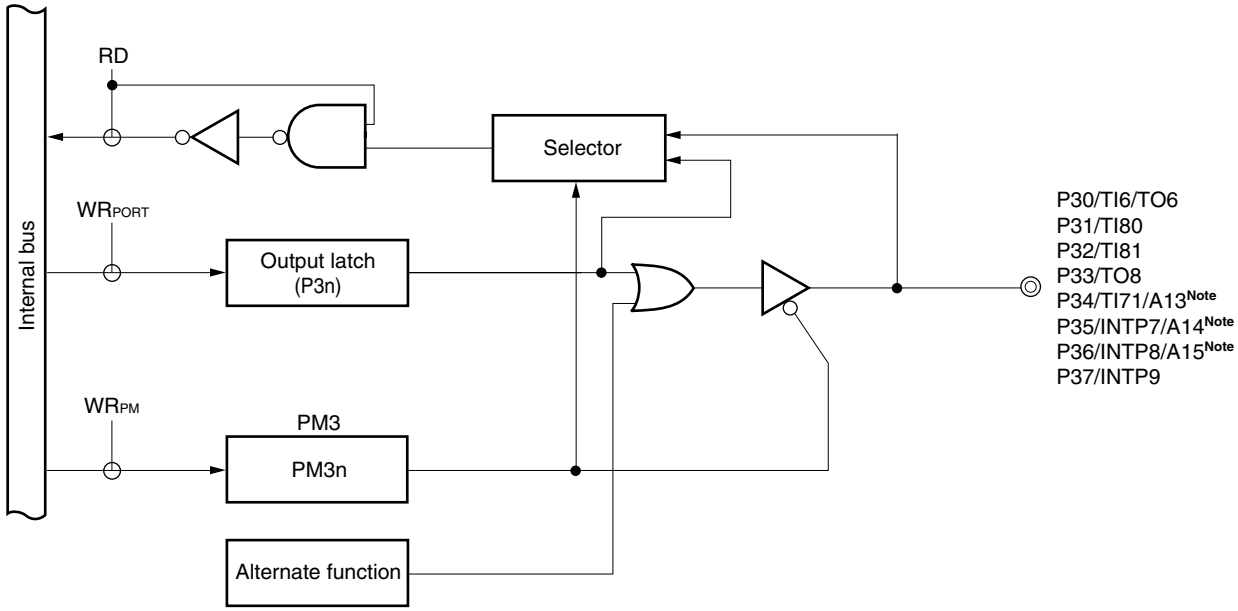
	<7>	<6>	<5>	4	3	2	1	0
EGN1	EGN17	EGN16	EGN15	0	0	0	0	0

EGN1n	Control of falling edge detection (n = 5 to 7)
0	Interrupt request signal does not occur at falling edge
1	Interrupt request signal occurs at falling edge

Remark n = 5 to 7: Control of INTP7 to INTP9 pins

(3) Block diagram (port 3)

Figure 5-7. Block Diagram of P30 to P37



Note Only for the V850/SC1 and V850/SC2

Remarks 1. PM3: Port 3 mode register
RD: Port 3 read signal
WR: Port 3 write signal

2. n = 0 to 7

5.2.5 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

(n = 4, 5)

Pnx	Control of output data (in output mode) (n = 4, 5, x = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P4 and P5 are read, the pin levels at that time are read. Writing to P4 and P5 writes the values to those ports. This does not affect the input pins.

In output mode: When P4 and P5 are read, their values are read. Writing to P4 and P5 writes the values to those ports, and those values are immediately output.

Ports 4 and 5 include the following alternate functions.

Table 5-6. 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 port 4 mode register (PM4) and port 5 mode register (PM5).

In output mode, the values set to each bit are output to port 4 and port 5 (P4 and P5).

When using these ports in input mode, the pin statuses can be read by reading P4 and P5. Also, the values of P4 and P5 (output latch) can be read by reading P4 and P5 while in output mode.

A software pull-up function is not implemented.

When using the alternate-function AD0 to AD15 pins, set the pin functions via the memory expansion mode register (MM). This does not affect the PM4 and PM5 registers.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 4 mode register and port 5 mode register (PM4 and PM5)

PM4 and PM5 can be read/written in 8- or 1-bit units.

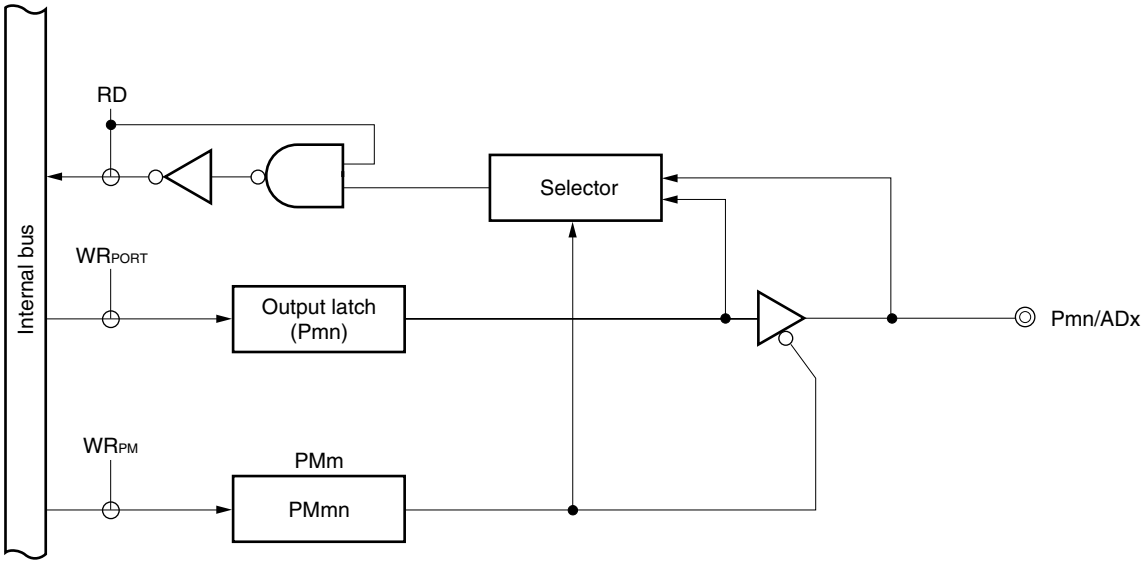
After reset: FFH R/W Address: FFFFF028H, FFFFF02AH

	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 5-8. 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

5.2.6 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 P6 is read, the pin levels at that time are read. Writing to P6 writes the values to that port. This does not affect the input pins.

In output mode: When P6 is read, the values of P6 are read. Writing to P6 writes the values to that port, and those values are immediately output.

Port 6 includes the following alternate functions.

Table 5-7. 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 port 6 (P6).

When using this port in input mode, the pin statuses can be read by reading P6. Also, the values of P6 (output latch) can be read by reading P6 while in output mode.

A software pull-up function is not implemented.

When using the alternate-function A16 to A21 pins, set the pin functions via the memory expansion mode register (MM). This does not affect 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- or 1-bit units.

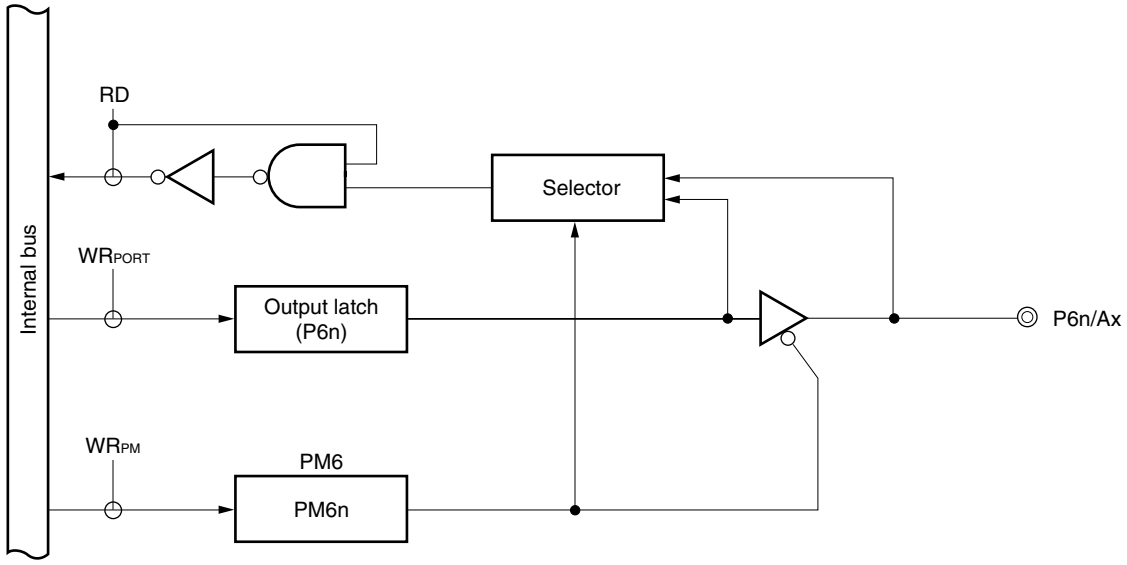
After reset: 3FH R/W Address: FFFFF02CH

	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 5-9. Block Diagram of 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

5.2.7 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- 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 5-8. 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.

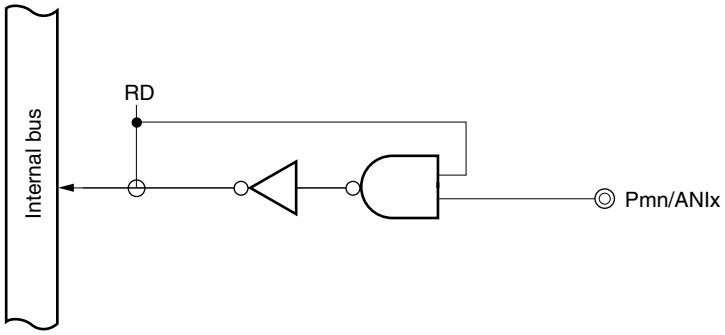
The pin statuses can be read by reading port 7 and port 8 (P7 and P8). Data cannot be written to P7 or P8.

A software pull-up function is not implemented.

Values read from pins specified as analog inputs are undefined values. Do not read values from P7 or P8 during A/D conversion.

(2) Block diagram (ports 7 and 8)

Figure 5-10. 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)

5.2.8 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	Outputs 0
1	Outputs 1

Remark In input mode: When P9 is read, the pin levels at that time are read. Writing to P9 writes the values to that port. This does not affect the input pins.

In output mode: When P9 is read, the values of P9 are read. Writing to P9 writes the values to that port, and those values are immediately output.

Port 9 includes the following alternate functions.

Table 5-9. Port 9 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note 1]	Remark
Port 9	P90	$\overline{\text{LBEN}}/\overline{\text{WRL}}$[Note 2]	I/O	No	–
	P91	$\overline{\text{UBEN}}$			
	P92	R/$\overline{\text{W}}$/$\overline{\text{WRH}}$[Note 2]			
	P93	$\overline{\text{DSTB}}/\overline{\text{RD}}$[Note 2]			
	P94	ASTB			
	P95	$\overline{\text{HLDAK}}$			
	P96	$\overline{\text{HLDRQ}}$			

Notes 1. Software pull-up function

2. Only for the V850/SC1 and V850/SC2

(1) Function of P9 pins

Port 9 is a 7-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 9 mode register (PM9).

In output mode, the values set to each bit are output to port 9 (P9).

When using this port in input mode, the pin statuses can be read by reading P9. Also, the values of P9 (output latch) can be read by reading P9 while in output mode.

A software pull-up function is not implemented.

When using 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 In the V850/SC1 and V850/SC2, 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.

(2) Control register

(a) Port 9 mode register (PM9)

PM9 can be read/written in 8- or 1-bit units.

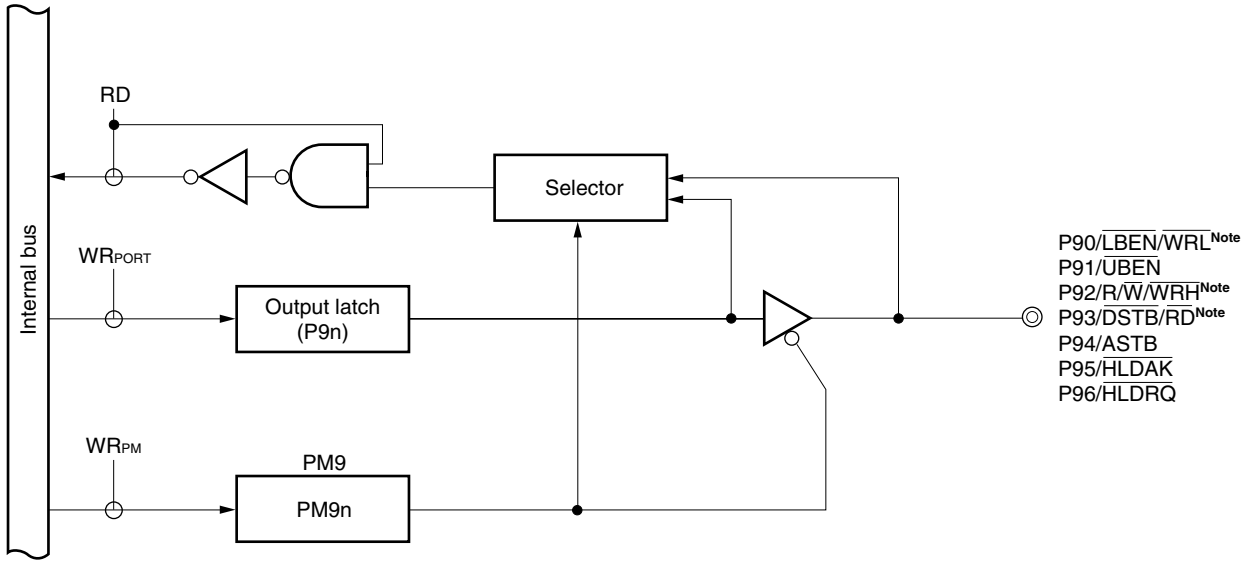
After reset: 7FH R/W Address: FFFFF032H

	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 5-11. Block Diagram of P90 to P96



Note Only for the V850/SC1 and V850/SC2

Remarks 1. PM9: Port 9 mode register
RD: Port 9 read signal
WR: Port 9 write signal

2. n = 0 to 6

5.2.9 Port 10

Port 10 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

Pull-up resistors can be connected in 1-bit units (software pull-up function).

When using P100 to P107 as the KR0 to KR7 pins, noise is eliminated by an 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	Outputs 0
1	Outputs 1

Remark In input mode: When P10 is read, the pin levels at that time are read. Writing to P10 writes the values to that port. This does not affect the input pins.

In output mode: When P10 is read, the values of P10 are read. Writing to P10 writes the values to that port, and those values are immediately output.

Port 10 includes the following alternate functions.

Table 5-10. Port 10 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note 1]	Remark
Port 10	P100	KR0/TO7/A5[Note 2]	I/O	Yes	Analog noise elimination
	P101	KR1/TI70/A6[Note 2]			
	P102	KR2/TI00/A7[Note 2]			
	P103	KR3/TI01/A8[Note 2]			
	P104	KR4/TO0/A9[Note 2]			
	P105	KR5/TI10/A10[Note 2]			
	P106	KR6/TI11/A11[Note 2]			
	P107	KR7/TO1/A12[Note 2]			

Notes 1. Software pull-up function

2. Only for the V850/SC1 and V850/SC2

(1) Function of P10 pins

Port 10 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 10 mode register (PM10).

In output mode, the values set to each bit are output to port 10 (P10).

When using this port in input mode, the pin statuses can be read by reading P10. Also, the values of P10 (output latch) can be read by reading P10 while in output mode.

Pull-up resistors can be connected in 1-bit units when specified via pull-up resistor option register 10 (PU10).

When used as KR0 to KR7 pins, noise is eliminated by the analog noise eliminator.

When used as the A5 to A12 pins in the case of the V850/SC1 and V850/ASC2, this port is set via the memory address output mode register (MAM).

Clear P10 and the PM10 register to 0 when using alternate-function pins as outputs. The logical sum (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 register

(a) Port 10 mode register (PM10)

PM10 can be read/written in 8- or 1-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- or 1-bit units.

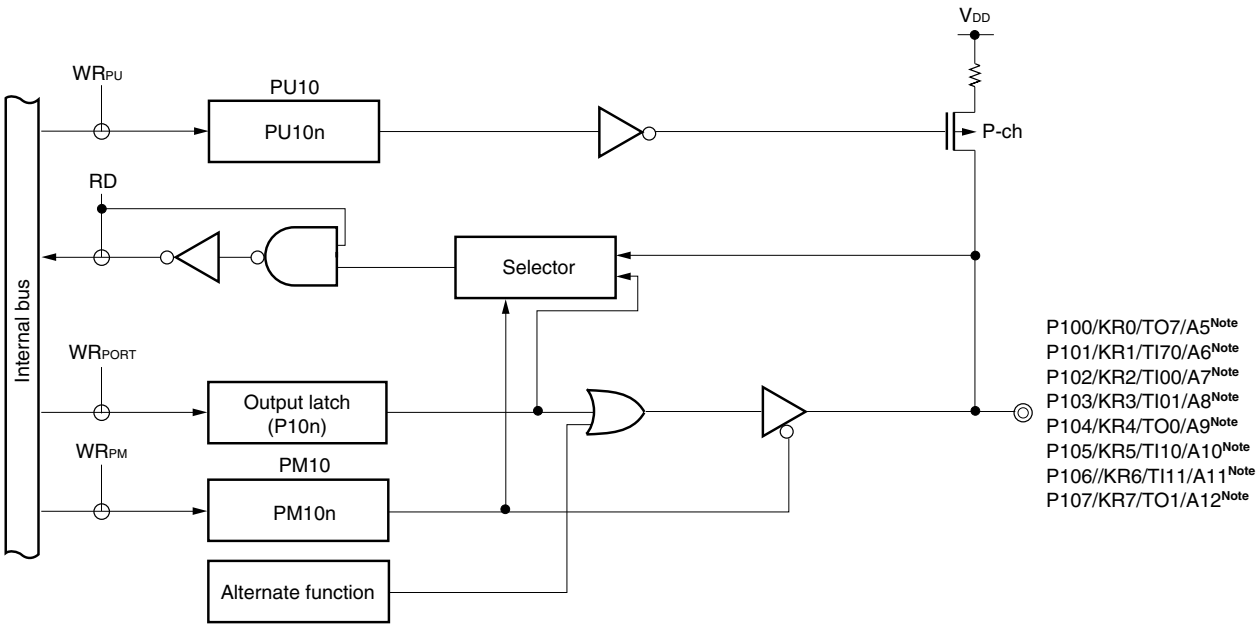
After reset: 00H R/W Address: FFFFF094H

	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

(3) Block diagram (port 10)

Figure 5-12. Block Diagram of P100 to P107



Note Only for the V850/SC1 and V850/SC2

Remarks 1. PU10: Pull-up resistor option register 10
PM10: Port 10 mode register
RD: Port 10 read signal
WR: Port 10 write signal

2. n = 0 to 7

5.2.10 Port 11

Port 11 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

P11 can be read/written in 8- or 1-bit units.

Turning on and off the wait function and setting use as the CANTX1, CANRX1, CANTX2, and CANRX2 pins can be performed via the port alternate-function control register (PAC) (CANTX2 and CANRX2 are available only for the μPD703089Y and 70F3089Y).

After reset: 00H R/W Address: FFFFF016H

	7	6	5	4	3	2	1	0
P11	P117	P116	P115	P114	P113	P112	P111	P110

P11n	Control of output data (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P11 is read, the pin levels at that time are read. Writing to P11 writes the values to that port. This does not affect the input pins.

In output mode: When P11 is read, the values of P11 are read. Writing to P11 writes the values to that port, and those values are immediately output.

Port 11 includes the following alternate functions.

Table 5-11. Port 11 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note 1]	Remark
Port 11	P110	$\overline{\text{WAIT}}$/A1[Note 2]	I/O	No	–
	P111	A2[Note 2]			
	P112	A3[Note 2]			
	P113	A4[Note 2]			
	P114	CANTX1[Note 3]			
	P115	CANRX1[Note 3]			
	P116	CANTX2[Note 4]			
	P117	CANRX2[Note 4]			

Notes 1. Software pull-up function

2. Only for the V850/SC1 and V850/SC2

3. Only for the V850/SC3

4. Only for the μPD703089Y and 70F3089Y

(1) Function of P11 pins

Port 11 is an 8-bit port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 11 mode register (PM11).

In output mode, the values set to each bit are output to port 11 (P11).

When using this port in input mode, the pin statuses can be read by reading P11. Also, the values of P11 (output latch) can be read by reading P11 while in output mode.

Turning on and off the wait function and setting use as the CANTX1, CANRX1, CANTX2, and CANRX2 pins can be performed via the port-alternate function control register (PAC).

When used as the A1 to A4 pins in the case of V850/SC1 and V850/SC2, this port is set via the memory address output mode register (MAM). In this case, be sure to set the PM11 register (PM110 to PM113) and P11 (P110 to P113) to 0.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 11 mode register (PM11)

PM11 can be read/written in 8- or 1-bit units.

After reset: FFH R/W Address: FFFFF036H

	7	6	5	4	3	2	1	0
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110

PM11n	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

(b) Port alternate-function control register (PAC)

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

After reset: 00H R/W Address: FFFFF040H

	7	6	5	4	3	2	1	<0>
PAC	PAC117[Note]	PAC116[Note]	PAC115	PAC114	0	0	0	WAC

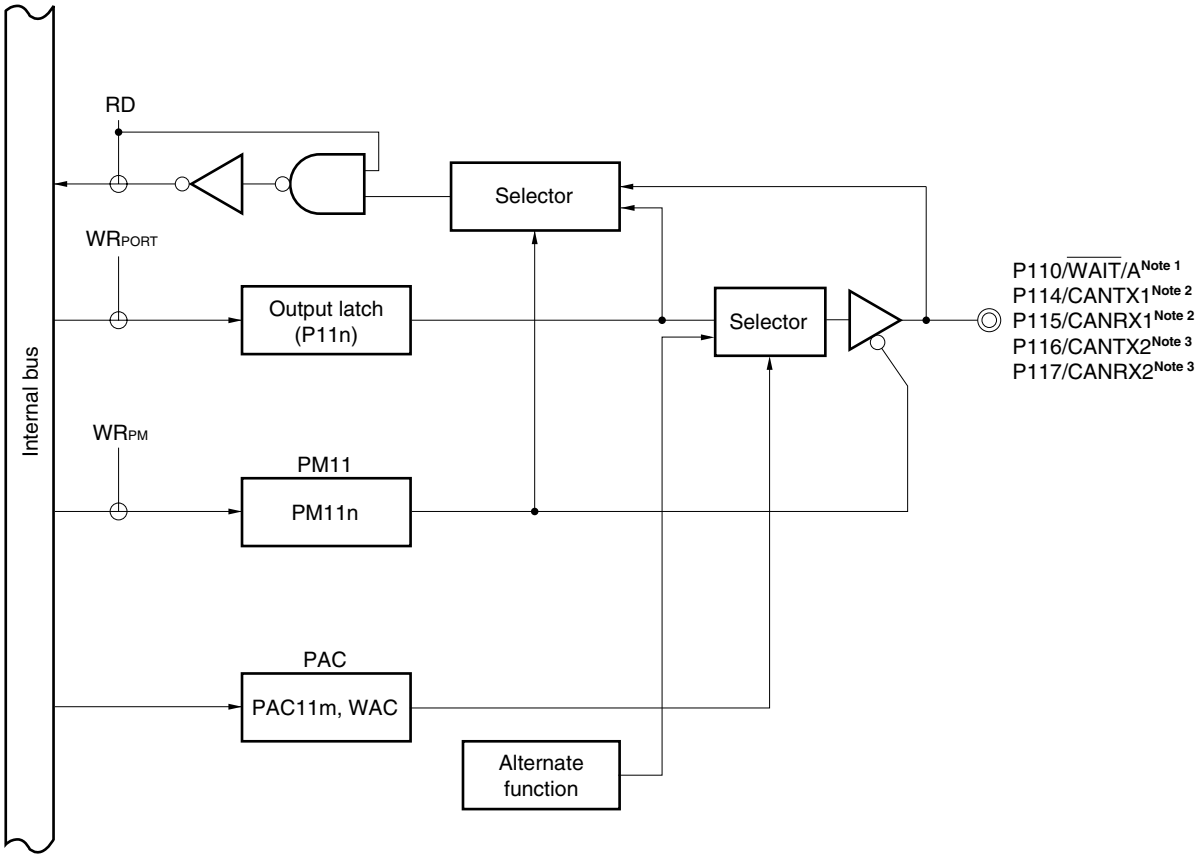
WAC	Control of wait function
0	Wait function OFF
1	Wait function ON

PAC11n	Control of port alternate function (n = 4 to 7)
0	Port function
1	Alternate function

Note Bits PAC117 and PAC116 are available only for the μPD703089Y and 70F3089Y. Set bits 7 and 6 to 0 when using the μPD703088Y.

(3) Block diagram (port 11)

Figure 5-13. Block Diagram of P110 and P114 to P117



Notes 1. Only for the V850/SC1 and V850/SC2

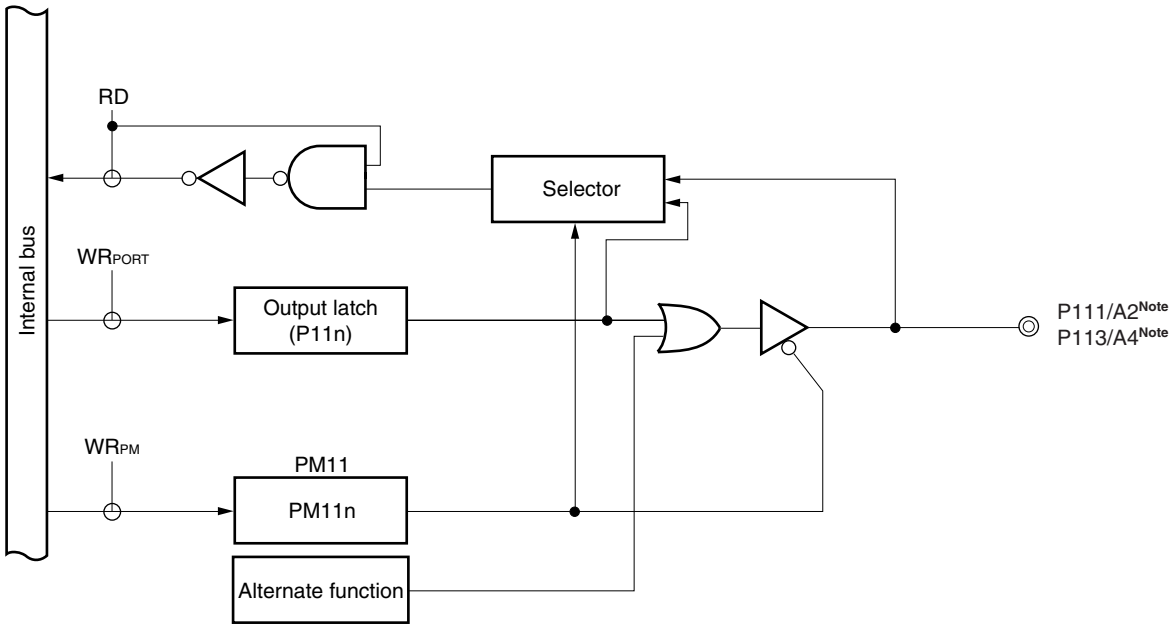
2. Only for the V850/SC3

3. Only for the μPD703089Y and 70F3089Y

Remarks 1. PM11: Port 11 mode register
RD: Port 11 read signal
WR: Port 11 write signal
PAC: Port alternate-function control register (PAC)

2. n = 0, 4 to 7
m = 4 to 7

Figure 5-14. Block Diagram of P111 to P113



Note Only for the V850/SC1 and V850/SC2

Remarks 1. PM11: Port 11 mode register
RD: Port 11 read signal
WR: Port 11 write signal

2. n = 1 to 3

5.2.11 Port 12

Port 12 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

P12 can be read/written in 8- or 1-bit units.

When using this port as the $\overline{SCK5}$, SI5, SO5, $\overline{SCK6}$, SI6, and SO6 pins, set via port alternate-function control register 2 (PAC2).

After reset: 00H R/W Address: FFFFF018H

	7	6	5	4	3	2	1	0
P12	P127	P126	P125	P124	P123	P122	P121	P120

P12n	Control of output data (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P12 is read, the pin levels at that time are read. Writing to P12 writes the values to that port. This does not affect the input pins.

In output mode: When P12 is read, the values of P12 are read. Writing to P12 writes the values to that port, and those values are immediately output.

Port 12 includes the following alternate functions.

Table 5-12. Port 12 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 12	P120	$\overline{SCK5}$	I/O	No	–
	P121	SI5			
	P122	SO5			
	P123	$\overline{SCK6}$			
	P124	SI6			
	P125	SO6			
	P126	TO10			
	P127	TO11			

Note Software pull-up function

(1) Function of P12 pins

Port 12 is an 8-bit port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 12 mode register (PM12).

In output mode, the values set to each bit are output to port 12 (P12).

When using this port in input mode, the pin statuses can be read by reading P12. Also, the values of P12 (output latch) can be read by reading P12 while in output mode.

When using this port as the $\overline{SCK5}$, SI5, SO5, $\overline{SCK6}$, SI6, and SO6 pins, set via port alternate-function control register 2 (PAC2).

When using this port as the TO10 and TO11 pins, set P126, P127, PM126, and PM127 to 0.

When a reset is input, the settings are initialized to input mode.

(2) Control registers

(a) Port 12 mode register (PM12)

PM12 can be read/written in 8- or 1-bit units.

After reset: FFH R/W Address: FFFFF038H

	7	6	5	4	3	2	1	0
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120

PM12n	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

(b) Port alternate-function control register 2 (PAC2)

PAC2 can be read/written in 8- or 1-bit units.

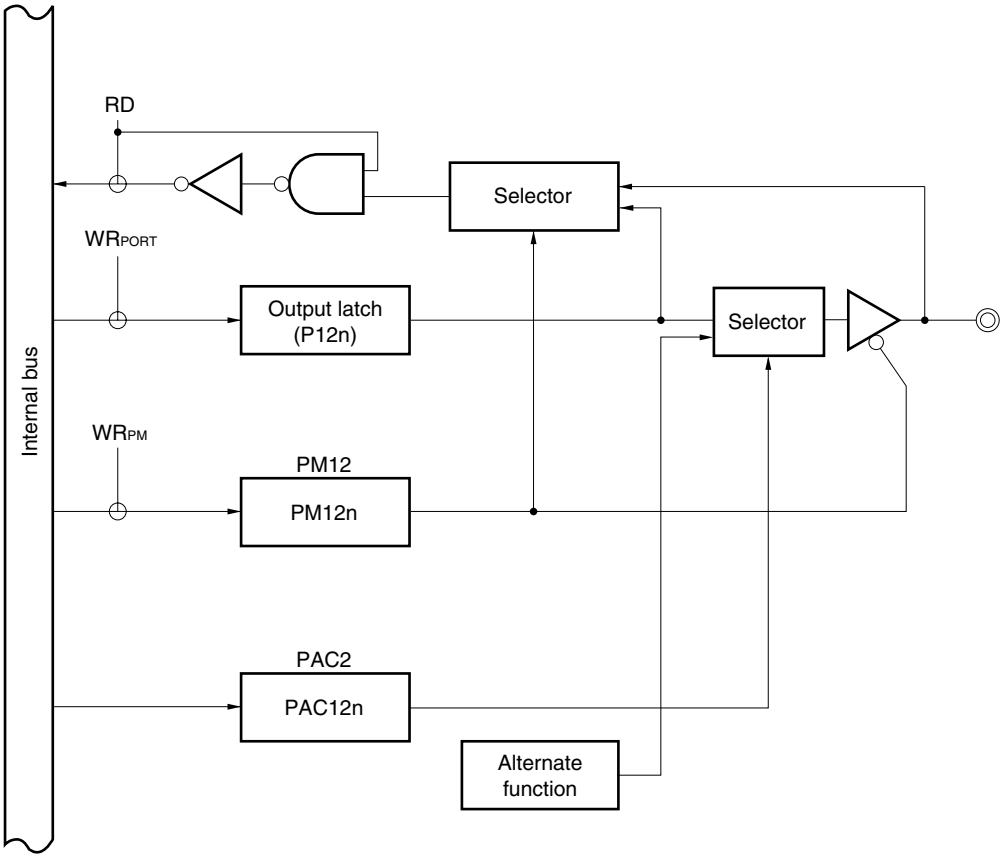
After reset: 00H R/W Address: FFFFF042H

	7	6	5	4	3	2	1	0
PAC2	0	0	PAC125	PAC124	PAC123	PAC122	PAC121	PAC120

PAC12n	Control of port alternate function (n = 0 to 5)
0	Port function
1	Alternate function

(3) Block diagram (port 12)

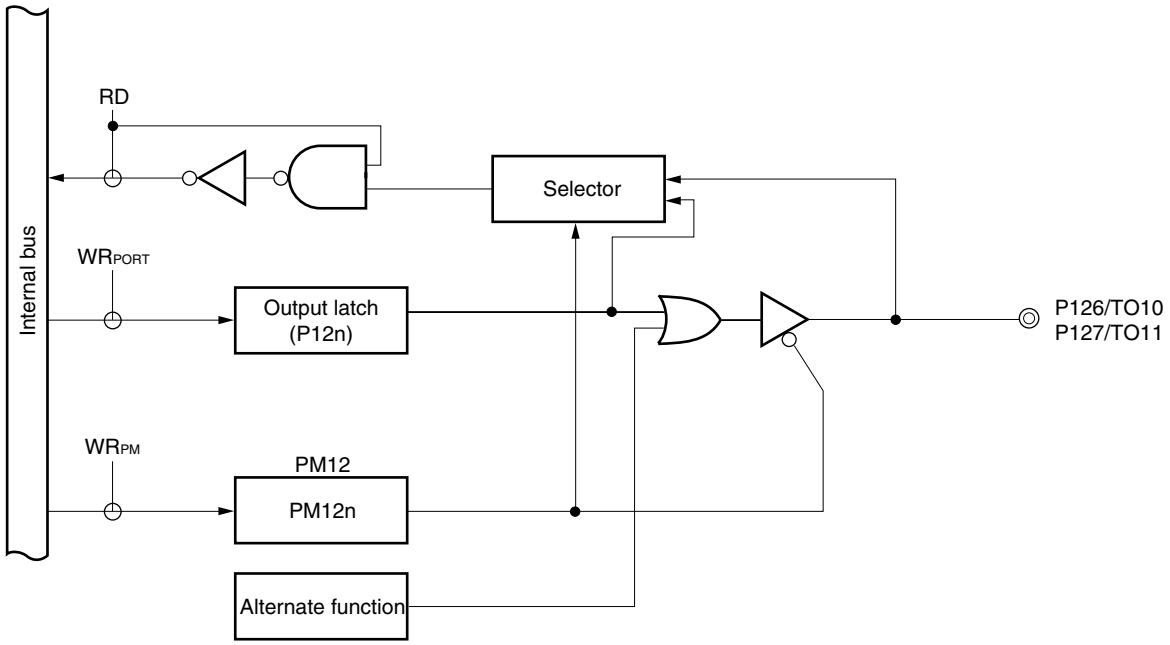
Figure 5-15. Block Diagram of P120 to P125



Remarks 1. PM12: Port 12 mode register
RD: Port 12 read signal
WR: Port 12 write signal
PAC2: Port alternate-function control register 2

2. n = 0 to 5

Figure 5-16. Block Diagram of P126 and P127



Remarks 1. PM12: Port 12 mode register
RD: Port 12 read signal
WR: Port 12 write signal

2. n = 6, 7

5.2.12 Port 13

Port 13 is a 4-bit I/O port for which I/O settings can be controlled in 1-bit units.

After reset: 00H R/W Address: FFFFF01AH

	7	6	5	4	3	2	1	0
P13	0	0	0	0	P133	P132	P131	P130

P13n	Control of output data (in output mode) (n = 0 to 3)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P13 is read, the pin levels at that time are read. Writing to P13 writes the values to that port. This does not affect the input pins.

In output mode: When P13 is read, the values of P13 are read. Writing to P13 writes the values to that port, and those values are immediately output.

(1) Function of P13 pins

Port 13 is a 4-bit port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 13 mode register (PM13).

In output mode, the values set to each bit are output to port 13 (P13).

When using this port in input mode, the pin statuses can be read by reading P13. Also, the values of P13 (output latch) can be read by reading P13 while in output mode.

When a reset is input, the settings are initialized to input mode.

(2) Control register

(a) Port 13 mode register (PM13)

PM13 can be read/written in 8- or 1-bit units.

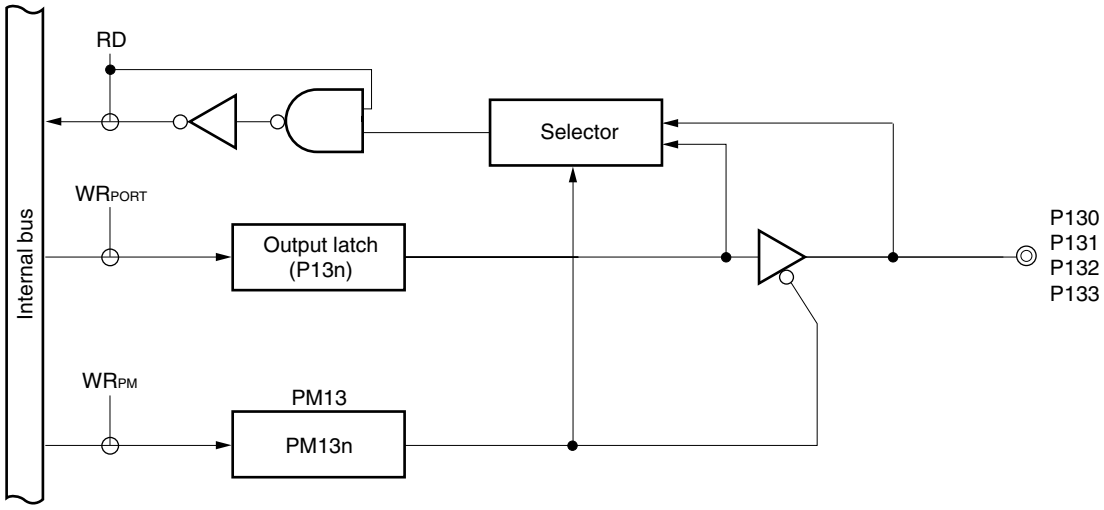
After reset: 0FH R/W Address: FFFFF03AH

	7	6	5	4	3	2	1	0
PM13	0	0	0	0	PM133	PM132	PM131	PM130

PM13n	Control of I/O mode (n = 0 to 3)
0	Output mode
1	Input mode

(3) Block diagram (port 13)

Figure 5-17. Block Diagram of P130 to P133



Remarks 1. PM13: Port 13 mode register
RD: Port 13 read signal
WR: Port 13 write signal

2. n = 0 to 3

5.2.13 Port 14

Port 14 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

After reset: 00H R/W Address: FFFFF01CH

	7	6	5	4	3	2	1	0
P14	P147	P146	P145	P144	P143	P142	P141	P140

P14n	Control of output data (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P14 is read, the pin levels at that time are read. Writing to P14 writes the values to that port. This does not affect the input pins.

In output mode: When P14 is read, the values of P14 are read. Writing to P14 writes the values to that port, and those values are immediately output.

Port 14 includes the following alternate functions.

Table 5-13. Port 14 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 14	P140	SI3/RXD1	I/O	No	–
	P141	SO3/TXD1			
	P142	$\overline{\text{SCK3}}$/ASCK1			
	P143	RXD2			
	P144	TXD2			
	P145	ASCK2			
	P146	TI100			
	P147	TI101			

Note Software pull-up function

(1) Function of P14 pins

Port 14 is an 8-bit port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 14 mode register (PM14).

In output mode, the values set to each bit are output to port 14 (P14).

When using this port in input mode, the pin statuses can be read by reading P14. Also, the values of P14 (output latch) can be read by reading P14 while in output mode.

Clear P14 and the PM14 register to 0 when using alternate-function pins as outputs. The logical sum (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 register

(a) Port 14 mode register (PM14)

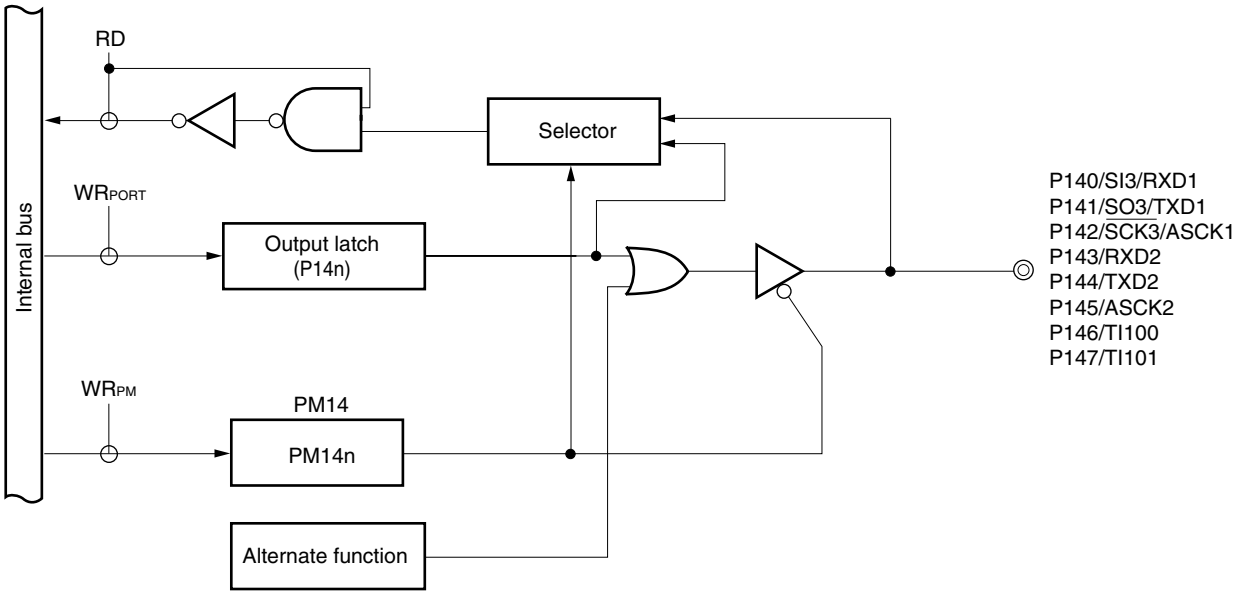
PM14 can be read/written in 8- or 1-bit units.

After reset: FFH R/W Address: FFFFF03CH

	7	6	5	4	3	2	1	0
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140

PM14n	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

Figure 5-18. Block Diagram of P140 to P147



Remarks 1. PM14: Port 14 mode register
RD: Port 14 read signal
WR: Port 14 write signal

2. n = 0 to 7

5.2.14 Port 15

Port 15 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.

After reset: 00H R/W Address: FFFFF01EH

	7	6	5	4	3	2	1	0
P15	P157	P156	P155	P154	P153	P152	P151	P150

P15n	Control of output data (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P15 is read, the pin levels at that time are read. Writing to P15 writes the values to that port. This does not affect the input pins.

In output mode: When P15 is read, the values of P15 are read. Writing to P15 writes the values to that port, and those values are immediately output.

Port 15 includes the following alternate functions.

Table 5-14. Port 15 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 15	P150	RXD3	I/O	No	–
	P151	TXD3			
	P152	ASCK3			
	P153	TI110			
	P154	TI111			
	P155	TO12			
	P156	TI120			
	P157	TI121			

Note Software pull-up function

(1) Function of P15 pins

Port 15 is an 8-bit port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 15 mode register (PM15).

In output mode, the values set to each bit are output to port 15 (P15).

When using this port in input mode, the pin statuses can be read by reading P15. Also, the values of P15 (output latch) can be read by reading P15 while in output mode.

Clear P15 and the PM15 register to 0 when using alternate-function pins as outputs. The logical sum (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 register

(a) Port 15 mode register (PM15)

PM15 can be read/written in 8- or 1-bit units.

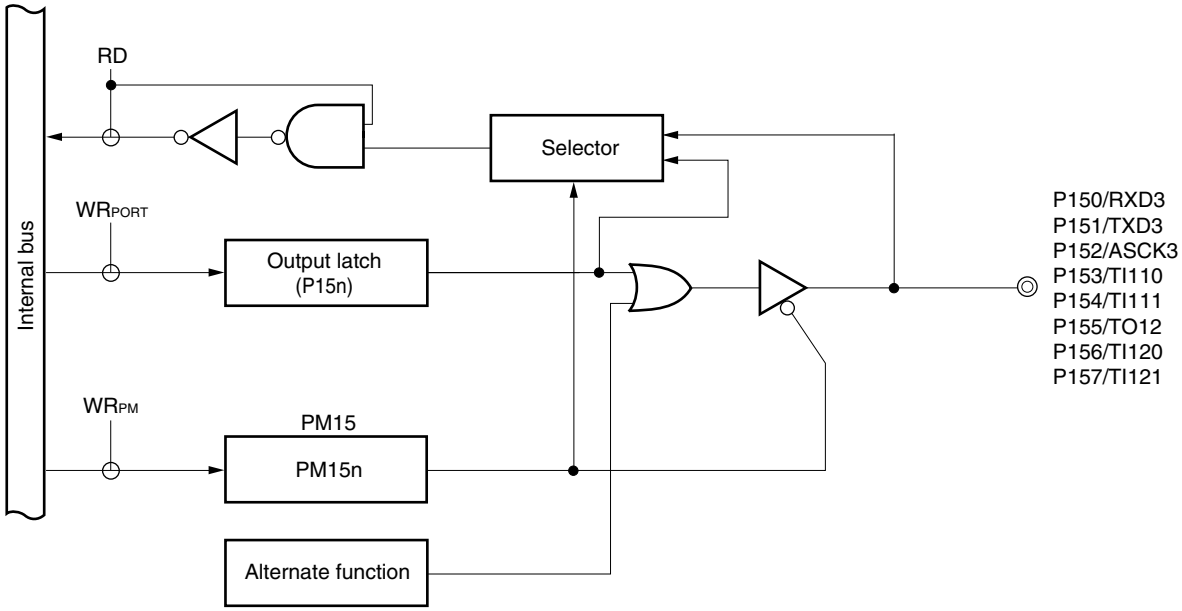
After reset: FFH R/W Address: FFFFF03EH

	7	6	5	4	3	2	1	0
PM15	PM157	PM156	PM155	PM154	PM153	PM152	PM151	PM150

PM15n	Control of I/O mode (n = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagram (port 15)

Figure 5-19. Block Diagram of P150 to P157



Remarks 1. PM15: Port 15 mode register
RD: Port 15 read signal
WR: Port 15 write signal

2. n = 0 to 7

5.2.15 Port 17

Port 17 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: FFFFF048H

	7	6	5	4	3	2	1	0
P17	0	P176	P175	P174	P173	P172	P171	P170

P17n	Control of output data (in output mode) (n = 0 to 6)
0	Outputs 0
1	Outputs 1

Remark In input mode: When P17 is read, the pin levels at that time are read. Writing to P17 writes the values to that port. This does not affect the input pins.

In output mode: When P17 is read, the values of P17 are read. Writing to P17 writes the values to that port, and those values are immediately output.

Port 17 includes the following alternate functions.

Table 5-15. Port 17 Alternate Function Pins

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 17	P170	–	I/O	No	–
	P171	–			
	P172	–			
	P173	–			
	P174	–			
	P175	–			
	P176	VM45			

Note Software pull-up function

(1) Function of P17 pins

Port 17 is a 7-bit port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 17 mode register (PM17).

In output mode, the values set to each bit are output to port 17 (P17).

When using this port in input mode, the pin statuses can be read by reading P17. Also, the values of P17 (output latch) can be read by reading P17 while in output mode.

When using this port as the VM45 pin, set via the VM45 control register (VM45C). In this case, be sure to set P176 and PM176 to 0.

When a reset is input, the settings are initialized to input mode.

(2) Control register

(a) Port 17 mode register (PM17)

PM17 can be read/written in 8- or 1-bit units.

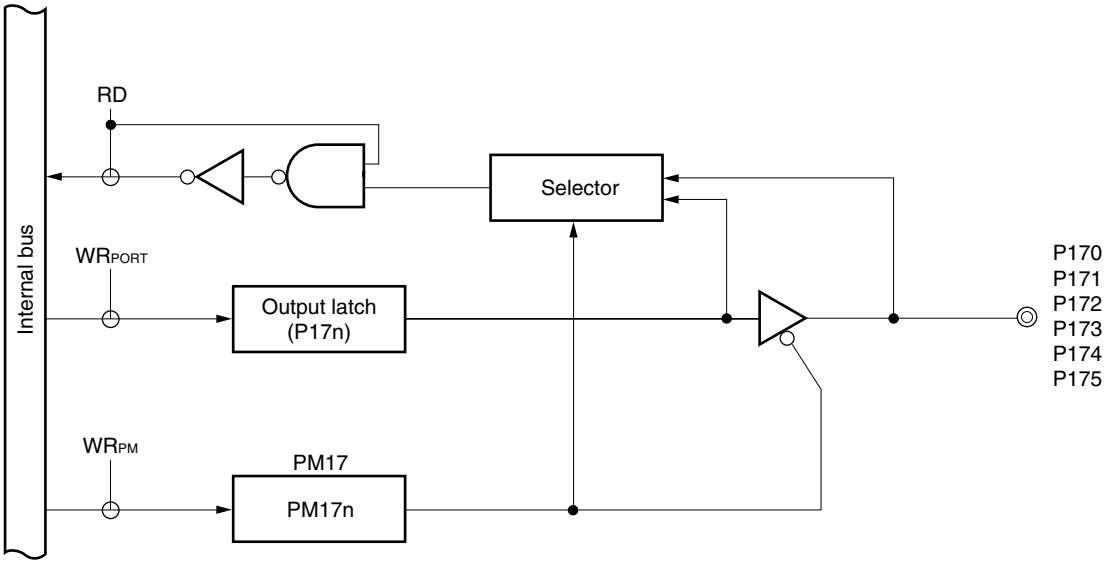
After reset: 7FH R/W Address: FFFFF058H

	7	6	5	4	3	2	1	0
PM17	0	PM176	PM175	PM174	PM173	PM172	PM171	PM170

PM17n	Control of I/O mode (n = 0 to 6)
0	Output mode
1	Input mode

(3) Block diagram (port 17)

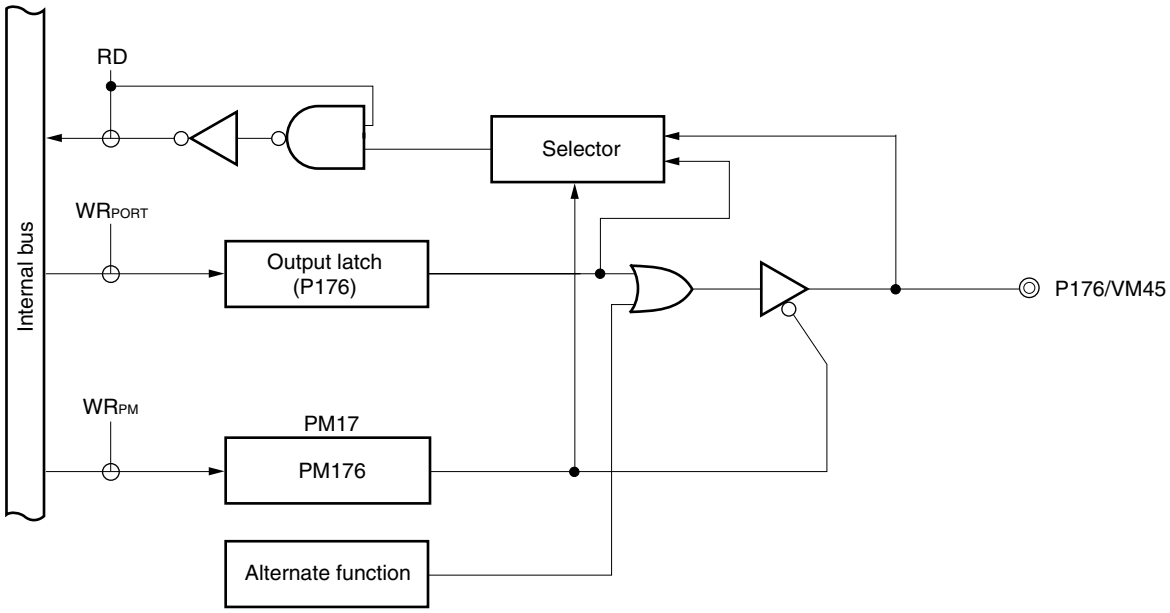
Figure 5-20. Block Diagram of P170 to P175



Remarks 1. PM17: Port 17 mode register
RD: Port 17 read signal
WR: Port 17 write signal

2. n = 0 to 5

Figure 5-21. Block Diagram of P176



Remark PM17: Port 17 mode register
RD: Port 17 read signal
WR: Port 17 write signal

5.3 Setting When Port Pin Is Used for Alternate Function

When a port pin is used for an alternate function, set the port n mode register (PM0 to PM6, PM9 to PM12, PM14, PM15, and PM17) and output latch as shown in Table 5-16 below.

Table 5-16. Setting When Port Pin Is Used for Alternate Function (1/6)

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	–
P07	INTP6	Input	PM07 = 1	Setting not needed for P07	–
P10	SI0	Input	PM10 = 1	Setting not needed for P10	–
★	SDA0	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	I/O			PF12 = 1
P13	SI4	Input	PM13 = 1	Setting not needed for P13	–
	RXD0	Input			
P14	SO4	Output	PM14 = 0	P14 = 0	–
	TXD0	Output			
P15	$\overline{\text{SCK4}}$	Input	PM15 = 1	Setting not needed for P15	–
		Output	PM15 = 0	P15 = 0	
	ASCK0	Input	PM15 = 1	Setting not needed for P15	
P17	TI5	Input	PM17 = 1	Setting not needed for P17	–
	TO5	Output	PM17 = 0	P17 = 0	

Table 5-16. Setting When Port Pin Is Used for Alternate Function (2/6)

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	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	I/O			PF22 = 1
P23	TI90	Input	PM23 = 1	Setting not needed for P23	–
P24	TI91	Input	PM24 = 1	Setting not needed for P24	–
P25	TO9	Output	PM25 = 0	P25 = 0	–
P26	$\overline{IERX0}$ Note 1	Input	PM26 = 1	Setting not needed for P26	–
P27	$\overline{IETX0}$ Note 1	Output	PM27 = 0	P27 = 0	–
P30	TI6	Input	PM30 = 1	Setting not needed for P30	–
	TO6	Output	PM30 = 0	P30 = 0	
P31	TI80	Input	PM31 = 1	Setting not needed for P31	–
P32	TI81	Input	PM32 = 1	Setting not needed for P32	–
P33	TO8	Output	PM33 = 0	P33 = 0	–
P34	TI71	Input	PM34 = 1	Setting not needed for P34	–
	A13 Note 2	Output	PM34 = 0	P34 = 0	Refer to **3.4.6 (2)** (MAM)
P35	INTP7	Input	PM35 = 1	Setting not needed for P35	–
	A14 Note 2	Output	PM35 = 0	P35 = 0	Refer to **3.4.6 (2)** (MAM)
P36	INTP8	Input	PM36 = 1	Setting not needed for P36	–
	A15 Note 2	Output	PM36 = 0	P36 = 0	Refer to **3.4.6 (2)** (MAM)
P37	INTP9	Input	PM37 = 1	Setting not needed for P37	–

Notes 1. Only for the V850/SC2

2. Only for the V850/SC1 and V850/SC2

Table 5-16. Setting When Port Pin Is Used for Alternate Function (3/6)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
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)
	$\overline{\text{WRL}}$[Note]	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)
	$\overline{\text{WRH}}$[Note]	Output			
★ P93	$\overline{\text{DSTB}}$	Output	Setting not needed for PM93	P93 = 1	Refer to **3.4.6 (1)** (MM)
	$\overline{\text{RD}}$[Note]	Output			
★ P94	ASTB	Output	Setting not needed for PM94	P94 = 1	Refer to **3.4.6 (1)** (MM)
★ P95	$\overline{\text{HLDAK}}$	Output	Setting not needed for PM95	Setting not needed for P95	Refer to **3.4.6 (1)** (MM)
★ P96	$\overline{\text{HLDRQ}}$	Input	Setting not needed for PM96	Setting not needed for P96	Refer to **3.4.6 (1)** (MM)
P100	KR0	Input	PM100 = 1	Setting not needed for P100	–
	TO7	Output	PM100 = 0	P100 = 0	
	A5[Note]	Output			Refer to **3.4.6 (2)** (MAM)
P101	KR1	Input	PM101 = 1	Setting not needed for P101	–
	TI70	Input			
	A6[Note]	Output	PM101 = 0	P101 = 0	Refer to **3.4.6 (2)** (MAM)
P102	KR2	Input	PM102 = 1	Setting not needed for P102	–
	TI00	Input			
	A7[Note]	Output	PM102 = 0	P102 = 0	Refer to **3.4.6 (2)** (MAM)

Note Only for the V850/SC1 and V850/SC2

Table 5-16. Setting When Port Pin Is Used for Alternate Function (4/6)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P103	KR3	Input	PM103 = 1	Setting not needed for P103	–
	TI01	Input	PM103 = 0	P103 = 0	
	A8[Note 1]	Output			Refer to **3.4.6 (2)** (MAM)
P104	KR4	Input	PM104 = 1	Setting not needed for P104	–
	TO0	Output	PM104 = 0	P104 = 0	
	A9[Note 1]	Output			Refer to **3.4.6 (2)** (MAM)
P105	KR5	Input	PM105 = 1	Setting not needed for P105	–
	TI10	Input			
	A10[Note 1]	Output	PM105 = 0	P105 = 0	Refer to **3.4.6 (2)** (MAM)
P106	KR6	Input	PM106 = 1	Setting not needed for P106	–
	TI11	Input			
	A11[Note 1]	Output	PM106 = 0	P106 = 0	Refer to **3.4.6 (2)** (MAM)
P107	KR7	Input	PM107 = 1	Setting not needed for P107	–
	TO1	Output	PM107 = 0	P107 = 0	
	A12[Note 1]	Output			Refer to **3.4.6 (2)** (MAM)
P110	$\overline{\text{WAIT}}$	Input	PM110 = 1	Setting not needed for P110	WAC = 1 (PAC)
	A1[Note 1]	Output	PM110 = 0	P110 = 0	Refer to **3.4.6 (2)** (MAM)
P111	A2[Note 1]	Output	PM111 = 0	P111 = 0	Refer to **3.4.6 (2)** (MAM)
P112	A3[Note 1]	Output	PM112 = 0	P112 = 0	Refer to **3.4.6 (2)** (MAM)
P113	A4[Note 1]	Output	PM113 = 0	P113 = 0	Refer to **3.4.6 (2)** (MAM)
P114	CANTX1[Note 2]	Output	PM114 = 0	P114 = 0	PAC114 = 1 (PAC)
P115	CANRX1[Note 2]	Input	PM115 = 1	P115 = 0	PAC115 = 1 (PAC)

Notes 1. Only for the V850/SC1 and V850/SC2

2. Only for the V850/SC3

Table 5-16. Setting When Port Pin Is Used for Alternate Function (5/6)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P116	CANTX2[Note]	Output	PM116 = 0	P116 = 0	PAC116 = 1 (PAC)
P117	CANRX2[Note]	Input	PM117 = 1	P117 = 0	PAC117 = 1 (PAC)
P120	$\overline{SCK5}$	Input	PM120 = 1	P120 = 0	PAC120 = 1 (PAC2)
		Output	PM120 = 0		
P121	SI5	Input	PM121 = 1	P121 = 0	PAC121 = 1 (PAC2)
P122	SO5	Output	PM122 = 0	P122 = 0	PAC122 = 1 (PAC2)
P123	$\overline{SCK6}$	Input	PM123 = 1	P123 = 0	PAC123 = 1 (PAC2)
		Output	PM123 = 0		
P124	SI6	Input	PM124 = 1	P124 = 0	PAC124 = 1 (PAC2)
P125	SO6	Output	PM125 = 0	P125 = 0	PAC125 = 1 (PAC2)
P126	TO10	Output	PM126 = 0	P126 = 0	–
P127	TO11	Output	PM127 = 0	P127 = 0	–
P140	SI3	Input	PM140 = 1	Setting not needed for P140	–
	RXD1	Input			
P141	SO3	Output	PM141 = 0	P141 = 0	–
	TXD1	Output			
P142	$\overline{SCK3}$	Input	PM142 = 1	Setting not needed for P142	–
		Output	PM142 = 0	P142 = 0	
	ASCK1	Input	PM142 = 1	Setting not needed for P142	
P143	RXD2	Input	PM143 = 1	Setting not needed for P143	–
P144	TXD2	Output	PM144 = 0	P144 = 0	–
P145	ASCK2	Input	PM145 = 1	Setting not needed for P145	–
P146	TI100	Input	PM146 = 1	Setting not needed for P146	–
P147	TI101	Input	PM147 = 1	Setting not needed for P147	–
P150	RXD3	Input	PM150 = 1	Setting not needed for P150	–
P151	TXD3	Output	PM151 = 0	P151 = 0	–

Note Only for the μPD703089Y and 70F3089Y

Table 5-16. Setting When Port Pin Is Used for Alternate Function (6/6)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn Register	Other Bits (Register)
	Function Name	I/O			
P152	ASCK3	Input	PM152 = 1	Setting not needed for P152	–
P153	TI110	Input	PM153 = 1	Setting not needed for P153	–
P154	TI111	Input	PM154 = 1	Setting not needed for P154	–
P155	TO12	Output	PM155 = 0	P155 = 0	–
P156	TI120	Input	PM156 = 1	Setting not needed for P156	–
P157	TI121	Input	PM157 = 1	Setting not needed for P157	–
P176	VM45	Output	PM176 = 0	P176 = 0	Refer to **14.3 (2)** (VM45C)

Caution When changing the output level of ports 0 and 3 by setting the port function output mode of ports 0 and 3, the interrupt request flag will be set because ports 0 and 3 also have alternate functions as external interrupt request inputs. Therefore, be sure to set the corresponding interrupt mask flag to 1 before using ports 0 and 3 as output pins.

Remark PMnx bit of PMn register and Pnx bit of Pn register

n: 0 (x = 0 to 7) n: 1 (x = 0 to 7) 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 7) n: 12 (x = 0 to 7) n: 14 (x = 0 to 7) n: 15 (x = 0 to 7)
n: 17 (x = 0 to 6)

★ 5.4 Operation of Port Function

The operation of a port differs depending on whether the port is in the input or output mode, as described below.

5.4.1 Writing data to I/O port

(1) In output mode

A value can be written to the output latch by using a transfer instruction. The contents of the output latch are output from the pin. Once data has been written to the output latch, it is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. Because the output buffer is off, however, the status of the pin does not change.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

Caution A bit manipulation instruction (CLR1, SET1, NOT1) manipulates 1 bit but accesses a port in 8-bit units. If this instruction is executed to manipulate a port with a mixture of input and output bits, the contents of the output latch of a pin set in the input mode, in addition to the bit to be manipulated, are overwritten to the current input pin status and become undefined.

5.4.2 Reading data from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch do not change.

(2) In input mode

The status of the pin can be read by using a transfer instruction. The contents of the output latch do not change.

CHAPTER 6 BUS CONTROL FUNCTION

The V850/SC1, V850/SC2, and V850/SC3 are provided with an external bus interface function by which external memories, such as ROM and RAM, and I/O can be connected.

6.1 Features

- Address bus (separate output possible only for the V850/SC1 and V850/SC2)
- 16-bit data bus
- Able to be connected to external devices via pins with alternate-functions as ports
- Wait function
  - Programmable wait function: up to 3 wait states can be inserted every 2 blocks
  - External wait control through $\overline{\text{WAIT}}$ input pin
- Idle state insertion function
- Bus mastership arbitration function
- Bus hold function

6.2 Bus Control Pins and Control Register

6.2.1 Bus control pins

The following pins are used for interfacing to external devices.

Table 6-1. 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)Note	Port 11 (P110 to P113)
Address bus (A5 to A12)Note	Port 10 (P100 to P107)
Address bus (A13 to A15)Note	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/$\overline{\text{W}}$, $\overline{\text{DSTB}}$, $\overline{\text{WRL}}$Note, $\overline{\text{WRH}}$Note, $\overline{\text{RD}}$Note)	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)

Note Only for the V850/SC1 and V850/SC2.

The bus interface function of each pin is enabled by setting the memory expansion mode register (MM) or memory address output mode register (MAM) for the V850/SC1 and V850/SC2, and the memory expansion mode register (MM) for the V850/SC3. For details of external bus interface operating mode specification, refer to **3.4.6 (1) Memory expansion mode register (MM)**, **(2) Memory address output mode register (MAM)**.

Caution For debugging when using the separate bus of the V850/SC1 and V850/SC2, refer to the user's manual of the relevant in-circuit emulator.

6.2.2 Control register

(1) System control register (SYC) (V850/SC1, V850/SC2)

This register switches control signals for the bus interface.

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

After reset: 00H R/W Address: FFFFF064H

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

BIC	Bus interface control
0	$\overline{DSTB}$, R/$\overline{W}$, $\overline{LBEN}$, $\overline{UBEN}$[Note] signals output
1	$\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{UBEN}$[Note] signals output

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

★ **Caution In the V850/SC1 and V850/SC2, when using port 9 as an I/O port, set the BIC bit to 0. Note that the BIC bit is 0 after system reset.**

6.3 Bus Access

6.3.1 Number of access clocks

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

Table 6-2. 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 wait insertions

6.3.2 Bus width

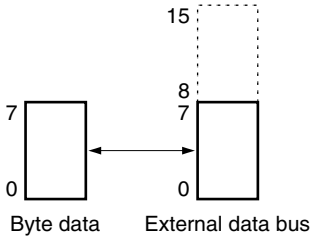
The CPU carries out peripheral I/O access and external memory access in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each access.

(1) Byte access (8 bits)

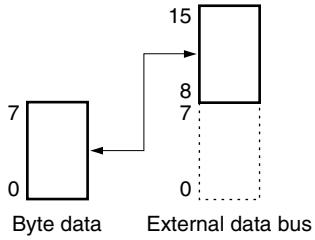
Byte access is divided into two types: access to even addresses and access to odd addresses.

Figure 6-1. Byte Access (8 Bits)

(a) Access to even addresses



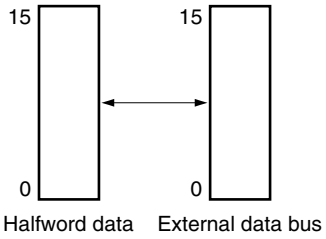
(b) Access to odd addresses



(2) Halfword access (16 bits)

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

Figure 6-2. Halfword Access (16 Bits)

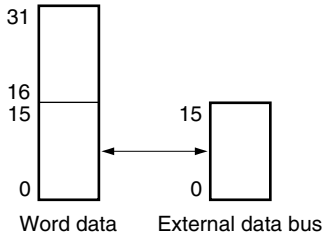


(3) Word access (32 bits)

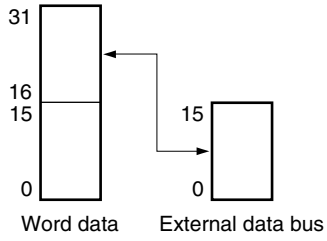
In word access to external memory, the lower halfword is accessed first and then the higher halfword is accessed.

Figure 6-3. Word Access (32 Bits)

First



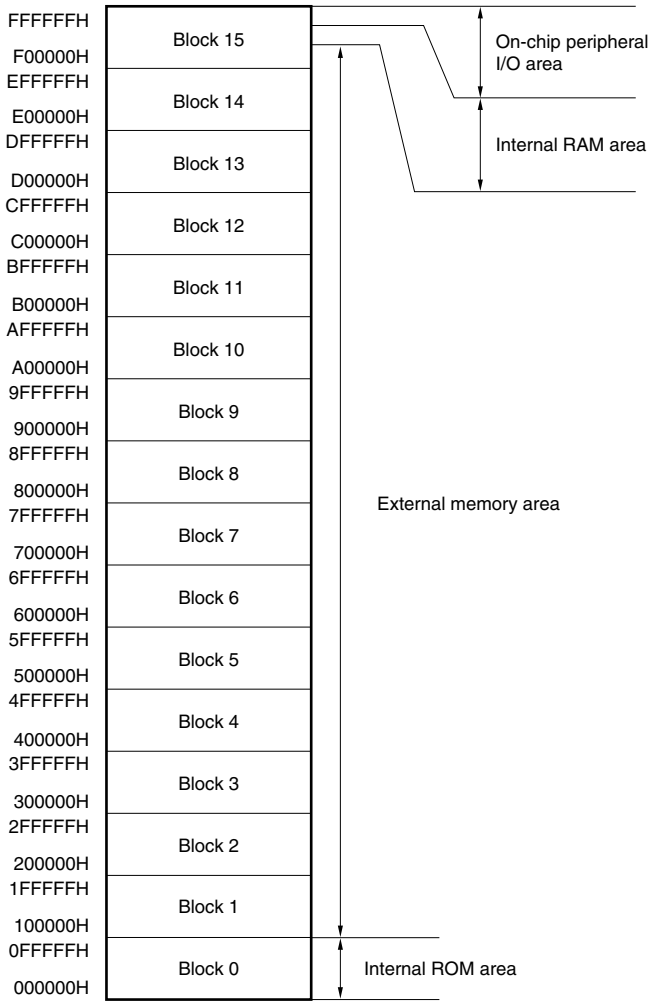
Second



6.4 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 6-4. Memory Block



6.5 Wait Function

6.5.1 Programmable wait function

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

The number of wait states can be programmed by using the data wait control register (DWC). Immediately after the system has been reset, three data wait insertion states are 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 also not subject to programmable wait control; wait control is dependent upon the execution of each peripheral function.

6.5.2 External wait function

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

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

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

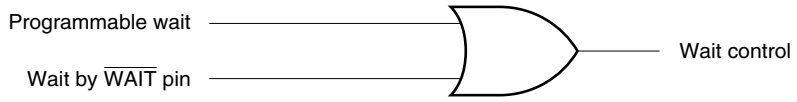
Caution Because the A1 pin and $\overline{\text{WAIT}}$ pin are alternate-function pins in the V850/SC1 and V850/SC2, the $\overline{\text{WAIT}}$ pin based wait function cannot be used when using a separate bus (programmable wait can be used, however).

Similarly, a separate bus cannot be used when the $\overline{\text{WAIT}}$ pin based wait function is being used.

6.5.3 Relationship between programmable wait and external wait

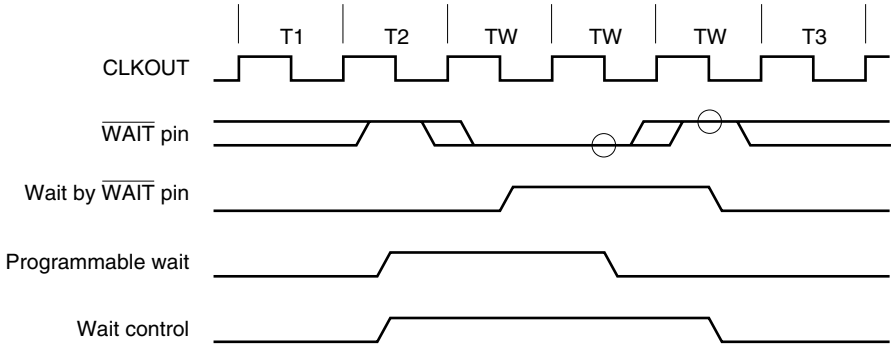
A wait cycle is inserted as a result of an OR operation between the wait cycle specified by the set value of a programmable wait and the wait cycle controlled by the $\overline{\text{WAIT}}$ pin. In other words, the number of wait cycles is determined by whichever has more cycles.

Figure 6-5. 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 6-6. Example of Inserting Wait States



Remark ○: Valid sampling timing

6.6 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 bus cycle following continuous bus cycles 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 insert 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.

6.7 Bus Hold Function

6.7.1 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.

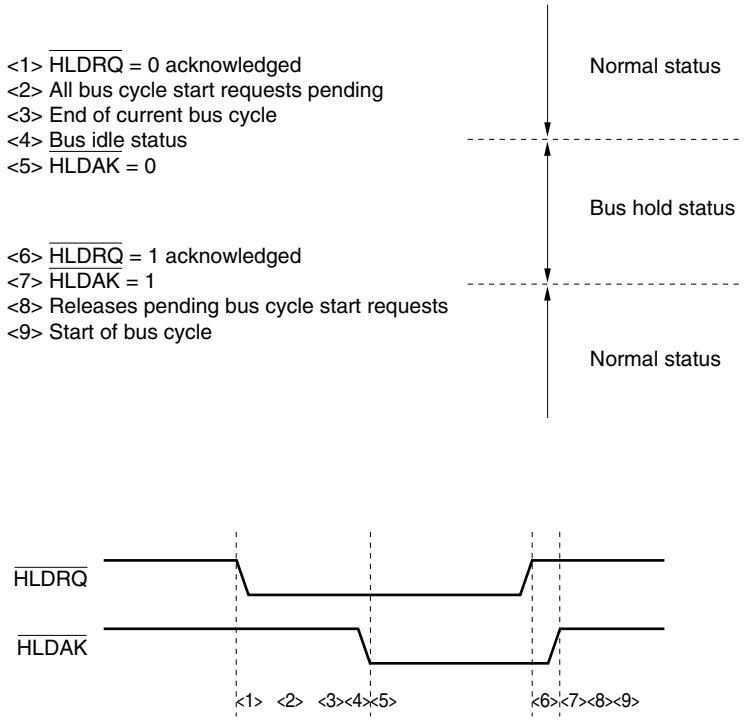
Bus hold requests are not acknowledged between the first and the second word access, nor between a read access and a write access in the read modify write access of a bit manipulation instruction.

Note When using a separate bus in the V850/SC1 and V850/SC2, pins A1 to A15 are in a held state.

6.7.2 Bus hold procedure

The procedure of the bus hold function is illustrated below.

Figure 6-7. Bus Hold Procedure



6.7.3 Operation in power save mode

In the IDLE or STOP mode, the system clock is stopped. Consequently, the bus hold status is not set even if the HLDRQ pin becomes active.

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

6.8 Bus Timing

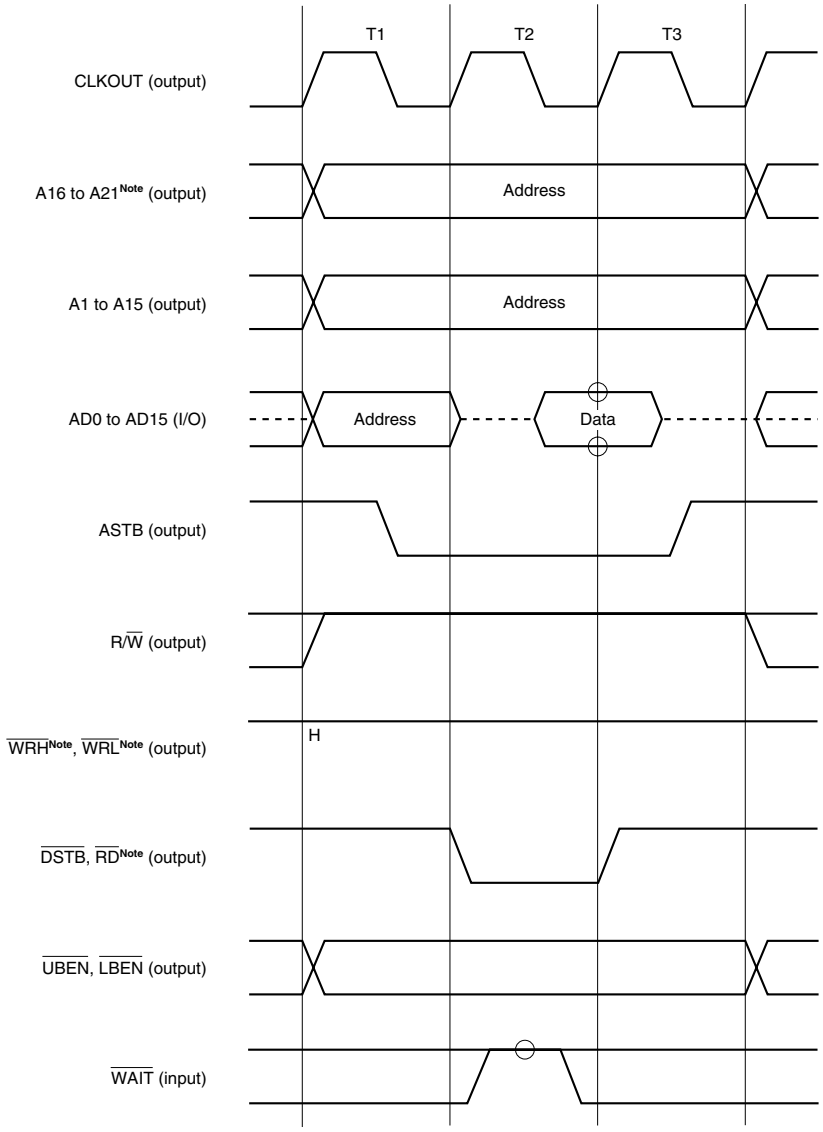
The V850/SC1, V850/SC2, and V850/SC3 can execute read/write control for an external device using the following two modes.

- 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 **6.2.2 (1) System control register (SYC) (V850/SC1, V850/SC2)**).

Figure 6-8. Memory Read (1/4)

(a) 0 waits



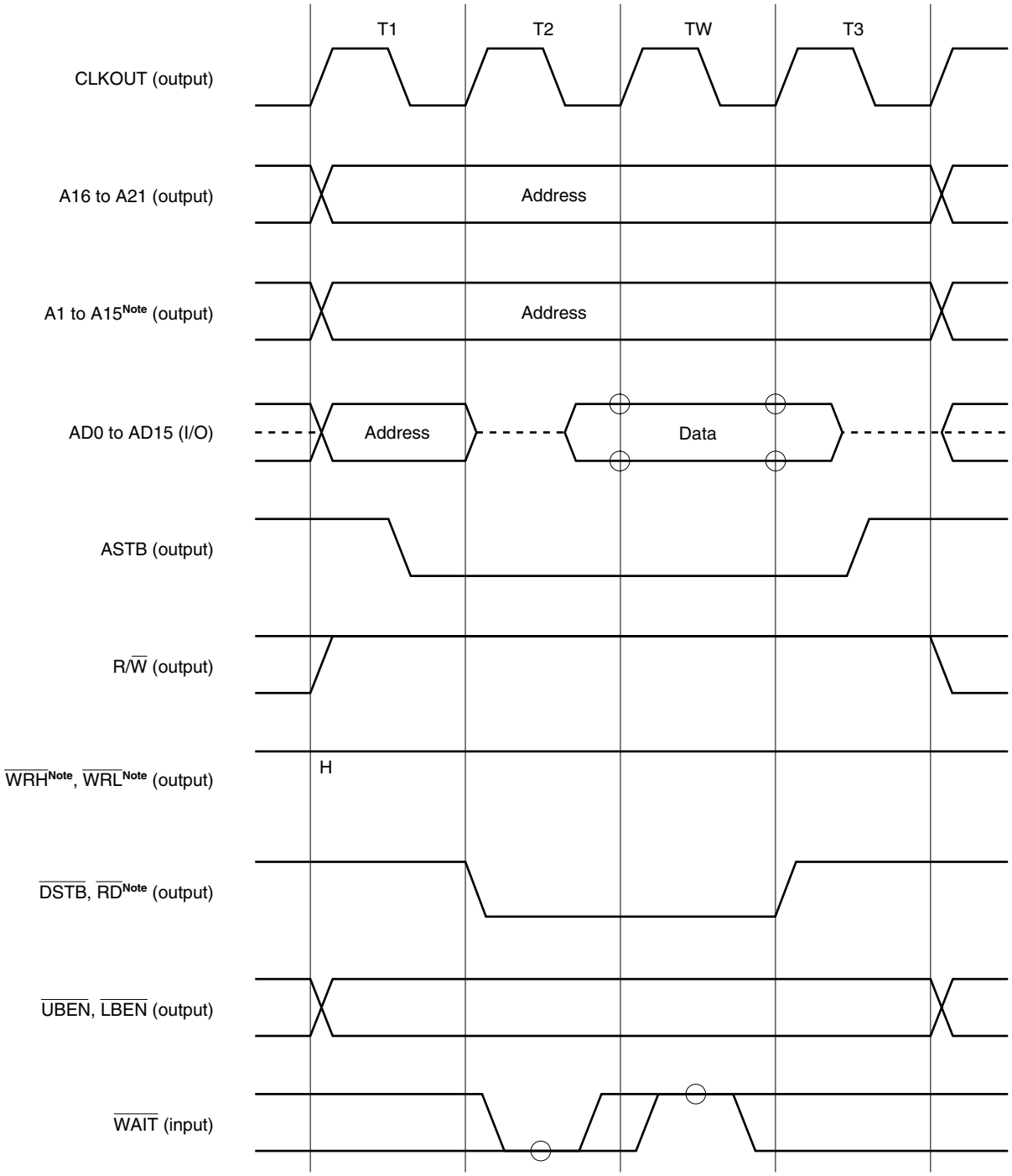
Note Only for the V850/SC1 and V850/SC2

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

2. The broken lines indicate the high-impedance state.

Figure 6-8. Memory Read (2/4)

(b) 1 wait



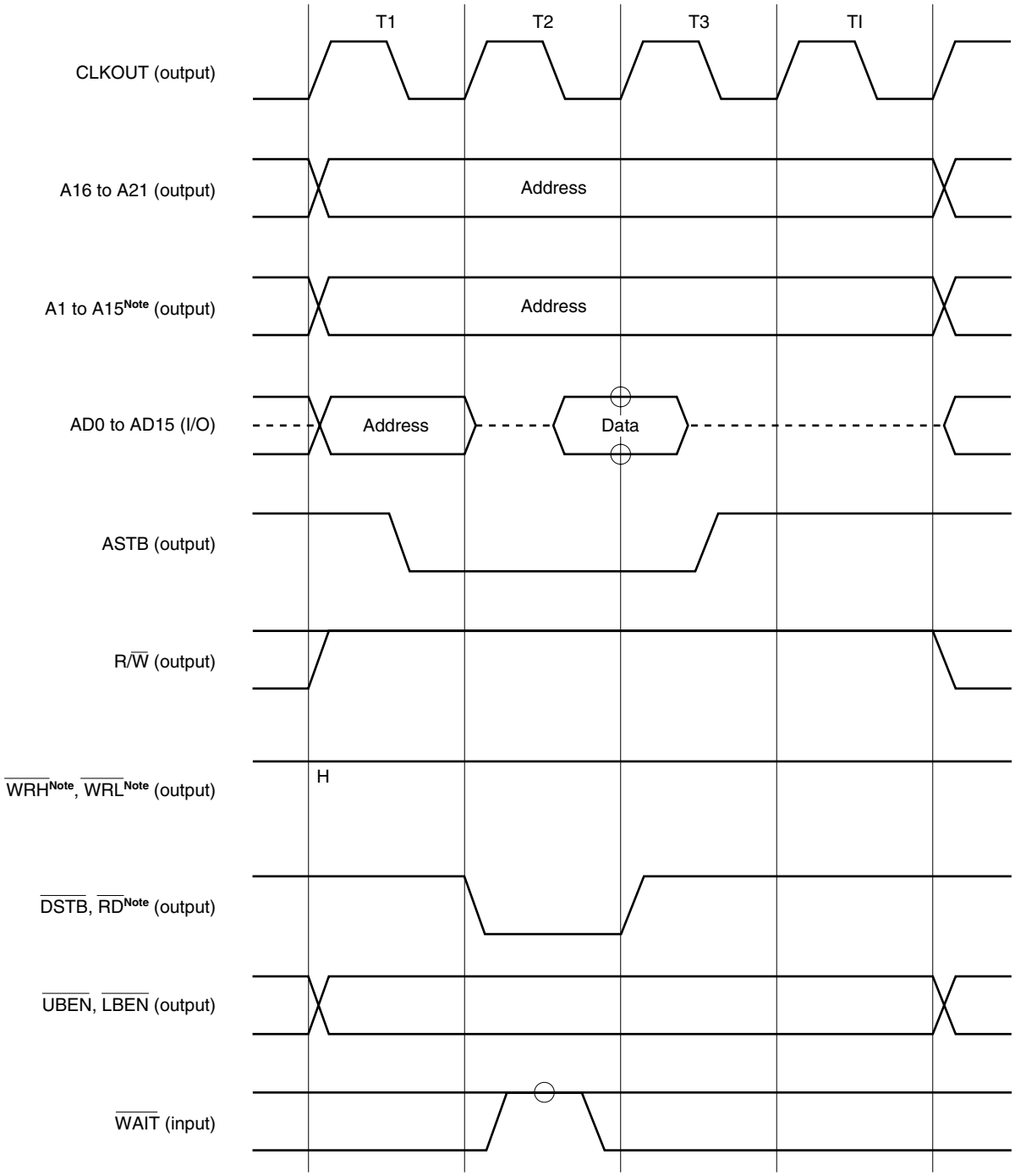
Note Only for the V850/SC1 and V850/SC2

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

2. The broken lines indicate the high-impedance state.

Figure 6-8. Memory Read (3/4)

(c) 0 waits, idle state



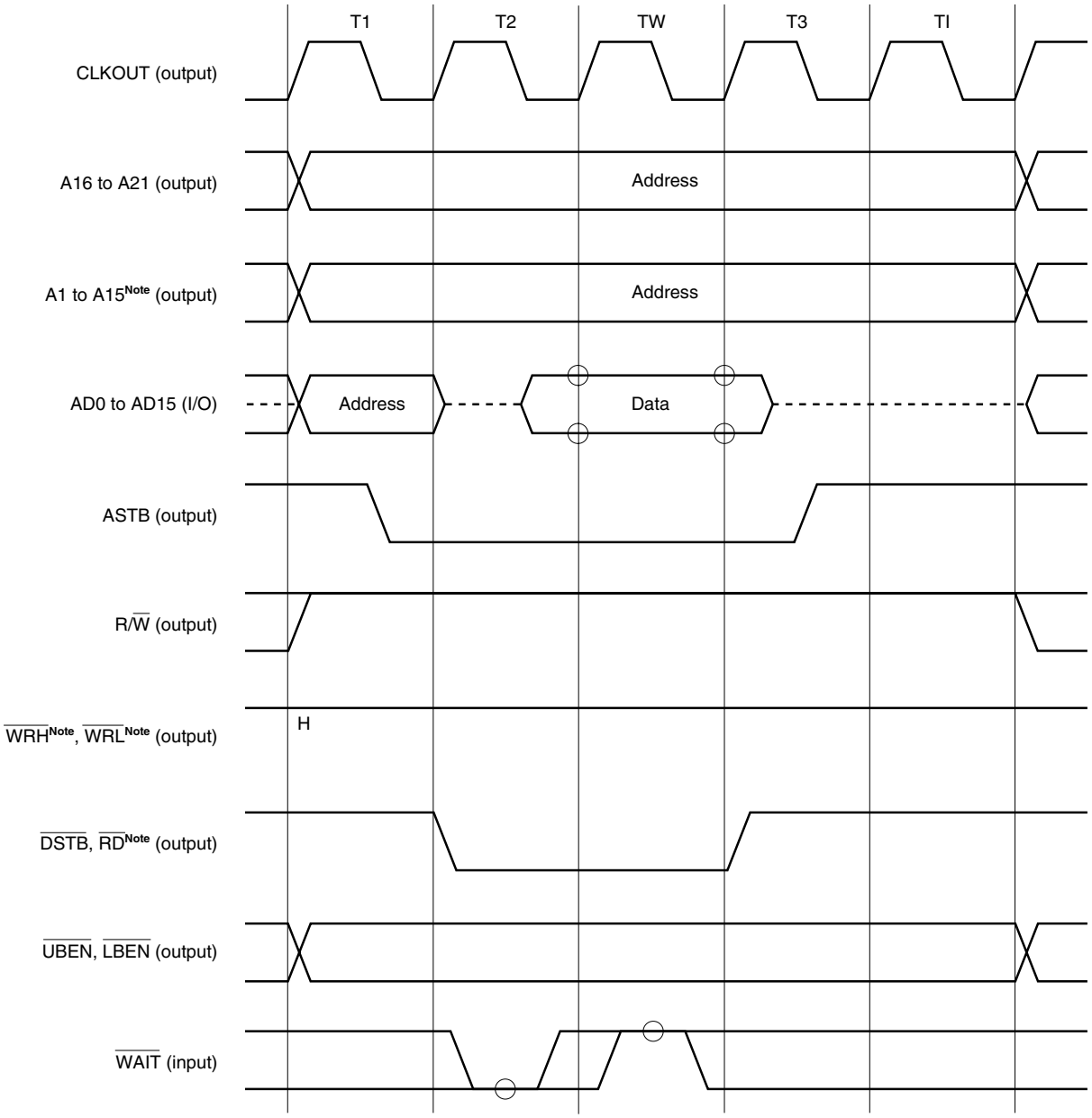
Note Only for the V850/SC1 and V850/SC2

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

2. The broken lines indicate the high-impedance state.

Figure 6-8. Memory Read (4/4)

(d) 1 wait, idle state



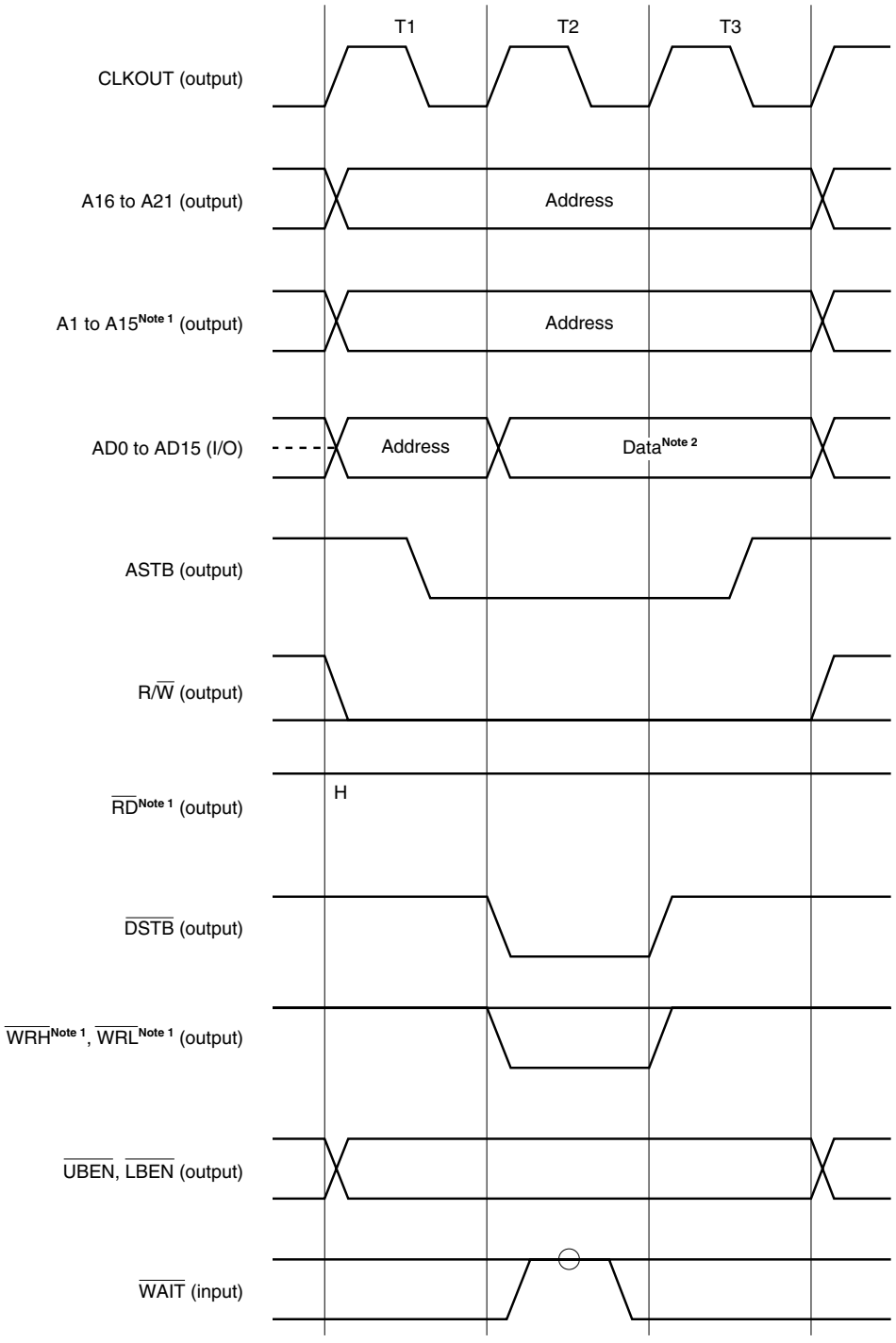
Note Only for the V850/SC1 and V850/SC2

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

2. The broken lines indicate the high-impedance state.

Figure 6-9. Memory Write (1/2)

(a) 0 waits



Notes 1. Only for the V850/SC1 and V850/SC2

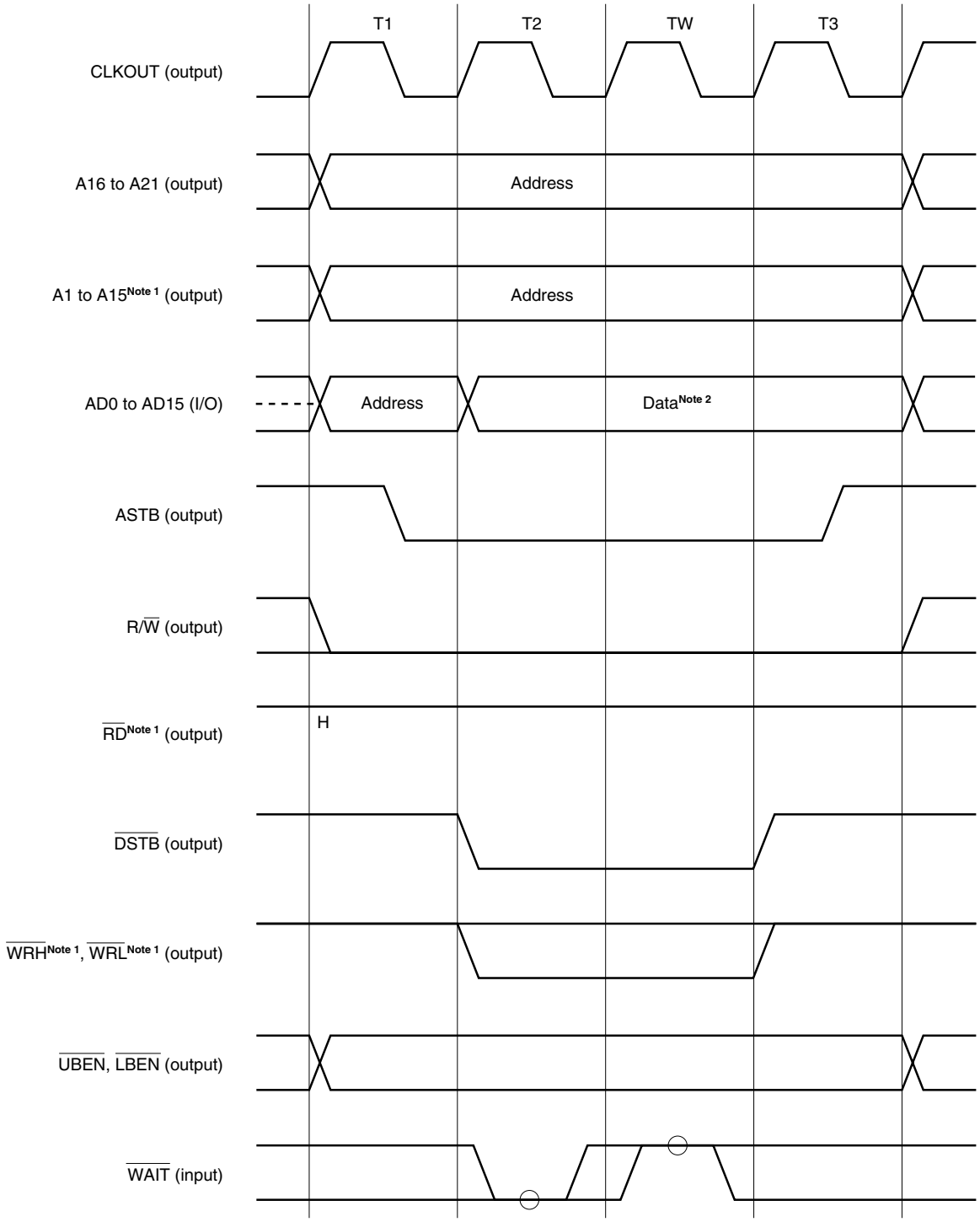
2. AD0 to AD7 output invalid data when odd-address byte data is accessed.
AD8 to AD15 output invalid data when even-address byte data is accessed.

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

2. The broken lines indicate the high-impedance state.

Figure 6-9. Memory Write (2/2)

(b) 1 wait



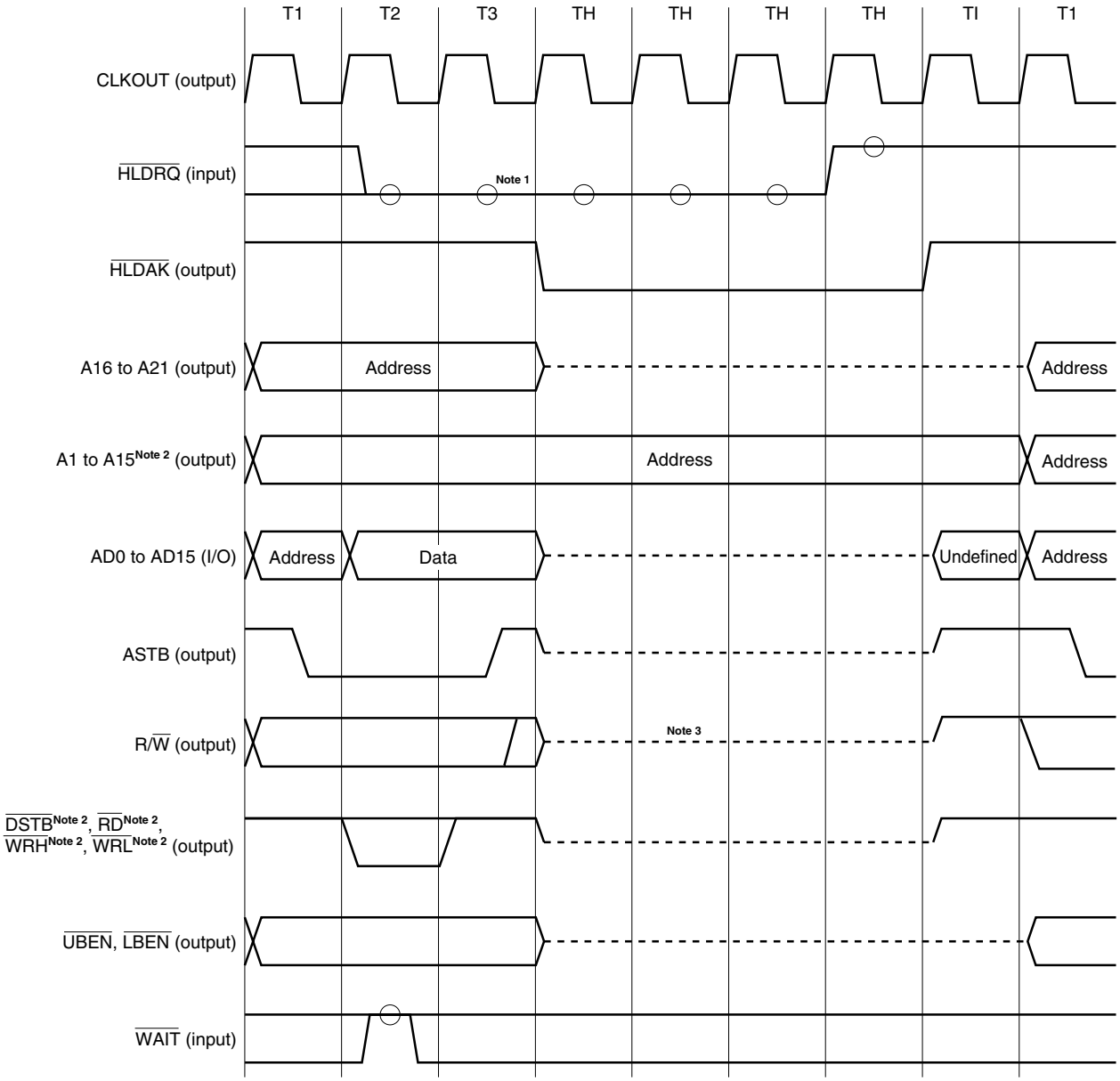
Notes 1. Only for the V850/SC1 and V850/SC2

2. 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 lines indicate the high-impedance state.

Figure 6-10. 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. Only for the V850/SC1 and V850/SC2

3. 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 lines indicate the high-impedance state.

6.9 Bus Priority

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

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

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

Table 6-3. Bus Priority

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

6.10 Memory Boundary Operation Condition

6.10.1 Program space

(1) Do not execute a branch to the on-chip peripheral I/O area or a continuous fetch from the internal RAM area to peripheral I/O area. If a branch or instruction fetch is executed, the NOP instruction code is continuously fetched and fetching from external memory is not performed.
(2) A prefetch operation extending 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.

6.10.2 Data space

Only the address aligned at the halfword boundary (when the least significant bit of the address is “0”)/word boundary (when the lowest 2 bits of the address are “0”) is accessed by halfword (16 bits)/word (32 bits) access, respectively.

Therefore, access that extends over the memory or memory block boundary does not take place.

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

CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION

7.1 Outline

The V850/SC1, V850/SC2, and V850/SC3 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 49 to 56 sources.

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

The V850/SC1, V850/SC2, and V850/SC3 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started (exception trap) by the TRAP instruction (software exception) or by generation of an exception event (fetching of an illegal op code).

7.1.1 Features

- Interrupts
  - Non-maskable: 2 sources
  - Maskable: (Number of maskable interrupt sources differs depending on product)
    (V850/SC1)
    μPD703068Y, 70F3089Y: 49 sources
    (V850/SC2)
    μPD703069Y, 70F3089Y: 51 sources
    (V850/SC3)
    μPD703088Y: 53 sources
    μPD703089Y, 70F3089Y: 56 sources
  - 8 levels of programmable priorities
  - Mask specification for interrupt requests according to priority
  - Masks can be individually specified for maskable interrupt requests.
  - 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 op code exception)

The interrupt/exception sources are listed in Table 7-1.

Table 7-1. Interrupt Source List (1/3)

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 op code	–	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	INTCSI5	CSI5 transmit end	SIO5	0100H	00000100H	nextPC	CSIC5
		9	INTAD	A/D conversion end	A/D	0110H	00000110H	nextPC	ADIC
		10	INTDMA0	DMA0 transfer end	DMA0	0120H	00000120H	nextPC	DMAIC0
		11	INTDMA1	DMA1 transfer end	DMA1	0130H	00000130H	nextPC	DMAIC1
		12	INTDMA2	DMA2 transfer end	DMA2	0140H	00000140H	nextPC	DMAIC2
		13	INTTM00	TM0 and CR00 match/ TI01 pin valid edge	TM0	0150H	00000150H	nextPC	TMIC00
		14	INTTM01	TM0 and CR01 match/ TI00 pin valid edge	TM0	0160H	00000160H	nextPC	TMIC01
		15	INTTM10	TM1 and CR10 match/ TI11 pin valid edge	TM1	0170H	00000170H	nextPC	TMIC10
		16	INTTM11	TM1 and CR11 match/ TI10 pin valid edge	TM1	0180H	00000180H	nextPC	TMIC11
		17	INTTM70	TM7 and CR70 match/ TI71 pin valid edge	TM7	0190H	00000190H	nextPC	TMIC70
		18	INTTM71	TM7 and CR71 match/ TI70 pin valid edge	TM7	01A0H	000001A0H	nextPC	TMIC71
		19	INTCSI6	CSI6 transmission/ reception completion	CSI6	01B0H	000001B0H	nextPC	CSIC6
		20	INTTM5/ INTP8[Note 2]	TM5 compare match/ OVF/INTP8 pin	TM5/pin	01C0H	000001C0H	nextPC	TMIC5
		21	INTWTN	Watch timer OVF	WTN	01D0H	000001D0H	nextPC	WTNIC
		22	INTWTNI	Watch timer prescaler	WTN	01E0H	000001E0H	nextPC	WTNIIC
		23	INTIIC0/ INTCSI0	I^2C0 interrupt/ CSI0 transmit end	I^2C0/ SIO0	01F0H	000001F0H	nextPC	CSIC0

Notes 1. n: 0 to FH

2. When using INTP8 or INTP9, stop TM5 and TM6 (TCEm0 bit of TMCm0 register = 0) and do not use them (m = 5, 6). When using TM5 or TM6, do not specify the valid edge for INTP8 and INTP9 (EGP1n bit of EGP1 register and EGN1n bit of EGN1 register = 0) and do not use them as external interrupts (they can be used as ports) (n = 6, 7).

Table 7-1. Interrupt Source List (2/3)

Type	Classification	Default Priority	Name	Trigger	Interrupt Source	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	24	INTTM6/ INTP9[Note 1]	TM6 compare match/ OVF/INTP9 pin	TM6/pin	0200H	00000200H	nextPC	TMIC6
		25	INTSR0/ INTCSI4	UART receive end/ CSI4 transmit end	UART0/ SIO4	0210H	00000210H	nextPC	CSIC4
		26	INTST0	UART0 transmit end	UART0	0220H	00000220H	nextPC	STIC0
		27	INTKR	Key return interrupt	KR	0230H	00000230H	nextPC	KRIC
		28	INTCE1[Note 2]/ INTIE1[Note 3]	FCAN1 serial error/ IEBus transfer end	FCAN1/ IEBus	0240H	00000240H	nextPC	CANIC1/ IEBIC1
		29	INTCR1[Note 2]/ INTIE2[Note 3]	FCAN1 reception/ IEBus communication end	FCAN1/ IEBus	0250H	00000250H	nextPC	CANIC2/ IEBIC2
		30	INTCT1[Note 2]	FCAN1 transmission	FCAN1	0260H	00000260H	nextPC	CANIC3
		31	INTCME[Note 2]	FCAN memory access error	FCAN1/ 2	0270H	00000270H	nextPC	CANIC7
		32	INTTM80	TM8 and CR80 match/ TI81 pin valid edge	TM8	0280H	00000280H	nextPC	TMIC80
		33	INTTM81	TM81 and CR81 match/ TI80 pin valid edge	TM8	0290H	00000290H	nextPC	TMIC81
		34	INTTM90	TM9 and CR90 match/ TI91 pin valid edge	TM9	02A0H	000002A0H	nextPC	TMIC90
		35	INTTM91	TM9 and CR91 match/ TI90 pin valid edge	TM9	02B0H	000002B0H	nextPC	TMIC91
		36	INTSR1/ INTCSI3	UART1 receive end/ CSI3 transmit end	UART1/ CSI3	02C0H	000002C0H	nextPC	CSIC3
		37	INTST1	UART1 transmit end	UART1	02D0H	000002D0H	nextPC	STIC1
		38	INTDMA3	DMA3 transfer end	DMA3	02E0H	000002E0H	nextPC	DMAIC3
		39	INTDMA4	DMA4 transfer end	DMA4	02F0H	000002F0H	nextPC	DMAIC4
		40	INTDMA5	DMA5 transfer end	DMA5	0300H	00000300H	nextPC	DMAIC5
		41	INTCE2[Note 4]	FCAN2 serial error	FCAN2	0310H	00000310H	nextPC	CANIC4
		42	INTCR2[Note 4]	FCAN2 reception	FCAN2	0320H	00000320H	nextPC	CANIC5
		43	INTCT2[Note 4]	FCAN2 transmission	FCAN2	0330H	00000330H	nextPC	CANIC6
		44	INTP7	INTP7 pin	Pin	0340H	00000340H	nextPC	PIC7
		45	INTSR2	UART2 receive end	UART2	0350H	00000350H	nextPC	SRIC2
		46	INTST2	UART2 transmit end	UART2	0360H	00000360H	nextPC	STIC2
		47	INTSR3	UART3 receive end	UART3	0370H	00000370H	nextPC	SRIC3
		48	INTST3	UART3 transmit end	UART3	0380H	00000380H	nextPC	STIC3
		49	INTTM100	TM10 and CR100 match/ TI101 pin valid edge	TM10	0390H	00000390H	nextPC	TMIC100

Notes 1. When using INTP8 or INTP9, stop TM5 and TM6 (TCEm0 bit of TMCm0 register = 0) and do not use them (m = 5, 6). When using TM5 or TM6, do not specify the valid edge for INTP8 and INTP9 (EGP1n bit of EGP1 register and EGN1n bit of EGN1 register = 0) and do not use them as external interrupts (they can be used as ports) (n = 6, 7).

2. Only for the V850/SC3

3. Only for the V850/SC2

4. Only for the μPD703089Y and 70F3089Y

Table 7-1. Interrupt Source List (3/3)

Type	Classification	Default Priority	Name	Trigger	Interrupt Source	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	50	INTTM101	TM10 and CR101 match/ TI100 pin valid edge	TM10	03A0H	000003A0H	nextPC	TMIC101
		51	INTTM110	TM11 and CR110 match/ TI111 pin valid edge	TM11	03B0H	000003B0H	nextPC	TMIC110
		52	INTTM111	TM11 and CR111 match/ TI110 pin valid edge	TM11	03C0H	000003C0H	nextPC	TMIC111
		53	INTTM120	TM12 and CR120 match/ TI121 pin valid edge	TM12	03D0H	000003D0H	nextPC	TMIC120
		54	INTTM121	TM12 and CR121 match/ TI120 pin valid edge	TM12	03E0H	000003E0H	nextPC	TMIC121
		55	INTIIC1/ INTIIC2	I^2C interrupt/ CSI2 transmit end	I^2C1/ SIO2	03F0H	000003F0H	nextPC	CSIC2

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 granted 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 op code exception occurs is calculated with (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 and maskable interrupts (INTWDT and INTWDTM) are set by the WDTM4 bit of the watchdog timer mode register (WDTM).

7.2 Non-Maskable Interrupts

Non-maskable interrupts are 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/SC1, V850/SC2, and V850/SC3.

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

When the valid edge specified by the 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 when the WDTM4 bit of the watchdog timer mode register (WDTM) is set to 1.

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

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

7.2.1 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 the exception code (0010H, 0020H) to the higher halfword (FECC) of ECR.
(4) Sets the NP and ID bits of the PSW and clears the EP bit.
(5) Loads the handler address (00000010H, 00000020H) of the non-maskable interrupt routine to the PC, and transfers control.

Figure 7-1. Non-Maskable Interrupt Servicing

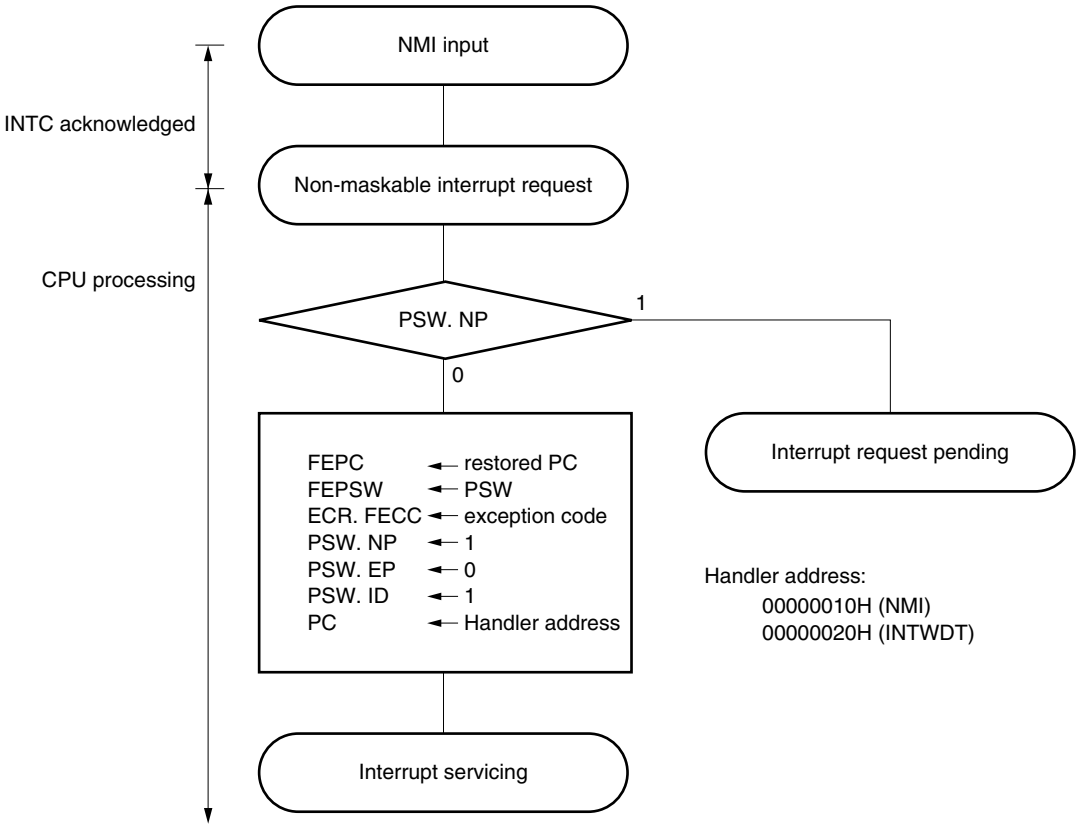
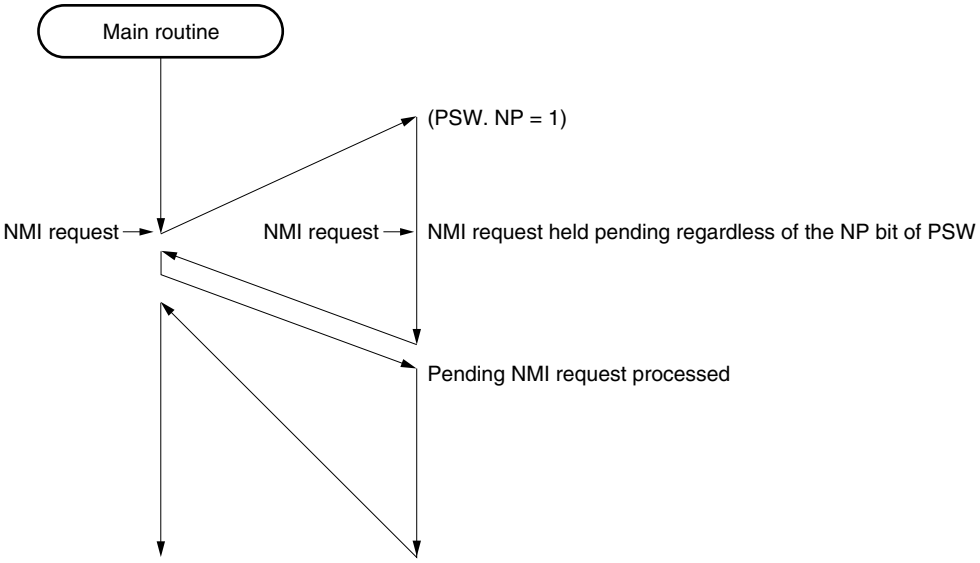
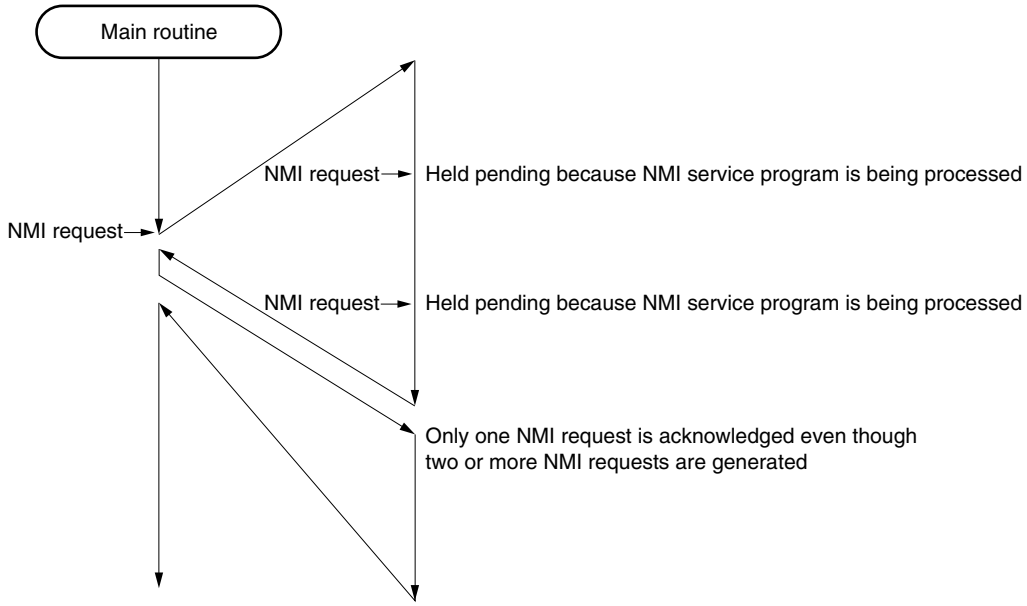


Figure 7-2. Acknowledging Non-Maskable Interrupt Requests

(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:



7.2.2 Restore

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

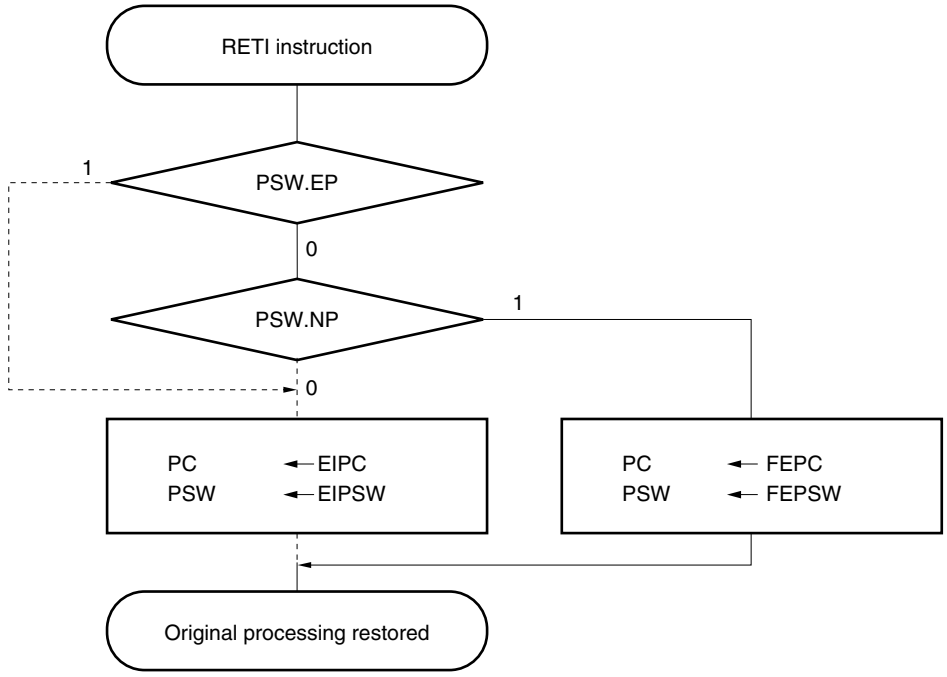
Operation of RETI instruction

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 7-3. 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.

7.2.3 NP flag

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

Figure 7-4. NP Flag (NP)

After reset: 00000020H

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

NP	NMI servicing state
0	No NMI interrupt servicing
1	NMI interrupt currently being serviced

7.2.4 Noise eliminator of NMI pin

NMI pin noise is eliminated by a 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 only after a certain period has elapsed.

The NMI pin is used for canceling the software stop mode. In the software stop mode, noise elimination using the system clock does not occur because the internal system clock is stopped.

7.2.5 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, neither rising nor falling edge detected.

Rising edge specification register 0 (EGP0) and falling edge specification register 0 (EGN0) specify the valid edge of non-maskable interrupts (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 “neither rising nor falling edge detected” 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 “neither rising nor falling edge detected”.

(1) Format of 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 validity control
0	No interrupt request signal occurs at the rising edge
1	Interrupt request signal occurs at the rising edge

n = 0: NMI pin control

n = 1 to 7: INTP0 to INTP6 pins control

(2) Format of 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 validity control
0	No interrupt request signal occurs at the falling edge
1	Interrupt request signal occurs at the falling edge

n = 0: NMI pin control

n = 1 to 7: INTP0 to INTP6 pins control

7.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850/SC1, V850/SC2, and V850/SC3 have 49 to 56 maskable interrupt sources (refer to **7.1.1 Features**).

If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers, allowing programmable priority control.

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

When the EI instruction is executed in an interrupt servicing routine, the interrupt enabled (EI) status is set, which enables interrupts having a higher priority to immediately interrupt the service routine currently 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).

7.3.1 Operation

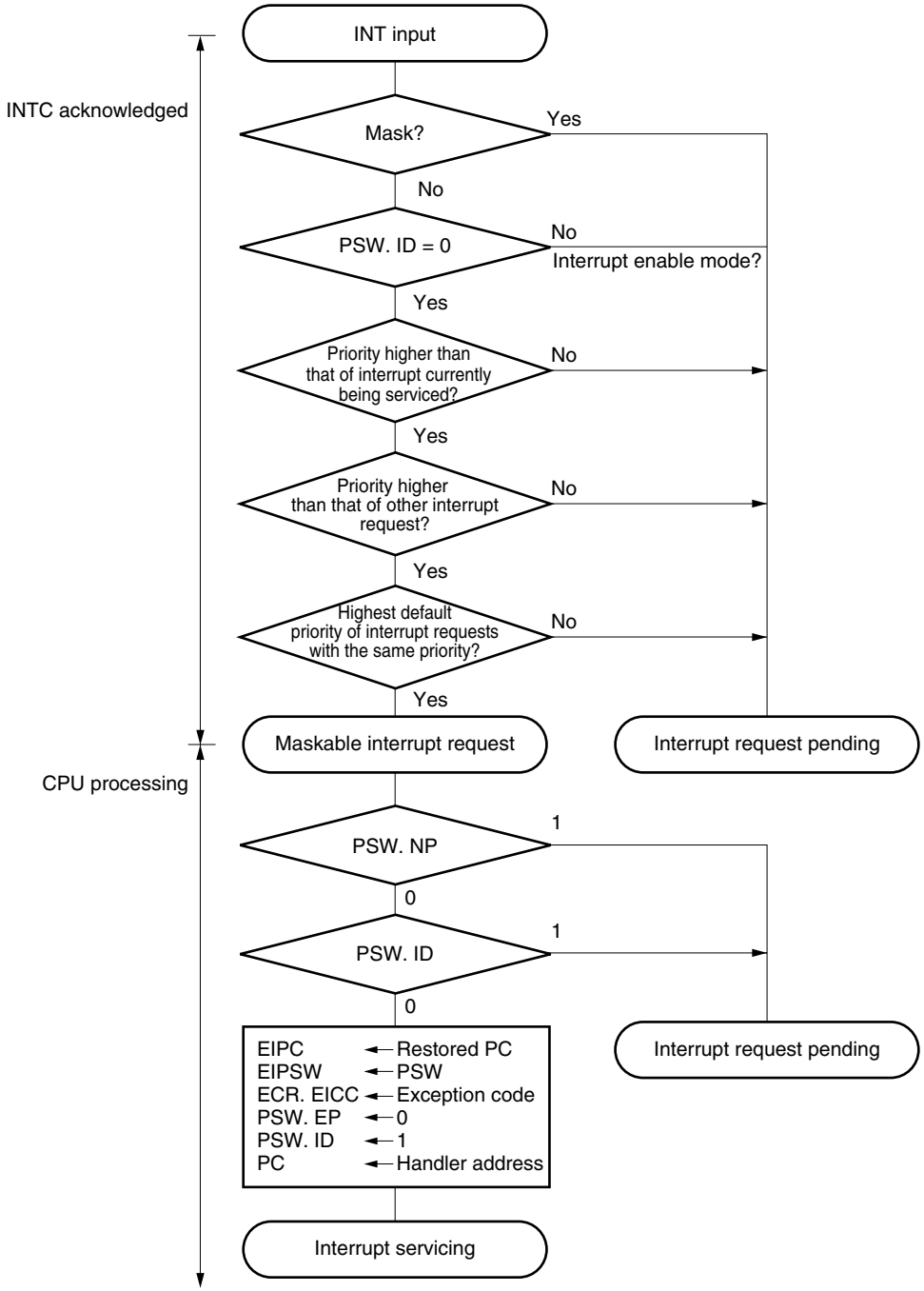
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine:

(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 while another interrupt is being serviced (when PSW.NP = 1 or PSW.ID = 1) are held pending internally. 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 a new maskable interrupt servicing.

How maskable interrupts are serviced is shown below.

Figure 7-5. Maskable Interrupt Servicing



7.3.2 Restore

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

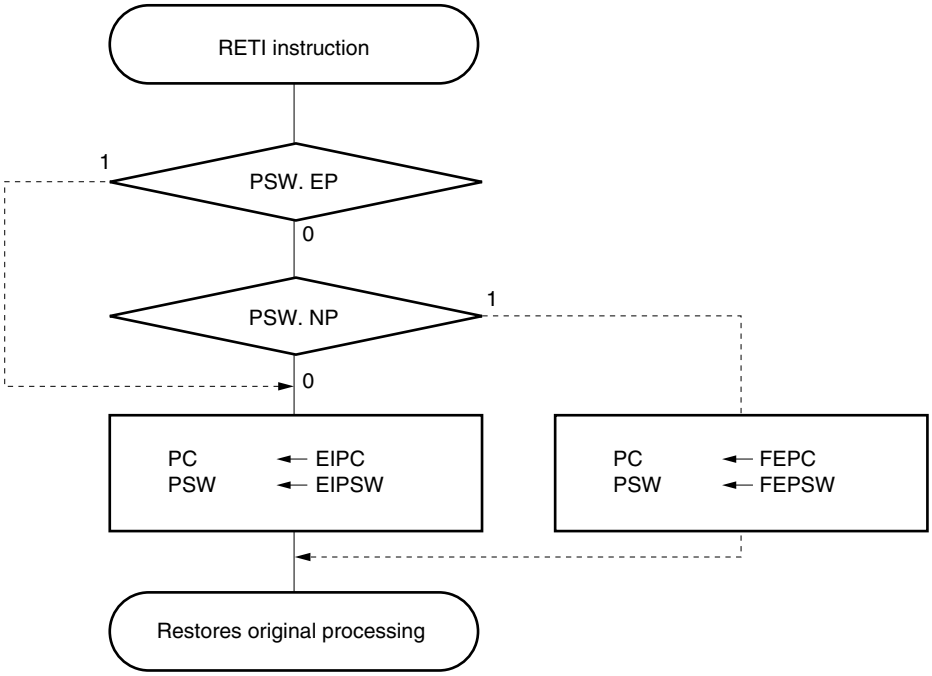
Operation of RETI instruction

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

(1) Restores the values of the PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.
(2) Transfers control to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 7-6. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

7.3.3 Priorities of maskable interrupts

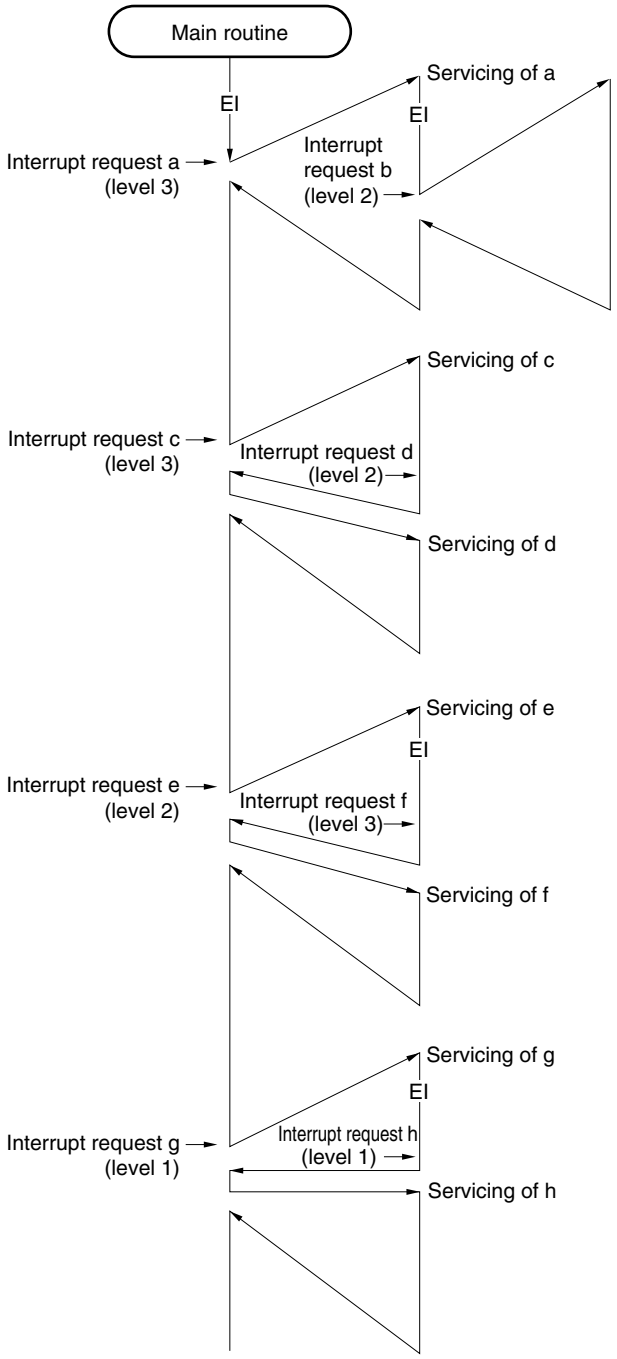
The V850/SC1, V850/SC2, and V850/SC3 provide a multiple interrupt service in which an interrupt can be acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels 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 7-1. Programmable priority control divides 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 (1). Therefore, when multiple interrupts are to be used, clear (0) the ID flag beforehand (for example, by placing the EI instruction into the interrupt service program) to set the interrupt enable mode.

Remark xx: Identifying name of each peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

Figure 7-7. Example of Interrupt Nesting Process (1/2)



Interrupt request b is acknowledged because the priority of b is higher than that of a and interrupts are enabled.

Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.

Interrupt request f is held pending even if interrupts are enabled because its priority is lower than that of e.

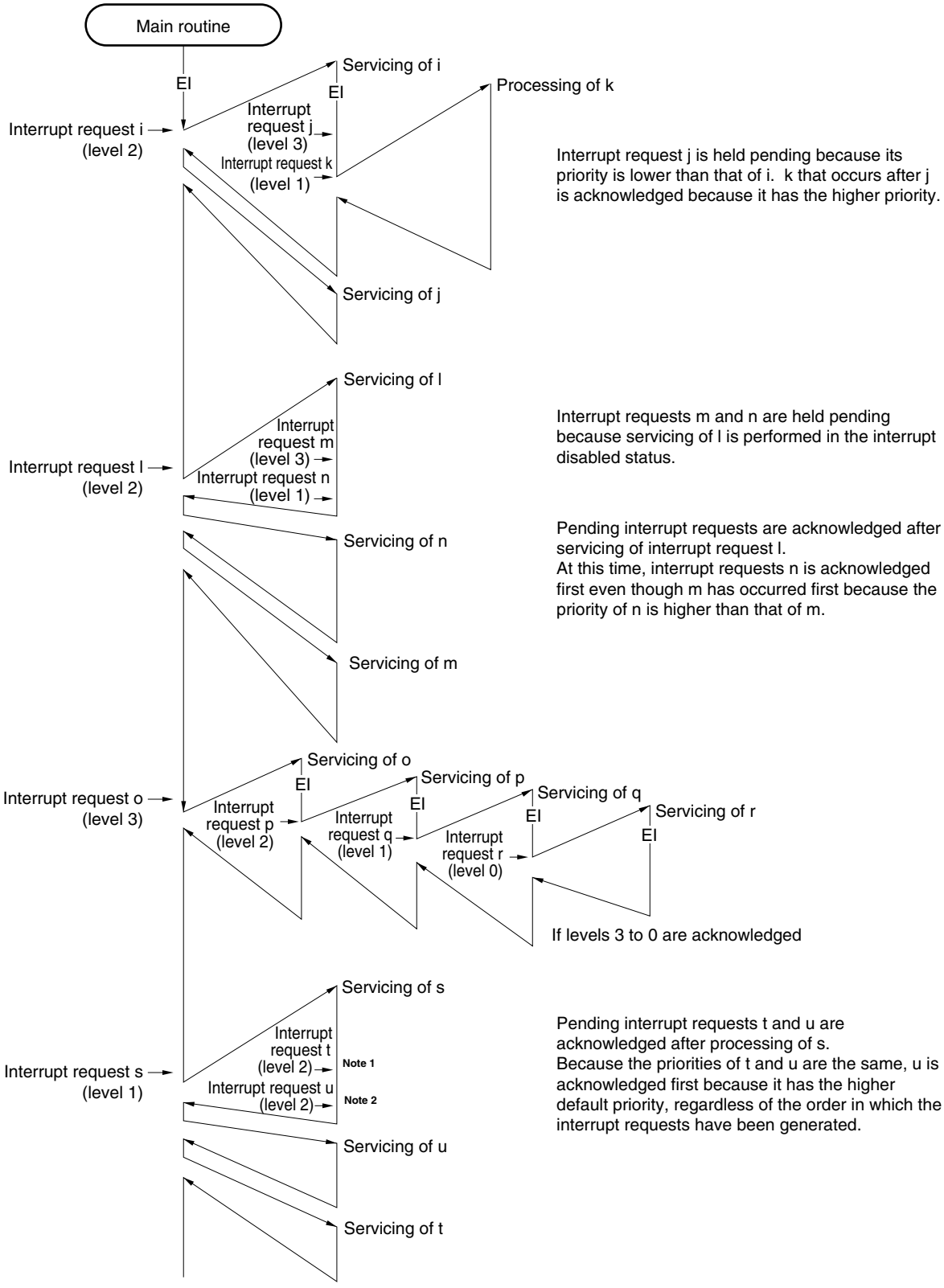
Interrupt request h is held pending even if interrupts are enabled because its priority is the same as that of g.

Caution 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.

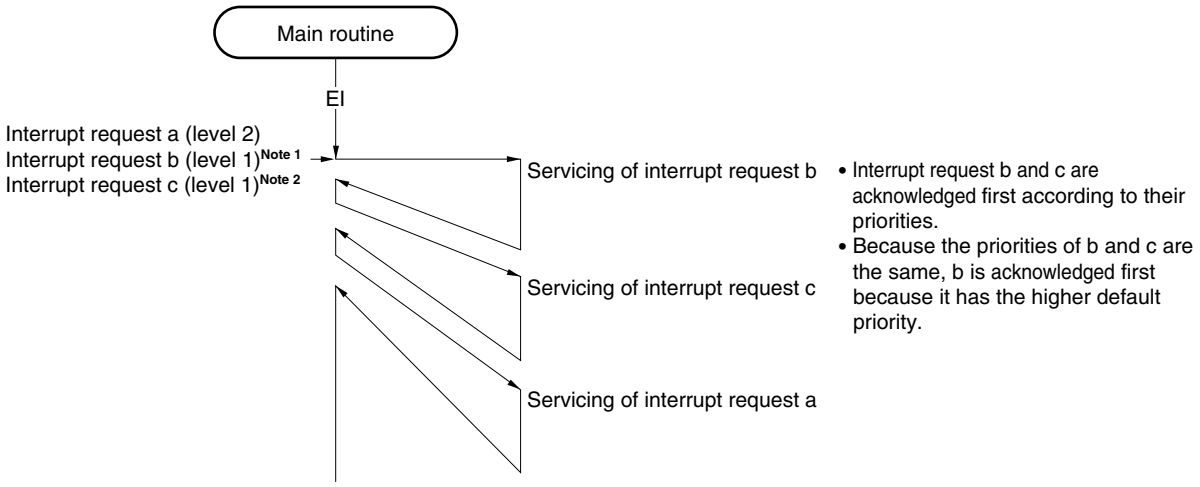
Figure 7-7. Example of Interrupt Nesting Process (2/2)



Notes 1. Lower default priority

2. Higher default priority

Figure 7-8. Example of Servicing Interrupt Requests Simultaneously Generated



Notes 1. Higher default priority

2. Lower default priority

Remarks 1. a to c 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.

7.3.4 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- or 1-bit units.

Caution If the following three conditions conflict, interrupt servicing is executed twice. However, this does not occur when DMA is not used.

- **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.**

After reset: 47H R/W Address: FFFFF100H to FFFFF172H

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	Enables interrupt servicing
1	Disables interrupt servicing (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 interrupt request is acknowledged.

Remark xx: Identifying name of each peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

The addresses and bits of the interrupt control registers are as follows.

Table 7-2. Interrupt Control Registers (xxICn) (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF100H	WDTIC	WDTIF	WDTMK	0	0	0	WDTPR2	WDTPR1	WDTPR0
FFFFF102H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF104H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF106H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF108H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF10AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF10CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF10EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF110H	CSIC5	CSIF5	CSMK5	0	0	0	CSPR52	CSPR51	CSPR50
FFFFF112H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF114H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF116H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF118H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
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	TMIC70	TMIF70	TMMK70	0	0	0	TMPR702	TMPR701	TMPR700
FFFFF124H	TMIC71	TMIF71	TMMK71	0	0	0	TMPR712	TMPR711	TMPR710
FFFFF126H	CSIC6	CSIF6	CSMK6	0	0	0	CSPR62	CSPR61	CSPR60
FFFFF128H	TMIC5	TMIF5	TMMK5	0	0	0	TMPR52	TMPR51	TMPR50
FFFFF12AH	WTNIC	WTNIF	WTNMK	0	0	0	WTNPR2	WTNPR1	WTNPR0
FFFFF12CH	WTNIIC	WTNIIF	WTNIMK	0	0	0	WTNIPR2	WTNIPR1	WTNIPR0
FFFFF12EH	CSIC0	CSIF0	CSMK0	0	0	0	CSPR02	CSPR01	CSPR00
FFFFF130H	TMIC6	TMIF6	TMMK6	0	0	0	TMPR62	TMPR61	TMPR60
FFFFF132H	CSIC4	CSIF4	CSMK4	0	0	0	CSPR42	CSPR41	CSPR40
FFFFF134H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF136H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF138H	CANIC1[Note 1]	CANIF1	CANMK1	0	0	0	CANPR12	CANPR11	CANPR10
	IEBIC1[Note 2]	IEBIF1	IEBMK1	0	0	0	IEBPR12	IEBPR11	IEBPR10
FFFFF13AH	CANIC2[Note 1]	CANIF2	CANMK2	0	0	0	CANPR22	CANPR21	CANPR20
	IEBIC2[Note 2]	IEBIF2	IEBMK2	0	0	0	IEBPR22	IEBPR21	IEBPR20
FFFFF13CH	CANIC3[Note 1]	CANIF3	CANMK3	0	0	0	CANPR32	CANPR31	CANPR30
FFFFF13EH	CANIC7[Note 1]	CANIF7	CANMK7	0	0	0	CANPR72	CANPR71	CANPR70
FFFFF140H	TMIC80	TMIF80	TMMK80	0	0	0	TMPR802	TMPR801	TMPR800
FFFFF142H	TMIC81	TMIF81	TMMK81	0	0	0	TMPR812	TMPR811	TMPR810
FFFFF144H	TMIC90	TMIF90	TMMK90	0	0	0	TMPR902	TMPR901	TMPR900
FFFFF146H	TMIC91	TMIF91	TMMK91	0	0	0	TMPR912	TMPR911	TMPR910
FFFFF148H	CSIC3	CSIF3	CSMK3	0	0	0	CSPR32	CSPR31	CSPR30
FFFFF14AH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10

Notes 1. Only for the V850/SC3

2. Only for the V850/SC2

Table 7-2. Interrupt Control Registers (xxICn) (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF14CH	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF14EH	DMAIC4	DMAIF4	DMAMK4	0	0	0	DMAPR42	DMAPR41	DMAPR40
FFFFF150H	DMAIC5	DMAIF5	DMAMK5	0	0	0	DMAPR52	DMAPR51	DMAPR50
FFFFF152H	CANIC4[Note]	CANIF4	CANMK4	0	0	0	CANPR42	CANPR41	CANPR40
FFFFF154H	CANIC5[Note]	CANIF5	CANMK5	0	0	0	CANPR52	CANPR51	CANPR50
FFFFF156H	CANIC6[Note]	CANIF6	CANMK6	0	0	0	CANPR62	CANPR61	CANPR60
FFFFF158H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF15AH	SRIC2	SRIF2	SRMK2	0	0	0	SRPR22	SRPR21	SRPR20
FFFFF15CH	STIC2	STIF2	STMK2	0	0	0	STPR22	STPR21	STPR20
FFFFF15EH	SRIC3	SRIF3	SRMK3	0	0	0	SRPR32	SRPR31	SRPR30
FFFFF160H	STIC3	STIF3	STMK3	0	0	0	STPR32	STPR31	STPR30
FFFFF162H	TMIC100	TMIF100	TMMK100	0	0	0	TMPR1002	TMPR1001	TMPR1000
FFFFF164H	TMIC101	TMIF101	TMMK101	0	0	0	TMPR1012	TMPR1011	TMPR1010
FFFFF168H	TMIC110	TMIF110	TMMK110	0	0	0	TMPR1102	TMPR1101	TMPR1100
FFFFF16AH	TMIC111	TMIF111	TMMK111	0	0	0	TMPR1112	TMPR1111	TMPR1110
FFFFF16CH	TMIC120	TMIF120	TMMK120	0	0	0	TMPR1202	TMPR1201	TMPR1200
FFFFF16EH	TMIC121	TMIF121	TMMK121	0	0	0	TMPR1212	TMPR1211	TMPR1210
FFFFF172H	CSIC2	CSIF2	CSMK2	0	0	0	CSPR22	CSPR21	CSPR20

Note Only for the μPD703089Y and 70F3089Y

7.3.5 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently requesting acknowledgement. When the interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt is set (1) and remains set while the interrupt is being serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request having the highest priority is automatically reset (0) by hardware. However, it is not reset (0) when execution is returned from non-maskable interrupt servicing or exception processing.

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

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 requesting acknowledgement
0	Interrupt request with priority n not acknowledged
1	Interrupt request with priority n acknowledged

Remark n: 0 to 7 (priority level)

7.3.6 ID flag

The interrupt disable flag (ID) controls the enabling and disabling of maskable interrupt requests, and is assigned to the PSW.

Figure 7-9. ID Flag

After reset: 00000020H

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

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

Note Interrupt disable flag (ID) function

ID is set (1) by the DI instruction and reset (0) by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupts and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set (1) by hardware.

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

Remark xx: Identifying name of each peripheral unit (see **Table 7-2**)

n: Peripheral unit number (see **Table 7-2**)

7.3.7 Watchdog timer mode register (WDTM)

This register can be read/written in 8- or 1-bit units (for details, refer to **CHAPTER 10 WATCHDOG TIMER**).

After reset: 00H R/W Address: FFFFF384H

Symbol	<7>	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Watchdog timer operation control
0	Count operation stopped
1	Count started after clearing

WDTM4	Timer mode selection/interrupt control by WDT
0	Interval timer mode
1	WDT mode

WDTM3	Internal reset signal generation selection
0	When overflow occurs, the internal reset signal is not generated
1	When overflow occurs, the internal reset signal is generated

Caution If the RUN, WDTM4, or WDTM3 bit is set to 1, that bit can only be cleared by reset input.

7.3.8 Noise elimination

(1) Elimination of noise from INTP0 to INTP3 and INTP7 to INTP9 pins

An on-chip noise eliminator is provided that uses analog delay to eliminate noise. Consequently, if a signal having a constant level is input for longer than a specified time, it is detected as a valid edge.

Edge detection occurs only after the specified amount of time has elapsed.

(2) Elimination of noise from INTP4 and INTP5 pins

A digital noise eliminator is provided on chip. 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 noise. Note the following:

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

(3) Elimination of noise from INTP6 pin

A digital noise eliminator is provided on chip. 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.

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- 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 is input within these 3 clocks, an interrupt request may occur. Therefore, observe the following points 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.**

(a) Noise elimination control register (NCC)

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	Sampling clock	Reliably eliminated noise width[Note]		
				f_{XX} = 20 MHz	f_{XX} = 18.87 MHz	f_{XX} = 16 MHz
0	0	0	f_{XX}	100.0 ns	105.0 ns	125.0 ns
0	0	1	$f_{XX}/64$	6.4 μs	6.7 μs	8.0 μs
0	1	0	$f_{XX}/128$	12.8 μs	13.5 μs	16.0 μs
0	1	1	$f_{XX}/256$	25.6 μs	27.1 μs	32.0 μs
1	0	0	$f_{XX}/512$	51.2 μs	54.2 μs	64.0 μs
1	0	1	$f_{XX}/1024$	102.4 μs	108.5 μs	128.0 μs
1	1	0	Setting prohibited			
1	1	1	f_{XT}	61 μs		

Note Since sampling is performed three times, the reliably eliminated noise width is 2 × sampling clock.

7.3.9 Edge detection function

Valid edges of the INTP0 to INTP9 pins can be selected for each pin from the following four types.

- Rising edge
- Falling edge
- Both rising and falling edges
- Neither rising nor falling edge detected

The validity of the rising edge is controlled by rising edge specification register n (EGPn), and the validity of the falling edge is controlled by falling edge specification register n (EGNn) (n = 0, 1). These can be read/written in 8- or 1-bit units. Refer to **7.2.5 (1) Format of rising edge specification register 0 (EGP0)** and **7.2.5 (2) Format of falling edge specification register 0 (EGN0)** for details of EGP0 and EGN0 and **(1) Format of rising edge specification register 1 (EGP1)** and **(2) Format of falling edge specification register 1 (EGN1)** for details of EGP1 and EGN1.

After reset, the valid edges of the INTP0 to INTP9 pins are set to the “neither rising nor falling edge detected” state. Therefore, the NMI pin functions as a normal port and interrupt requests cannot be acknowledged, unless a valid edge is specified by using the EGP0 and EGN0 registers.

When using P01 to P07 or P35 to P37 as output ports, set the valid edges of INTP0 to INTP6 or INTP7 to INTP9 to “neither rising nor falling edge detected” or mask interrupt requests.

(1) Format of rising edge specification register 1 (EGP1)

After reset: 00H R/W Address: FFFFF0C4H

Symbol	<7>	<6>	<5>	4	3	2	1	0
EGP1	EGP17	EGP16	EGP15	0	0	0	0	0

EGP1n	Rising edge validity control
0	No interrupt request signal occurs at the rising edge
1	Interrupt request signal occurs at the rising edge

n = 5 to 7: Control of INTP7 to INTP9 pins

(2) Format of falling edge specification register 1 (EGN1)

After reset: 00H R/W Address: FFFFF0C6H

Symbol	<7>	<6>	<5>	4	3	2	1	0
EGN1	EGN17	EGN16	EGN15	0	0	0	0	0

EGN1n	Falling edge validity control
0	No interrupt request signal occurs at the falling edge
1	Interrupt request signal occurs at the falling edge

n = 5 to 7: Control of INTP7 to INTP9 pins

7.4 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.**

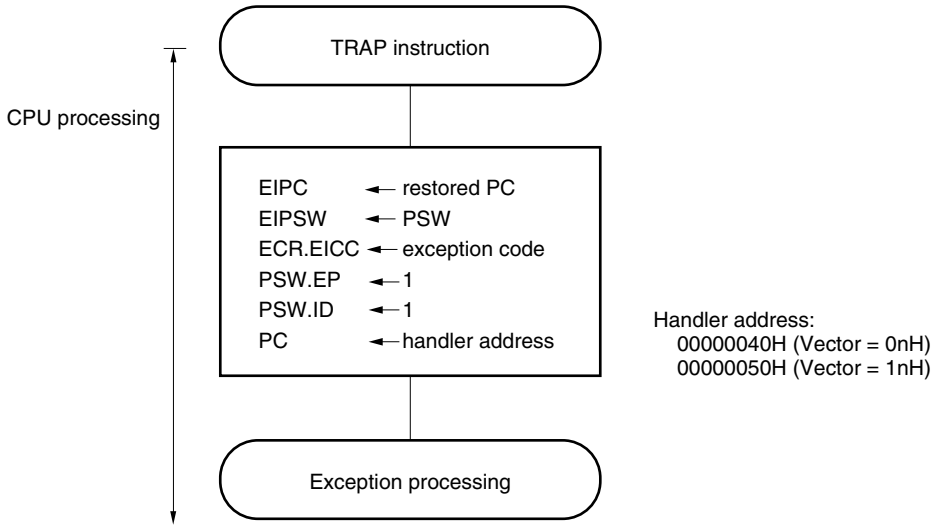
7.4.1 Operation

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 7-10. Software Exception Processing



7.4.2 Restore

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

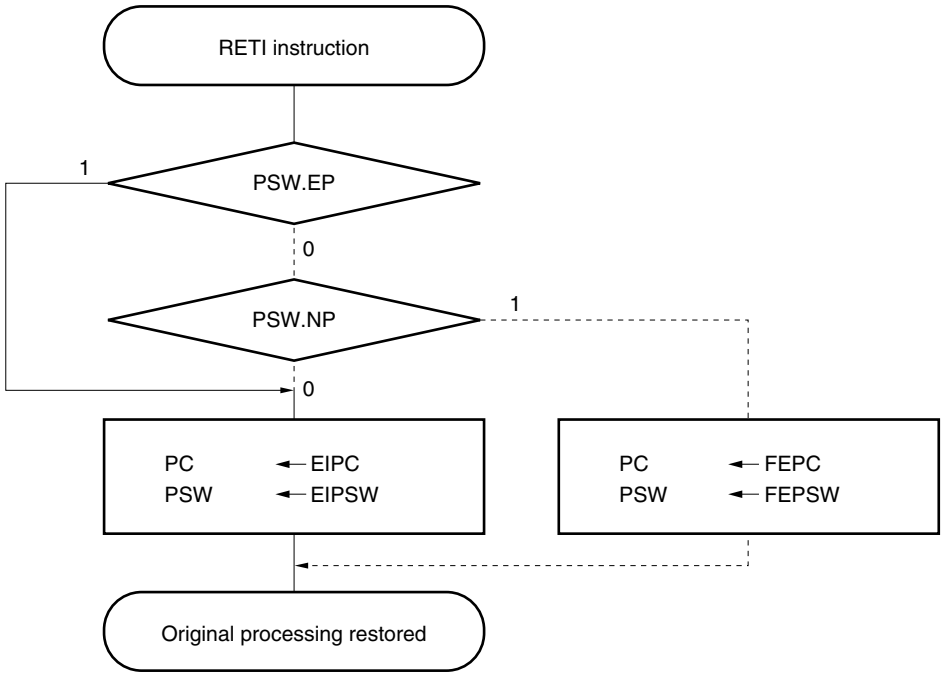
Operation of RETI instruction

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 7-11. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

7.4.3 EP flag

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

Figure 7-12. EP Flag (EP)

After reset: 00000020H

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

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

7.5 Exception Trap

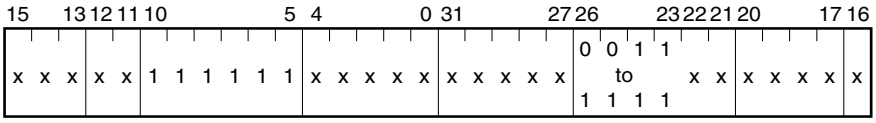
The exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850/SC1, V850/SC2, and V850/SC3, an illegal op code exception (ILGOP: ILeGal OPcode trap) is considered as an exception trap.

- Illegal op code exception: occurs if the sub op code field of an instruction to be executed next is not a valid op code.

7.5.1 Illegal op code definition

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

Figure 7-13. Illegal Op Code



x: don't care

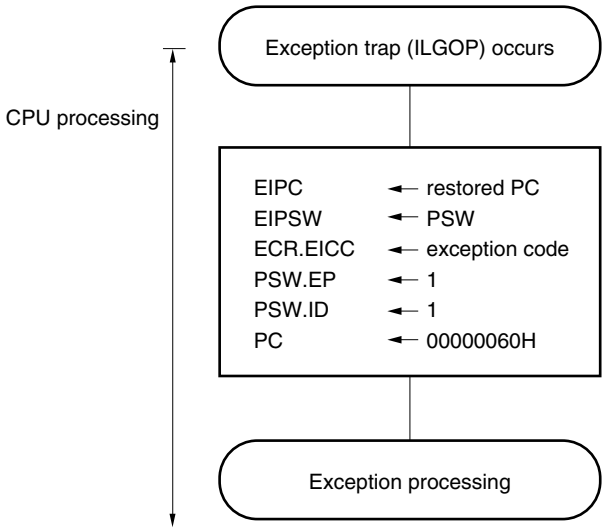
7.5.2 Operation

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 7-14. Exception Trap Processing



7.5.3 Restore

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

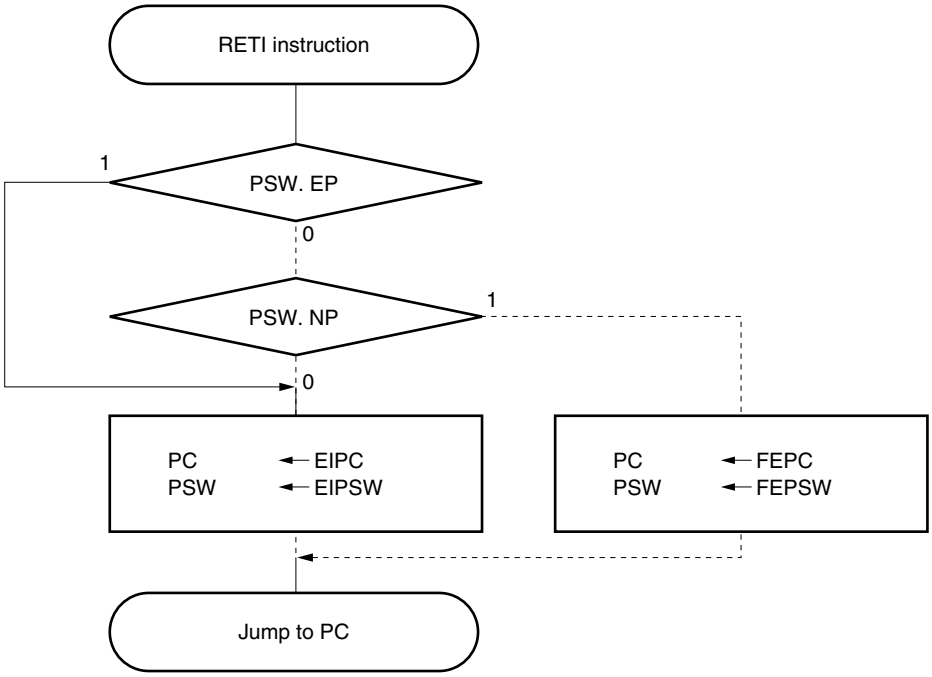
Operation of RETI instruction

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 7-15. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during exception trap processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

7.6 Priority Control

7.6.1 Priorities of interrupts and exceptions

Table 7-3. 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 op code exception
*: Item on the left ignores the item above.
×: Item on the left is ignored by the item above.
↑: Item above is higher than the item on the left in priority.
←: Item on the left is higher than the item above in priority.

7.6.2 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 an interrupt service routine currently in progress. Execution of the original routine will resume once the higher priority interrupt routine is completed.

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

Multiple interrupt servicing control is performed when interrupts are enabled (ID = 0). Even in an interrupt servicing routine, multiple interrupt control must be performed while interrupts are enabled (ID = 0). If a maskable interrupt or exception is generated in a maskable interrupt or exception service program, 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

...	
...	
• EIPC saved to memory or register	
• EIPSW saved to memory or register	
• EI instruction (enables interrupt acknowledgement)	
...	← Acknowledges interrupts such as INTP input.
...	
• DI instruction (disables interrupt acknowledgement)	
• Saved value restored to EIPSW	
• Saved value restored to EIPC	
• RETI instruction	

(2) To generate exception in service program

Service program of maskable interrupt or exception

...	
...	
• EIPC saved to memory or register	
• EIPSW saved to memory or register	
• EI instruction (enables interrupt acknowledgement)	
...	
• TRAP instruction	← Acknowledges exceptions such as TRAP instruction.
• Illegal op code	← Acknowledges exceptions such as illegal op code.
...	
• Saved value restored to EIPSW	
• Saved value restored to EIPC	
• RETI instruction	

Priorities 0 to 7 (0 is the highest) can be programmed for each maskable interrupt request in 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, interrupt requests are masked by the xxMKn bit, and the priority level is set to 7 by the xxPRn0 to xxPRn2 bits.

Remark xx: Identifying name of each peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

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.

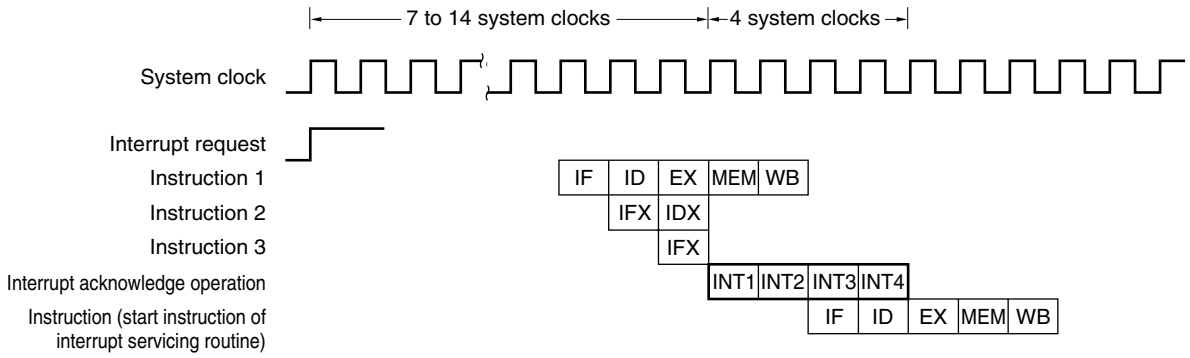
A pending interrupt request is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

Caution In a non-maskable interrupt servicing routine (in the time until the RETI instruction is executed), maskable interrupts are not acknowledged and held pending.

7.7 Response Time

The following table describes the interrupt response time (from interrupt request generation to start of interrupt servicing).

Figure 7-16. Pipeline Operation at Interrupt Request Acknowledgement



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

Interrupt response time (system clock)			Conditions
	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 • Access to interrupt control register
Maximum	18	20	

7.8 Periods in Which Interrupts Are Not Acknowledged

Interrupts are 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)

★ 7.8.1 Interrupt request valid timing following EI instruction

When an interrupt request is generated (IF flag = 1) in the status in which interrupts have been disabled by the DI instruction and interrupts are not masked (MK flag = 0), 7 system clocks are required until the interrupt request is acknowledged following execution of the EI instruction (interrupt enable). If the DI instruction (interrupt disable) is executed during the 7 system clocks, the interrupt request is not acknowledged by the CPU.

Therefore, instructions equivalent to 7 system clocks must be inserted as the number of instruction execution clocks after executing the EI instruction (interrupt enable). However, securing 7 system clocks is disabled under the following conditions because an interrupt request is not acknowledged even if 7 system clocks are secured.

- IDLE/STOP mode
- Interrupt request non-sampling instruction (instruction to manipulate PSD.ID bit)
- Access to interrupt request control register (xxICn)

The following shows an example of program processing.

[Program processing example]

```
DI
 :              ;(MK flag = 0)
 :              ;← Interrupt request generated (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 (branched to LP1 routine)       ┘
 :
LP1             ;LP1 routine
DI              ;After EI instruction executed, executed at the 8th clock by NOP
                 x 4 and JR instructions
```

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

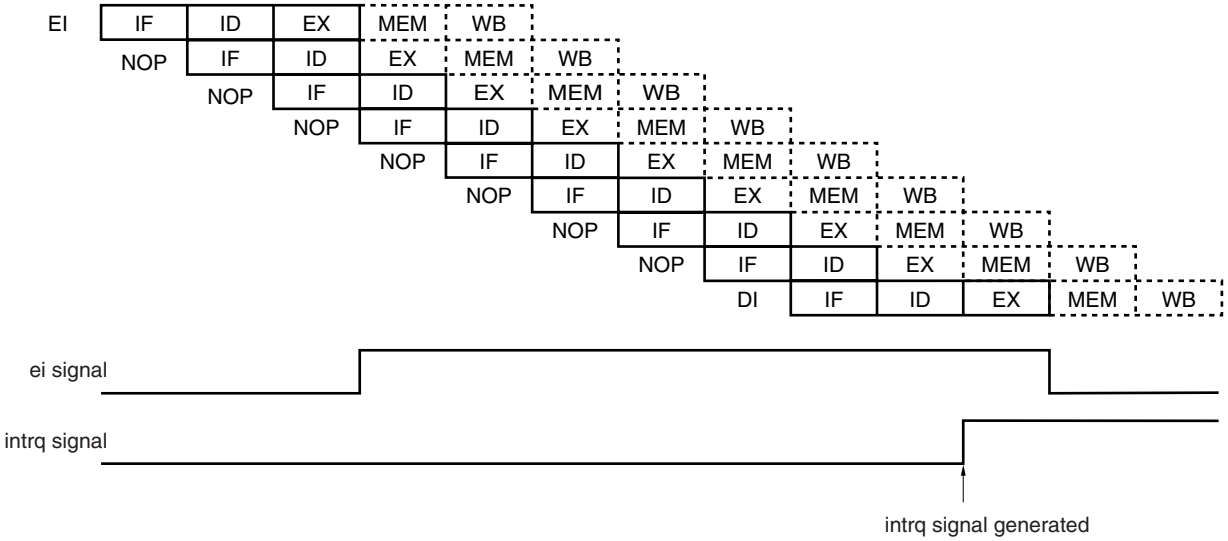
Remarks 1. In this example, the DI instruction is executed at the 8th clock after EI instruction execution, so an interrupt request is acknowledged by the CPU and the interrupt is serviced.

2. This timing does not imply that the interrupt servicing routine instruction is executed at the 8th clock after EI instruction. The interrupt servicing routine instruction is executed 4 system clocks after interrupt request acknowledgement by the CPU.

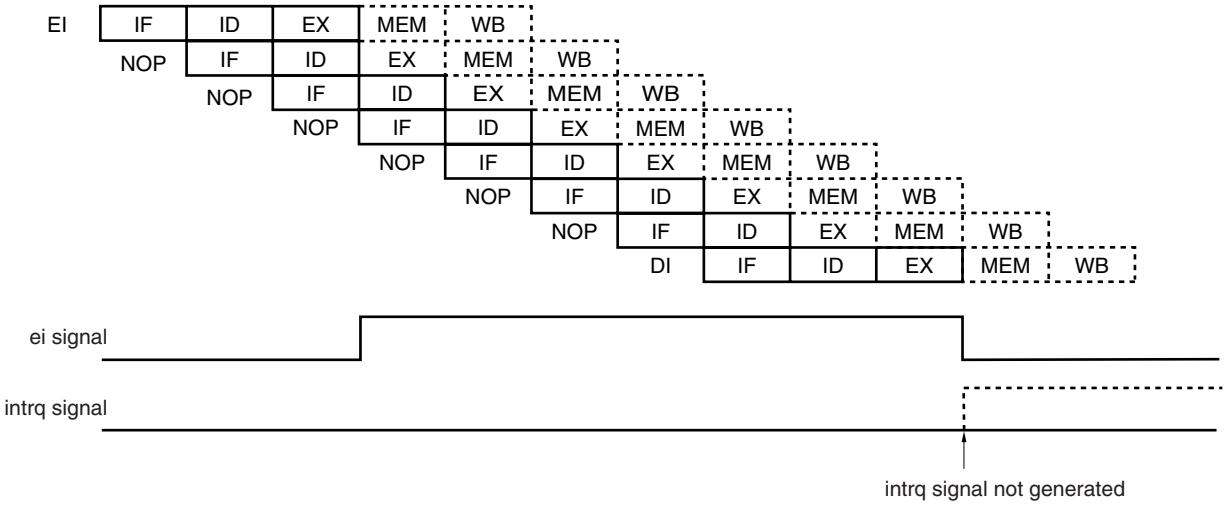
3. This example indicates the case where an interrupt request is generated (IF flag = 1) before the EI instruction is executed. In the case where an interrupt request is generated (IF flag = 1) after the EI instruction is executed, the interrupt request is also not acknowledged by the CPU if interrupts are disabled (PSW.ID = 1) within 7 system clocks after the IF flag is set (1).

Figure 7-17. Pipeline Flow and Interrupt Request Generation Timing

(a) When DI instruction is executed at 8th system clock after EI instruction execution (interrupt request is acknowledged)



(b) When DI instruction is executed at 7th system clock after EI instruction execution (interrupt request is not acknowledged)



★ 7.9 Bit Manipulation Instruction of Interrupt Control Register on DMA Transfer

When using the DMA function, execute the DI instruction before performing bit manipulation of the interrupt control register (xxICn) in the EI status and execute the EI instruction after performing manipulation. Alternately, clear (0) the xxIF bit at the start of the interrupt servicing routine.

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

Remark xx: Identifying name of each peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

7.10 Key Interrupt Function

A key interrupt can be generated by inputting a falling edge to 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- or 1-bit units.

After reset: 00H R/W Address: FFFFF3D0H

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

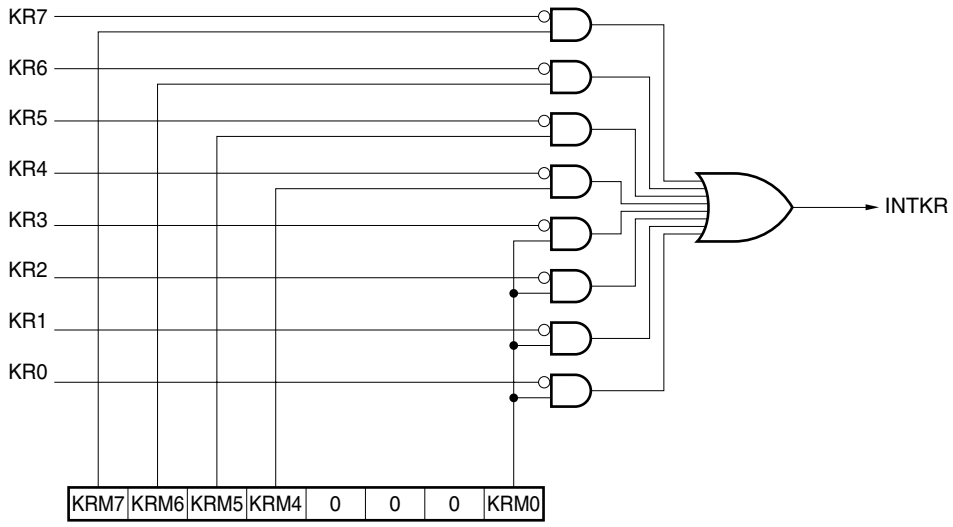
KRMn	Key return mode control
0	Does not detect key return signal
1	Detects key return signal

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 7-4. 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 7-18. Block Diagram of Key Return



Key return mode register (KRM)

CHAPTER 8 TIMER/COUNTER FUNCTION

8.1 16-Bit Timer (TM0, TM1, TM7 to TM12)

8.1.1 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 specifications
- Timer output operated by match detection: 1 each (TOn)
  When using the P104/TO0, P107/TO1, and P100/TO7 pins as TO0, TO1, and TO7 (timer output), set the value of port 10 (P10) to 0 (port mode output) and the port 10 mode register (PM10) to 0.
  When using the P33/TO8 pins as TO8 (timer output), set the value of port 3 (P3) to 0 (port mode output) and the port 3 mode register (PM3) to 0.
  When using the P25/TO9 pins as TO9 (timer output), set the value of port 2 (P2) to 0 (port mode output) and the port 2 mode register (PM2) to 0.
  When using the P126/TO10 and P127/TO11 pins as TO10 and TO11 (timer output), set the value of port 12 (P12) to 0 (port mode output) and the port 12 mode register (PM12) to 0.
  When using the P155/TO12 pins as TO12 (timer output), set the value of port 15 (P15) to 0 (port mode output) and the port 15 mode register (PM15) to 0.
  The ORed value of the output of the port and the timer is output.

  Remark n = 0, 1, 7 to 12

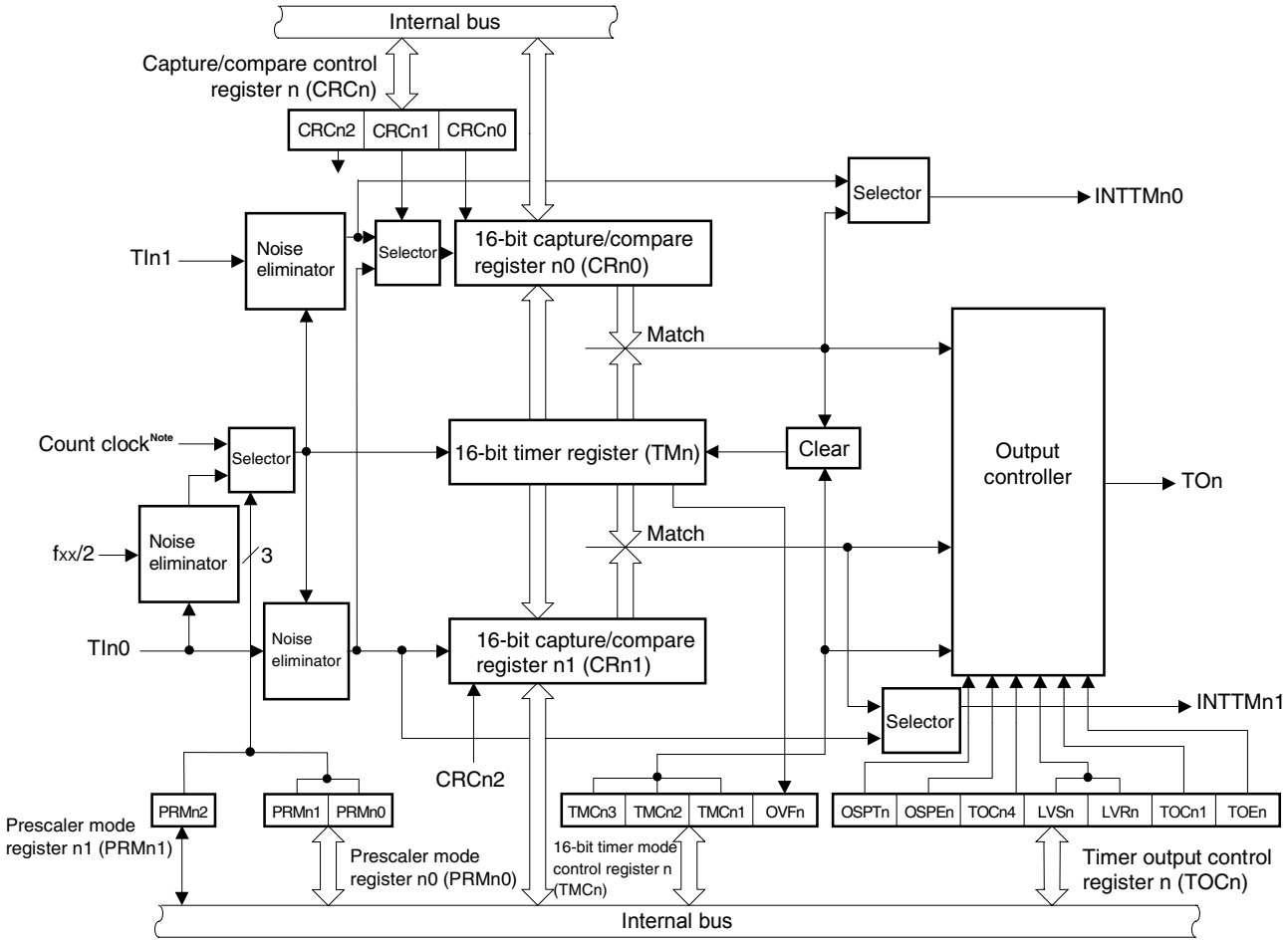
8.1.2 Function

TM0, TM1, and TM7 to TM12 have the following functions:

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

The following shows the block diagram.

Figure 8-1. Block Diagram of TM0, TM1, and TM7 to TM12



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

Remark n = 0, 1, 7 to 12

(1) Interval timer

Generates an interrupt at preset time intervals.

(2) PPG output

Can output a square wave with a frequency and output-pulse width that can be set 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.

8.1.3 Configuration

Timers 0, 1, and 7 to 12 include the following hardware.

Table 8-1. Configuration of Timers 0, 1, and 7 to 12

Item	Configuration
Timer registers	16 bits × 8 (TM0, TM1, TM7 to TM12)
Registers	Capture/compare registers: 16 bits × 6 (CRn0, CRn1)
Timer outputs	8 (TO0, TO1, TO7 to TO12)
Control registers	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)

Remark n = 0, 1, 7 to 12

(1) 16-bit timer registers 0, 1, 7 to 12 (TM0, TM1, TM7 to TM12)

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 (n = 0, 1, 7 to 12):

<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 on a match between TMn and CRn0
<5> If OSPTn is set or if the valid edge of TIn0 is input in the one-shot pulse output mode

(2) Capture/compare register n0 (CR00, CR10, CR70 to CR120)

CRn0 is a 16-bit register that functions as both a capture register and a compare register. Whether this register functions as a capture or compare register is specified by using bit 0 (CRCn0) of the CRCn register (n = 0, 1, 7 to 12).

(a) When using CRn0 as compare register

The value set to CRn0 is continually 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 the register that holds the interval time (n = 0, 1, 7 to 12).

(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 for TIn0 or TIn1 is set by using the PRMn0 register.

When the valid edge for the TIn0 pin is specified as the capture trigger, refer to **Table 8-2**. When the valid edge for the TIn1 pin is specified as the capture trigger, refer to **Table 8-3** (n = 0, 1, 7 to 12).

Table 8-2. 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, 7 to 12

Table 8-3. 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, 7 to 12

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

★ When used as a compare register, CRn0 can be read/written, but when used as a capture register, CRn0 can only be read.

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

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

(3) Capture/compare register n1 (CR01, CR11, CR71 to CR121)

This is a 16-bit register that can be used as both a capture register and a compare register. Whether it is used as a capture register or compare register is specified by bit 2 (CRCn2) of the CRCn register (n = 0, 1, 7 to 12).

(a) When using CRn1 as compare register

The value set to CRn1 is continually compared with the count value of TMn. When the values of the two match, an interrupt request (INTTMn1) is generated (n = 0, 1, 7 to 12).

(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 using 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 8-4. Valid Edge of TIn0 Pin and Capture Trigger of CRn1

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, 7 to 12

CRn1 is set using a 16-bit memory manipulation instruction.

★ When used as a compare register, CRn1 can be read/written, but when used as a capture register, CRn1 can only be read.

★ $\overline{\text{RESET}}$ input sets these registers to 0000H.

Caution In the clear & start mode entered on a match between TMn and CRn1, set CRn1 to a value other than 0000H. In the free-running mode or the TIn0 valid edge clear mode, however, an interrupt request (INTTMn1) is generated after an overflow (FFFFH) when CRn1 is set to 0000H.

8.1.4 Timer 0, 1, 7 to 12 control registers

Timers 0, 1, and 7 to 12 are controlled by the following registers.

- 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)

Remark n = 0, 1, 7 to 12

(1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)

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

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

$\overline{\text{RESET}}$ input clears TMC0, TMC1, and TMC7 to TMC12 to 00H.

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

After reset: 00H R/W Address: TMC0: FFFFF208H TMC1: FFFFF218H TMC7: FFFFF3A8H
TMC8: FFFFF398H TMC9: FFFFF3B8H TMC10: FFFFF0D8H
TMC11: FFFFF0E8H TMC12: FFFFF0F8H

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

(n = 0, 1, 7 to 12)

TMCn3	TMCn2	TMCn1	Selects operation mode and clear mode	Selects TOn output timing	Generation of interrupt
0	0	0	Operation stops (TMn is cleared to 0)	Not affected	Does not generate
0	0	1			
0	1	0	Free-running mode	Match between TMn and CRn0 or match between TMn and CRn1	Generates 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	Does not overflow
1	Overflows

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

2. The valid edge of the TIn0 pin is set 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 flag is set to 1 when the count value of TMn changes from FFFFH to 0000H with CRn0 set to FFFFH.

★ **4. Always set bits 7 to 4 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, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)

CRCn controls the operation of 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, CRC1, and CRC7 to CRC12 to 00H.

After reset: 00H R/W Address: CRC0: FFFFF20AH CRC1: FFFFF21AH CRC7: FFFFF3AAH CRC8: FFFFF39AH CRC9: FFFFF3BAH CRC10: FFFFF0DAH CRC11: FFFFF0EAH CRC12: FFFFF0FAH

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

(n = 0, 1, 7 to 12)

CRCn2	Selects operation mode of CRn1
0	Operates as compare register
1	Operates as capture register

CRCn1	Selects capture trigger of CRn0
0	Captured at valid edge of TIn1
1	Captured in reverse phase of valid edge of TIn0

CRCn0	Selects operation mode of CRn0
0	Operates as compare register
1	Operates as capture register

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

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

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

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

★ **5. Always set bits 7 to 3 to 0.**

(3) 16-bit timer output control registers 0, 1, 7 to 12 (TOC0, TOC1, TOC7 to TOC12)

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 an output trigger for a one-shot pulse by software.

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

$\overline{\text{RESET}}$ input clears TOC0, TOC1, and TOC7 to TOC12 to 00H.

After reset: 00H R/W

Address: TOC0: FFFFF20CH TOC1: FFFFF21CH TOC7: FFFFF3ACH
TOC8: FFFFF39CH TOC9: FFFFF3BCH TOC10: FFFFF0DCH
TOC11: FFFFF0ECH TOC12: FFFFF0FCH

	7	<6>	<5>	4	<3>	<2>	1	<0>
TOCn	0	OSPTn	OSPEn	TOCn4	LVSn	LVRn	TOCn1	TOEn

(n = 0, 1, 7 to 12)

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

OSPEn	Controls one-shot pulse output operation
0	Successive pulse output
1	One-shot pulse output[Note]

TOCn4	Controls timer output F/F on match between CRn1 and TMn
0	Disables reverse timer output F/F
1	Enables reverse timer output F/F

LVSn	LVRn	Sets 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	Controls timer output F/F on match between CRn0 and TMn or valid edge of TIn0
0	Disables reverse timer output F/F
1	Enables reverse timer output F/F

TOEn	Controls output of timer n
0	Disables output (output is fixed to 0 level)
1	Enables output

Note The one-shot pulse output operates only in the free-running mode and in the clear & start mode entered upon 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 (1) for other than one-shot pulse output.

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

PRM0n selects the count clock of the 16-bit timer (TM0) and the valid edges of the TI00 and TI01 inputs. 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: FFFFF20EH

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

After reset: 00H R/W Address: FFFFF206H

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

ES011	ES010	Selects 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	Selects valid edge of TI00
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM02	PRM01	PRM00	Count clock selection			
			Count clock	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$f_{XX}/2$	100 ns	0.5 ns	125 ns
0	0	1	$f_{XX}/16$	800 ns	848 μs	1 μs
0	1	0	INTWTNI	–	–	–
0	1	1	TI00 valid edge[Note]	–	–	–
1	0	0	$f_{XX}/4$	200 ns	212 ns	250 ns
1	0	1	$f_{XX}/64$	3.2 μs	3.4 μs	4 μs
1	1	0	$f_{XX}/256$	12.8 μs	13.6 μs	16 μs
1	1	1	Setting prohibited	–	–	–

Note A pulse longer than twice the internal clock ($f_{XX}/2$) is required for an external clock.

Cautions 1. When selecting the valid edge of TI00 as the count clock, do not specify the valid edge of TI00 to clear and start the timer and as a capture trigger.

2. Before setting data to PRM00 and PRM01, 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 TI00 pin while the TI00 or TI01 pin is high level immediately after system reset, the rising edge is detected immediately after the rising edge or both edges is specified. Care is therefore needed when pulling up the TI00 or TI01 pin. However, the rising edge is not detected when operation is enabled after it has been stopped.

(5) Prescaler mode registers 10, 11, 70, 71 (PRM10, PRM11, PRM70, PRM71)

PRM1n selects the count clock of the 16-bit timer (TM1, TM7) and the valid edge of the TIn0 and TIn1 inputs. PRMn0 and PRMn1 are set by an 8-bit memory manipulation instruction (n = 1, 7).

$\overline{\text{RESET}}$ input clears PRMn0 and PRMn1 to 00H.

After reset: 00H R/W Address: FFFFF21EH, FFFFF3AEH

	7	6	5	4	3	2	1	0
PRMn1	0	0	0	0	0	0	0	PRMn2

(n = 1, 7)

After reset: 00H R/W Address: FFFFF216H, FFFFF3A6H

	7	6	5	4	3	2	1	0
PRMn0	ESn11	ESn10	ESn01	ESn00	0	0	PRMn1	PRMn0

(n = 1, 7)

ESn11	ESn10	Selects valid edge of TIn1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ESn01	ESn00	Selects valid edge of TIn0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRMn2	PRMn1	PRMn0	Count clock selection			
			Count clock	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$f_{XX}/2$	100 ns	105 ns	125 ns
0	0	1	$f_{XX}/4$	200 ns	212 ns	250 ns
0	1	0	$f_{XX}/16$	800 ns	848 ns	1 μs
0	1	1	TIn0 valid edge[Note]	–	–	–
1	0	0	$f_{XX}/32$	1.6 μs	1.7 μs	2 μs
1	0	1	$f_{XX}/128$	6.4 μs	6.8 μs	8 μs
1	1	0	$f_{XX}/256$	12.8 μs	13.6 μs	16 μs
1	1	1	Setting prohibited	–	–	–

Note A pulse longer than twice the internal clock ($f_{XX}/2$) is required for an external clock.

Cautions 1. When selecting the valid edge of TIn0 as the count clock, do not specify the valid edge of TIn0 to clear and start the timer and as a capture trigger.

2. Before setting data to PRMn0 and PRMn1, always stop the timer operation.

3. If the 16-bit timer (TM1, TM7) operation is enabled by specifying the rising edge or both edges as the valid edge of the TIn0 or TIn1 pin while the TIn0 or TIn1 pin is high level immediately after system reset, the rising edge is detected immediately after the rising edge or both edges is specified. Care is therefore needed when pulling up the TIn0 or TIn1 pin. However, the rising edge is not detected when operation is enabled after it has been stopped (n = 1, 7).

(6) Prescaler mode registers 80, 81, 100, 101, 120, 121 (PRM80, PRM81, PRM100, PRM101, PRM120, PRM121)

PRM1n selects the count clock of the 16-bit timer (TM8, TM10, TM12) and the valid edge of the TIn0 and TIn1 inputs. PRMn0 and PRMn1 are set by an 8-bit memory manipulation instruction (n = 8, 10, 12).

$\overline{\text{RESET}}$ input clears PRMn0 and PRMn1 to 00H.

After reset: 00H R/W Address: FFFFF39EH, FFFFF0DEH, FFFFF0FEH

	7	6	5	4	3	2	1	0
PRMn1	0	0	0	0	0	0	0	PRMn2

(n = 8, 10, 12)

After reset: 00H R/W Address: FFFFF396H, FFFFF0D6H, FFFFF0F6H

	7	6	5	4	3	2	1	0
PRMn0	ESn11	ESn10	ESn01	ESn00	0	0	PRMn1	PRMn0

(n = 8, 10, 12)

ESn11	ESn10	Selects valid edge of TIn1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ESn01	ESn00	Selects valid edge of TIn0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRMn2	PRMn1	PRMn0	Count clock selection			
			Count clock	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$f_{XX}/2$	100 ns	105 ns	125 ns
0	0	1	$f_{XX}/4$	400 ns	424 ns	250 ns
0	1	0	$f_{XX}/16$	800 ns	848 ns	1 μs
0	1	1	TIn0 valid edge[Note]	–	–	–
1	0	0	$f_{XX}/32$	1.6 μs	1.7 μs	2 μs
1	0	1	$f_{XX}/128$	6.4 μs	6.8 μs	8 μs
1	1	0	$f_{XX}/256$	12.8 μs	13.6 μs	16 μs
1	1	1	Setting prohibited	–	–	–

Note A pulse longer than twice the internal clock ($f_{XX}/2$) is required for an external clock.

Cautions 1. When selecting the valid edge of TIn0 as the count clock, do not specify the valid edge of TIn0 to clear and start the timer and as a capture trigger.

2. Before setting data to PRMn0 and PRMn1, always stop the timer operation.

3. If the 16-bit timer (TM8, TM10, TM12) operation is enabled by specifying the rising edge or both edges for the valid edge of the TIn0 or TIn1 pin while the TIn0 or TIn1 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 TIn0 or TIn1 pin. However, the rising edge is not detected when operation is enabled after it has been stopped (n = 8, 10, 12).

(7) Prescaler mode registers 90, 91, 110, 111 (PRM90, PRM91, PRM110, PRM111)

PRM1n selects the count clock of the 16-bit timer (TM9, TM11) and the valid edge of the TIn0 and TIn1 inputs.

PRMn0 and PRMn1 are set by an 8-bit memory manipulation instruction (n = 9, 11).

$\overline{\text{RESET}}$ input clears PRMn0 and PRMn1 to 00H.

After reset: 00H R/W Address: FFFFF3BEH, FFFFF0EEH

	7	6	5	4	3	2	1	0
PRMn1	0	0	0	0	0	0	0	PRMn2

(n = 9, 11)

After reset: 00H R/W Address: FFFFF3B6H, FFFFF0E6H

	7	6	5	4	3	2	1	0
PRMn0	ESn11	ESn10	ESn01	ESn00	0	0	PRMn1	PRMn0

(n = 9, 11)

ESn11	ESn10	Selects valid edge of TIn1
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ESn01	ESn00	Selects valid edge of TIn0
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRMn2	PRMn1	PRMn0	Count clock selection			
			Count clock	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$f_{XX}/4$	200 ns	212 ns	250 ns
0	0	1	$f_{XX}/8$	400 ns	424 ns	500 ns
0	1	0	$f_{XX}/32$	1.6 μs	1.7 μs	2 μs
0	1	1	TIn0 valid edge[Note]	–	–	–
1	0	0	$f_{XX}/64$	3.2 μs	3.4 μs	4 μs
1	0	1	$f_{XX}/128$	6.4 μs	6.8 μs	8 μs
1	1	0	$f_{XX}/512$	25.6 μs	27.1 μs	32 μs
1	1	1	Setting prohibited	–	–	–

Note A pulse longer than twice the internal clock ($f_{XX}/2$) is required for an external clock.

Cautions 1. When selecting the valid edge of TIn0 as the count clock, do not specify the valid edge of TIn0 to clear and start the timer and as a capture trigger.

2. Before setting data to PRMn0 and PRMn1, always stop the timer operation.

3. If the 16-bit timer (TM9, TM11) operation is enabled by specifying the rising edge or both edges for the valid edge of the TIn0 or TIn1 pin while the TIn0 or TIn1 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 TIn0 or TIn1 pin. However, the rising edge is not detected when operation is enabled after it has been stopped (n = 9, 11).

8.2 16-Bit Timer (TM0, TM1, TM7 to TM12) Operation

8.2.1 Operation as interval timer

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 8-2 (n = 0, 1).

In this case, TMn repeatedly generates an interrupt at the time interval specified by the count value preset to 16-bit capture/compare register n0 (CRn0).

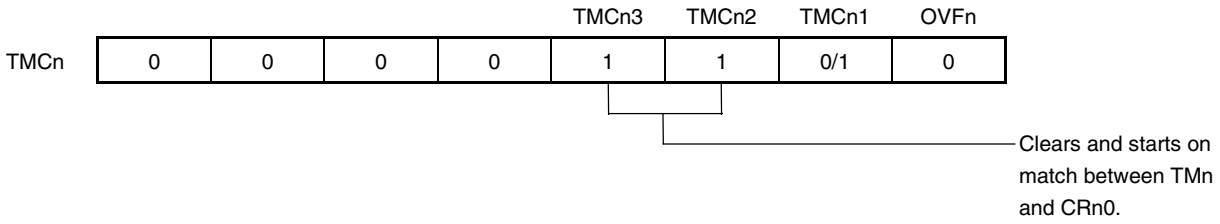
When the count value of TMn matches the set value of CRn0, the value of TMn is cleared to 0, and the timer continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.

The count clock of the 16-bit timer/event counter can be selected by bits 0 and 1 (PRMn0 and PRMn1) of prescaler mode register n0 (PRMn0) and by bits 0 (PRMn2) of prescaler mode register n1 (PRMn1).

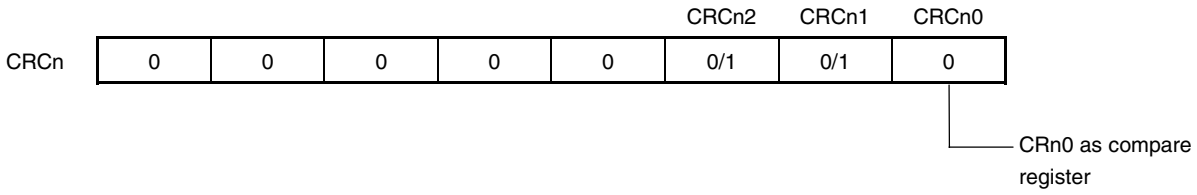
Remark n = 0, 1, 7 to 12

Figure 8-2. Control Register Settings When TMn Operates as Interval Timer

(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)

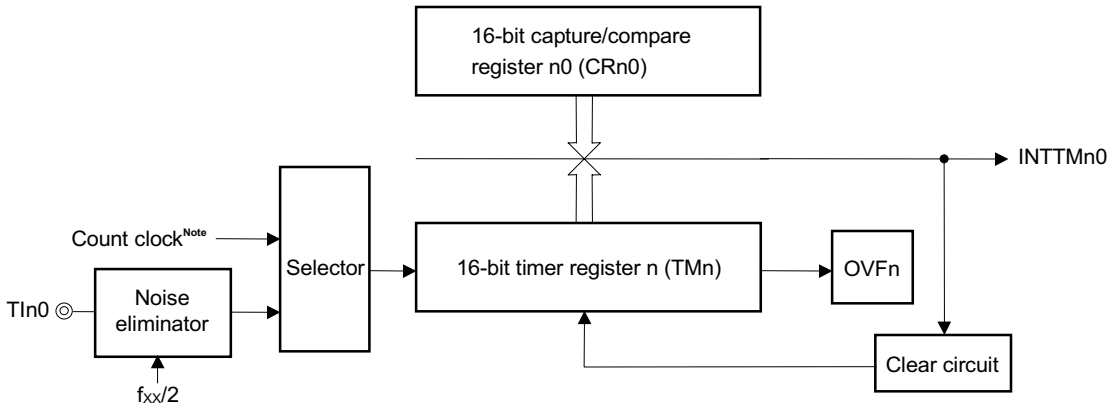


(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

Figure 8-3. Configuration of Interval Timer

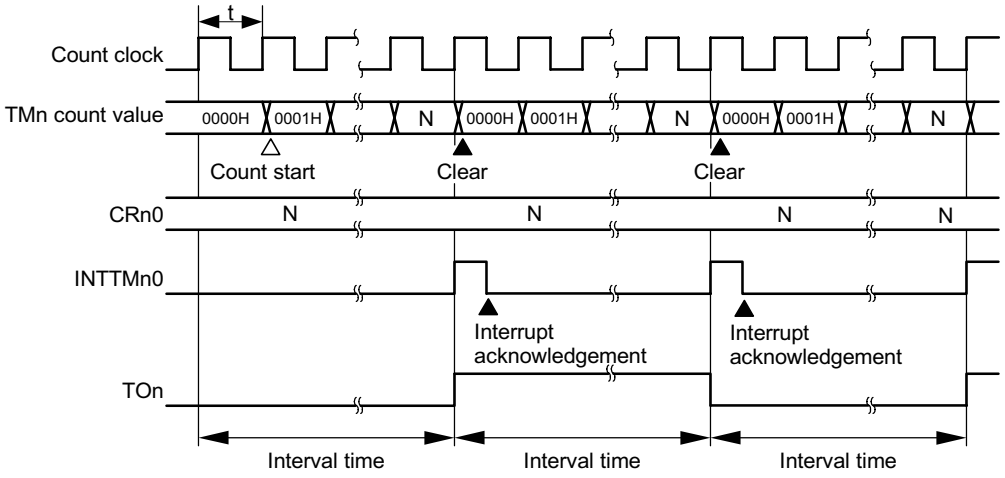


Note The count clock is set by the PRMn0 and PRMn1 registers.

Remarks 1. " ◎—— " indicates the signal that can be directly connected to ports.

2. n = 0, 1, 7 to 12

Figure 8-4. Timing of Interval Timer Operation



Remarks 1. Interval time = (N + 1) × t: N = 0001H to FFFFH

2. n = 0, 1, 7 to 12

8.2.2 PPG output operation

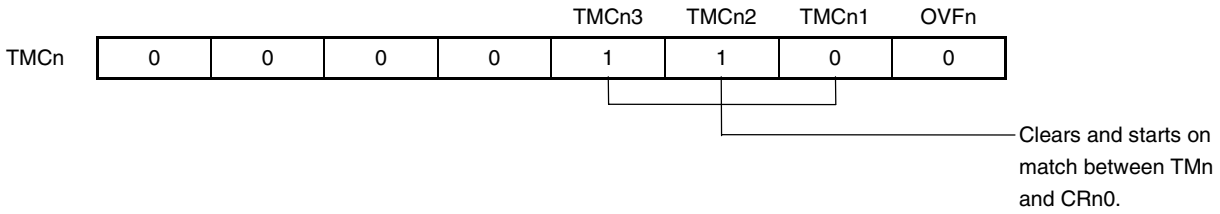
TMn can be used for PPG (Programmable Pulse Generator) output by setting 16-bit timer mode control register n (TMCn) and capture/compare control register n (CRCn) as shown in Figure 8-5.

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

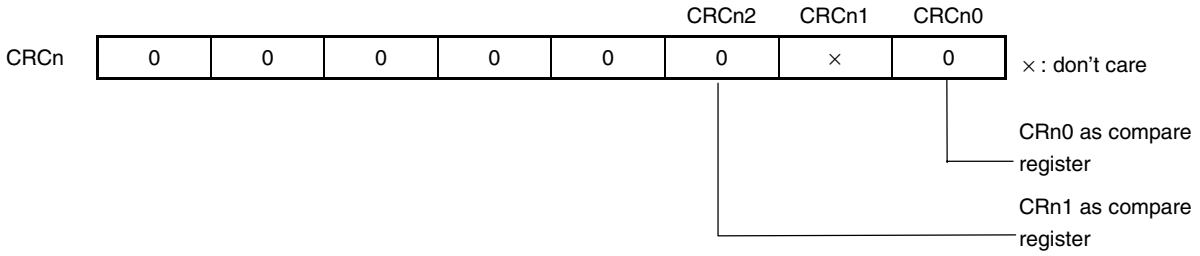
Remark n = 0, 1, 7 to 12

Figure 8-5. Control Register Settings in PPG Output Operation

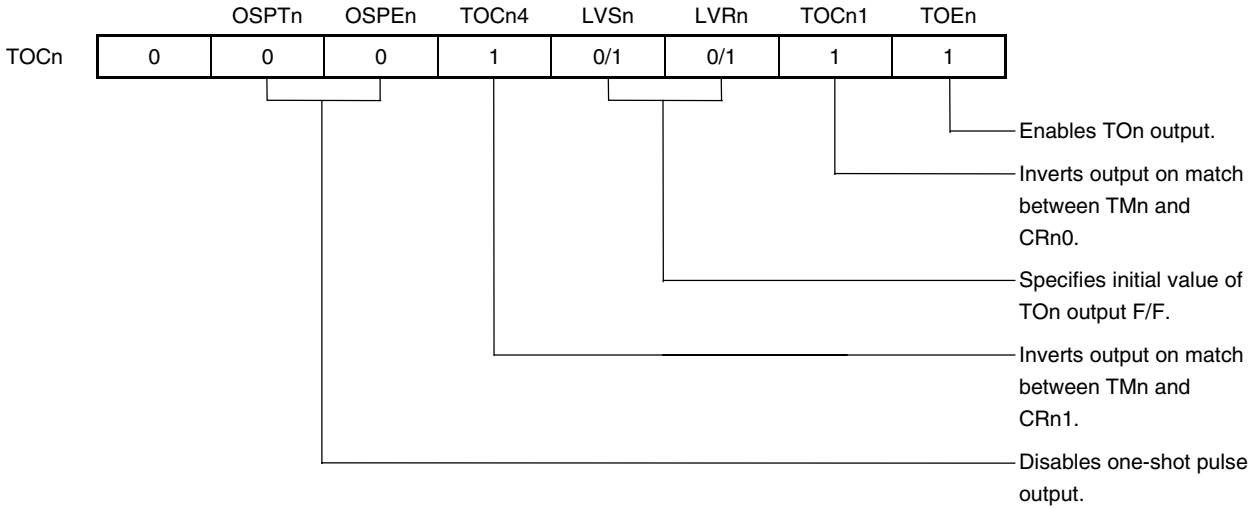
(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



(c) 16-bit timer output control registers 0, 1, 7 to 12 (TOC0, TOC1, TOC7 to TOC12)

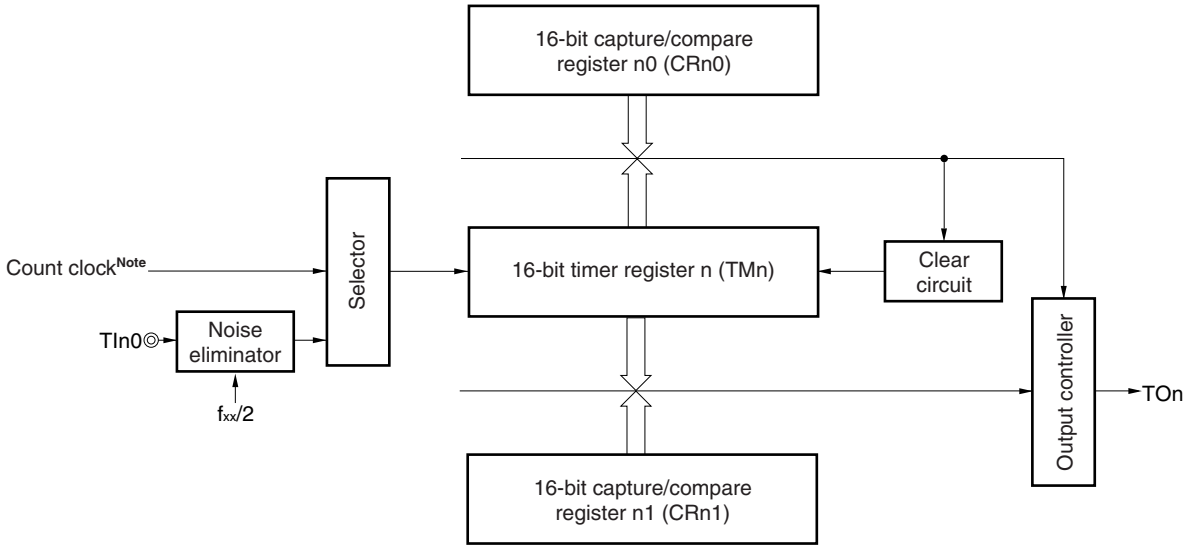


Cautions 1. Make sure that CRn0 and CRn1 are set to 0000H < CRn1 < CRn0 ≤ FFFFH.

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

Remark n = 0, 1, 7 to 12

★ **Figure 8-6. Configuration of PPG Output**

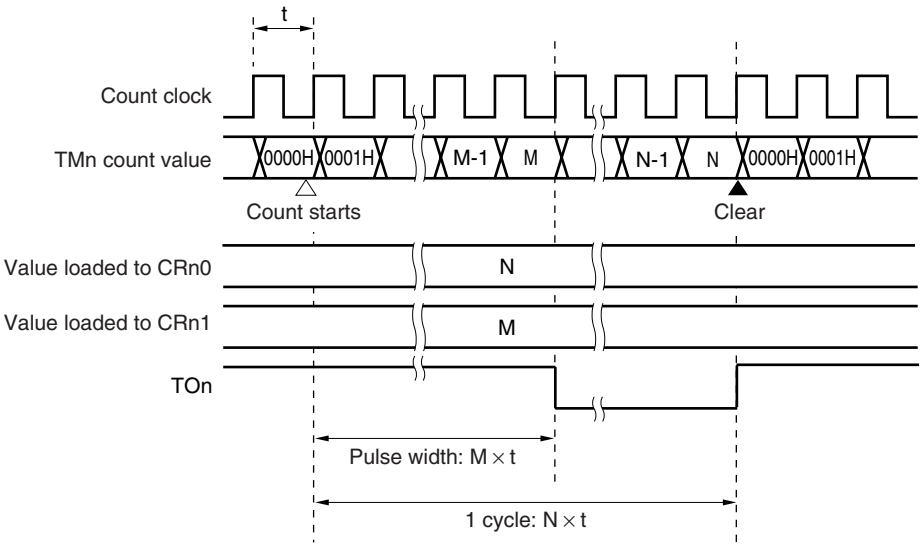


Note The count clock is set by the PRMn0 and PRMn1 registers.

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

2. n = 0, 1, 7 to 12

★ **Figure 8-7. PPG Output Operation Timing**



Remarks 1. 0000H < M < N ≤ FFFFH

2. n = 0, 1, 7 to 12

8.2.3 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

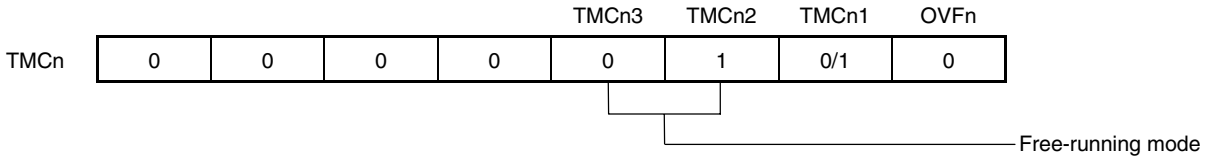
If the edge specified by prescaler mode register n0 (PRMn0) is input to the TIn0 pin when 16-bit timer register n (TMn) is used as a free-running counter (refer to **Figure 8-8**), the value of TMn is loaded to 16-bit capture/compare register n1 (CRn1), and an external interrupt request signal (INTTMn1)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

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

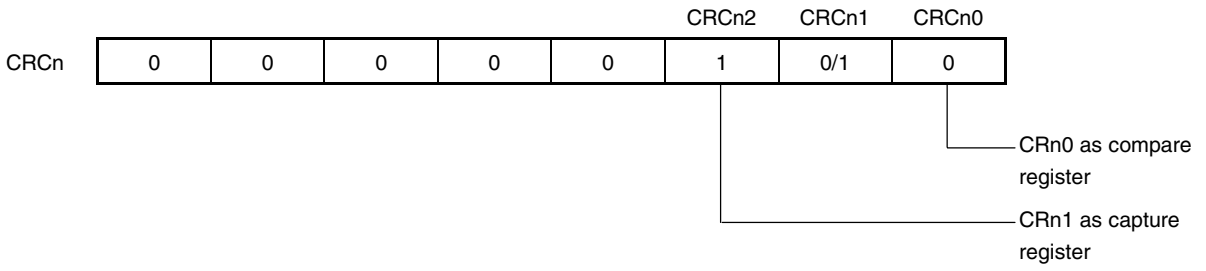
The valid edge is detected through sampling at a count clock cycle selected by prescaler mode registers n0 and n1 (PRMn0, PRMn1), and the capture operation is not performed until the valid level is detected two times, eliminating noise with a short pulse width.

Figure 8-8. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register

(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



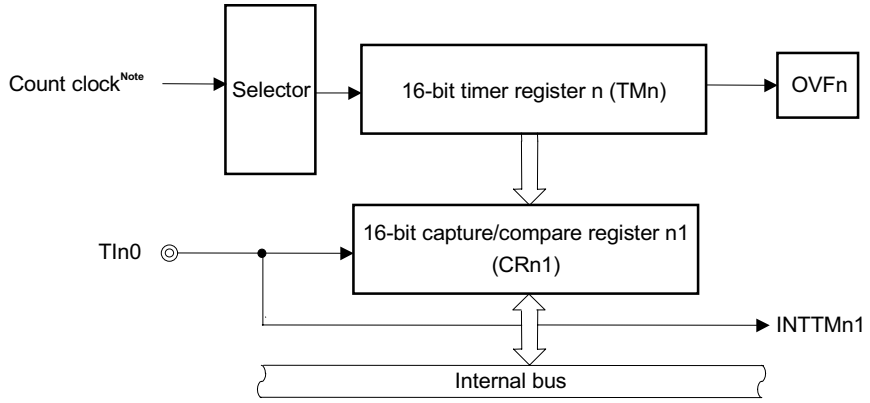
(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



Remarks 1. 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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-9. Configuration for Pulse Width Measurement with Free-Running Counter

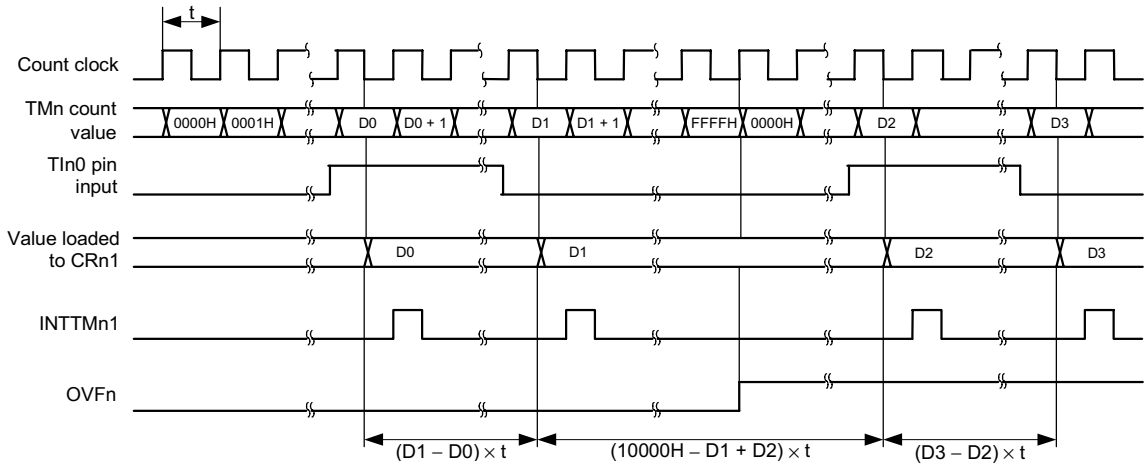


Note The count clock is set by the PRMn0 and PRMn1 registers.

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

2. n = 0, 1, 7 to 12

Figure 8-10. Timing of Pulse Width Measurement with Free-Running Counter and One Capture Register (with Both Edges Specified)



Remark n = 0, 1, 7 to 12

(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 8-11**).

When the edge specified by bits 4 and 5 (ESn00 and ESn01) 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.

When the edge specified by bits 6 and 7 (ESn10 and ESn11) of 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.

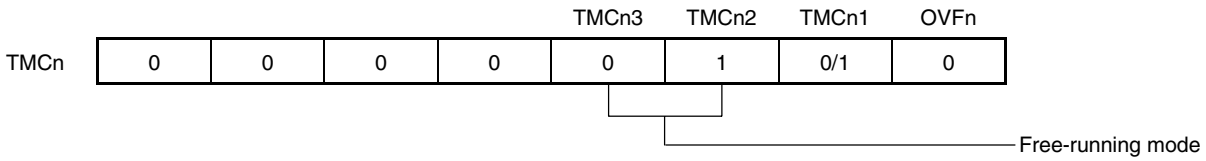
The edges of the TIn0 and TIn1 pins are specified by bits 4 and 5 (ESn00 and ESn01) and bits 6 and 7 (ESn10 and ESn11) of PRMn0, respectively. The rising, falling, or both rising and falling edges can be specified.

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

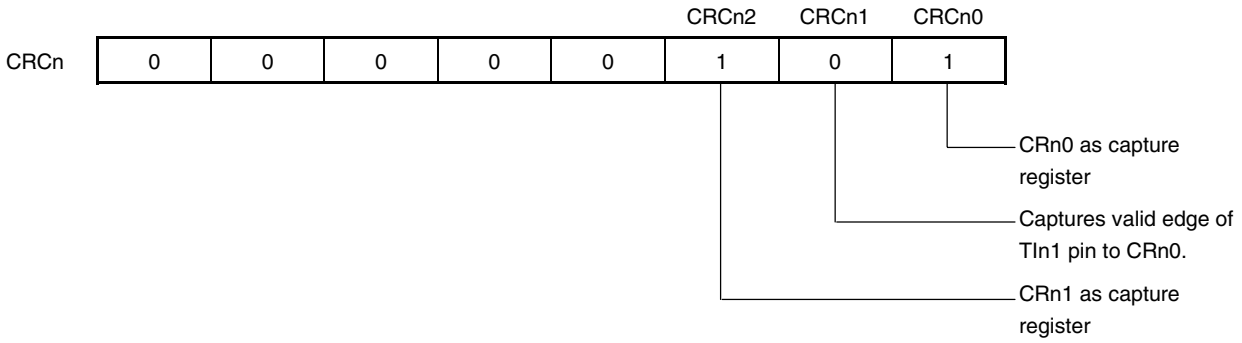
Remark n = 0, 1, 7 to 12

Figure 8-11. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter

(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



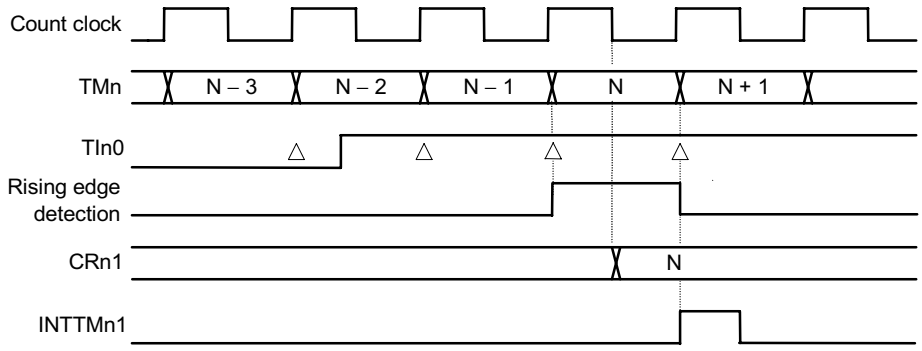
Remarks 1. 0/1: When these bits are reset to 0 or set to 1, other functions can be used along with the pulse width measurement function. For details, refer to **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

- **Capture operation (free-running mode)**

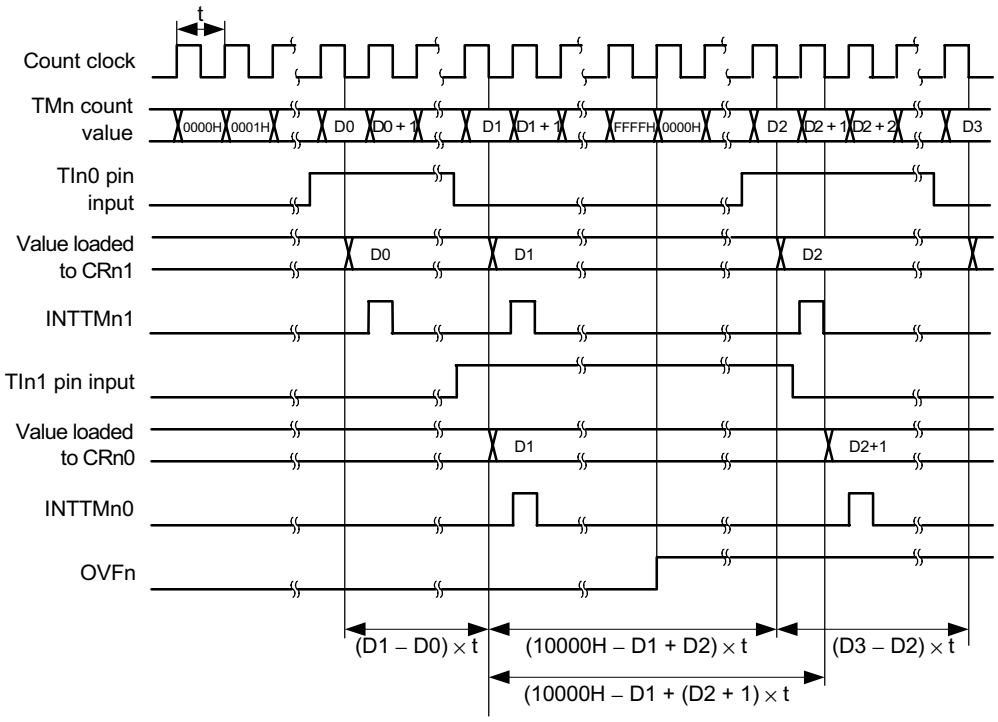
  The following figure illustrates the operation of the capture register when the capture trigger is input.

Figure 8-12. CRn1 Capture Operation with Rising Edge Specified



Remark n = 0, 1, 7 to 12

Figure 8-13. Timing of Pulse Width Measurement with Free-Running Counter (with Both Edges Specified)



Remark n = 0, 1, 7 to 12

(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 8-14**), the pulse width of the signal input to the TIn0 pin can be measured.

When the edge specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register 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 that is the reverse of the one that triggers capturing to CRn1 is input.

The edge of the TIn0 pin is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n (PRMn0). The rising or falling edge can be specified.

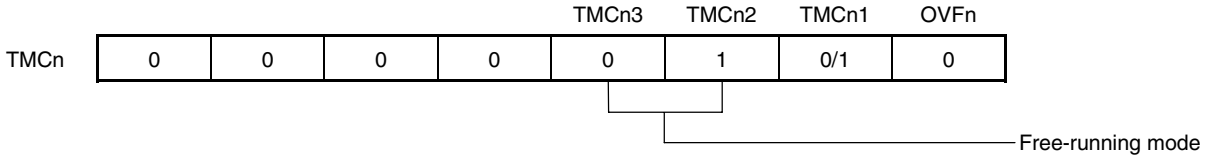
The valid edge of TIn0 is detected through sampling at a count clock cycle selected by prescaler mode registers n0 and n1 (PRMn0, PRMn1), and the capture operation is not performed until the valid level is detected two times, eliminating noise with a short pulse width.

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 a capture operation.

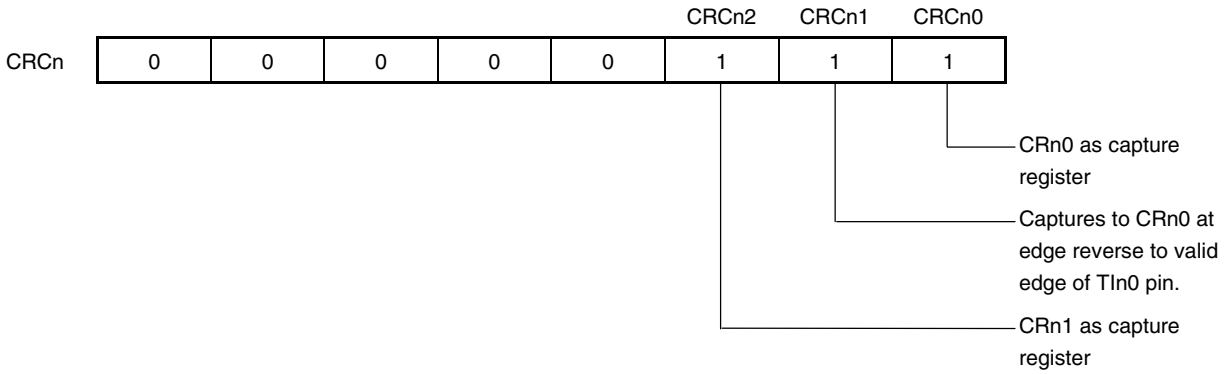
Remark n = 0, 1, 7 to 12

Figure 8-14. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers

(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



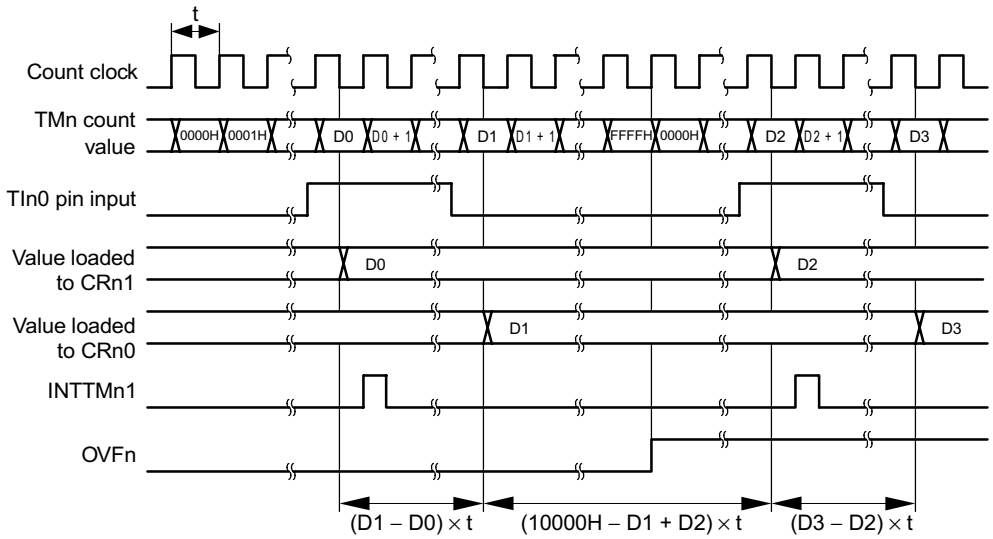
(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



Remarks 1. 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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-15. Timing of Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Remark n = 0, 1, 7 to 12

(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 8-17**).

The edge is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n0 (PRMn0). The rising or falling edge can be specified.

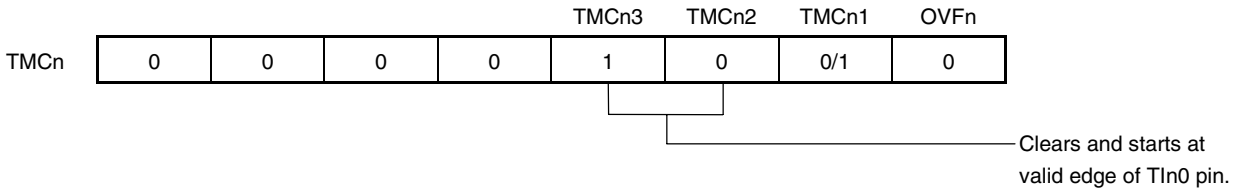
The valid edge is detected through sampling at a count clock cycle selected by prescaler mode registers n0 and n1 (PRMn0, PRMn1) and the capture operation is not performed until the valid level is detected two times, eliminating noise with a short pulse width.

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 a capture operation.

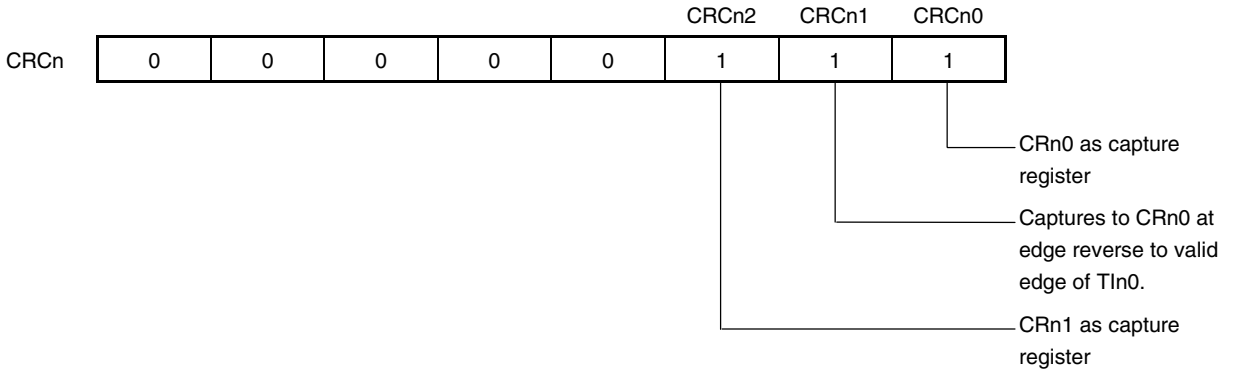
Remark n = 0, 1, 7 to 12

Figure 8-16. Control Register Settings for Pulse Width Measurement by Restarting

(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



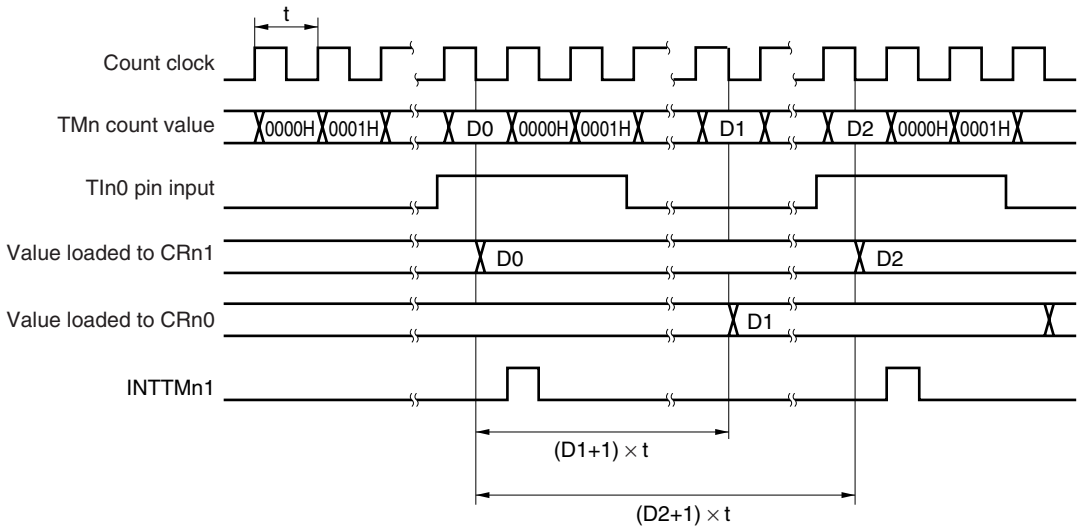
(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



Remarks 1. 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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-17. Timing of Pulse Width Measurement by Restarting (with Rising Edge Specified)



Remark n = 0, 1, 7 to 12

8.2.4 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) is input, TMn is incremented.

When the count value of TMn matches the value of 16-bit capture/compare register n0 (CRn0), TMn is cleared to 0, and an interrupt request signal (INTTMn0) is generated.

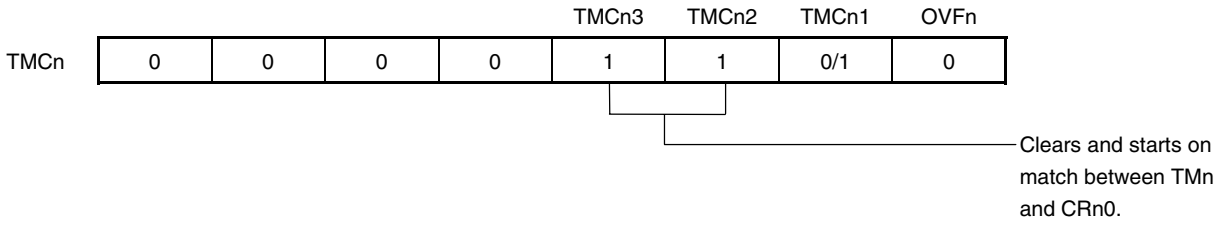
The edge is specified by bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n0 (PRMn0). The rising, falling, or both rising and falling edges can be specified.

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, eliminating noise with a short pulse width.

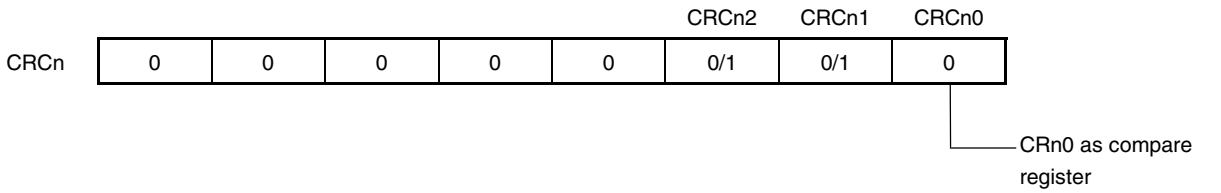
Remark n = 0, 1, 7 to 12

Figure 8-18. Control Register Settings in External Event Counter Mode

(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



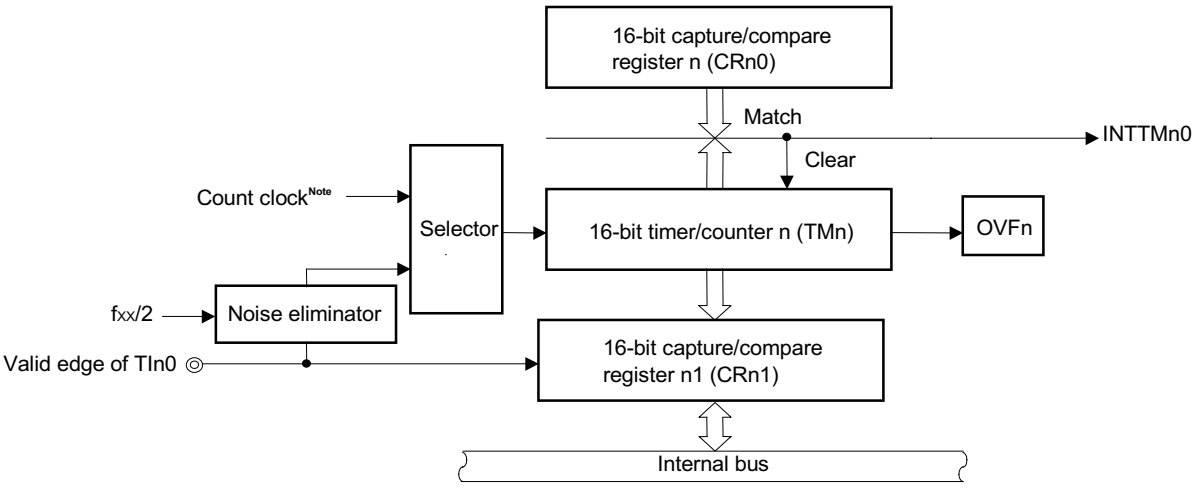
(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



Remarks 1. 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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-19. Configuration of External Event Counter

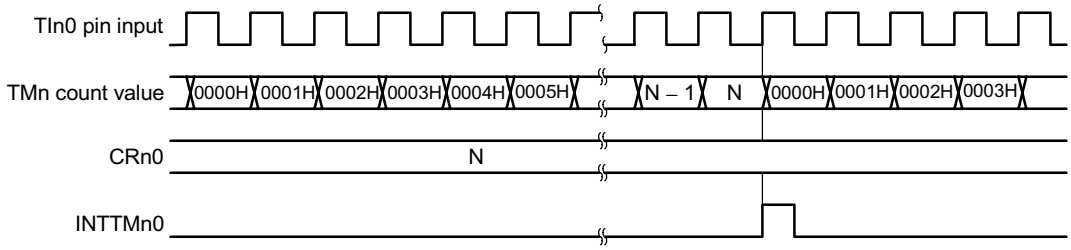


Note The count clock is set by the PRMn0 and PRMn1 registers.

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

2. n = 0, 1, 7 to 12

Figure 8-20. 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 to 12

8.2.5 Operation as square-wave output

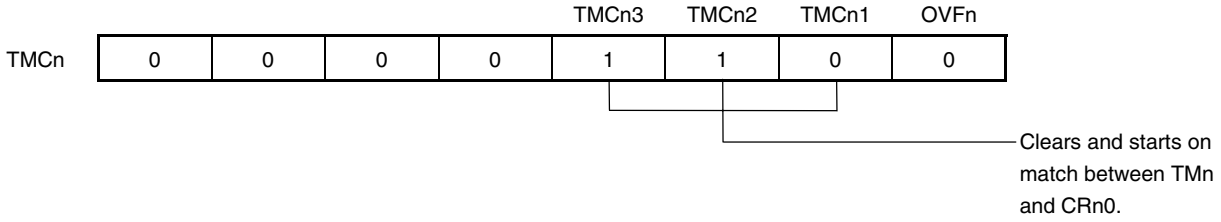
TMn can be used to output a square-wave with any frequency at an interval specified by the count value preset to 16-bit capture/compare register n0 (CRn0).

By setting bits 0 (TOEn) and 1 (TOCn1) of 16-bit timer output control register n (TOCn) to 1, the output status of the TOn pin is inverted at an interval specified by the count value preset to CRn1. In this way, a square wave with any frequency can be output.

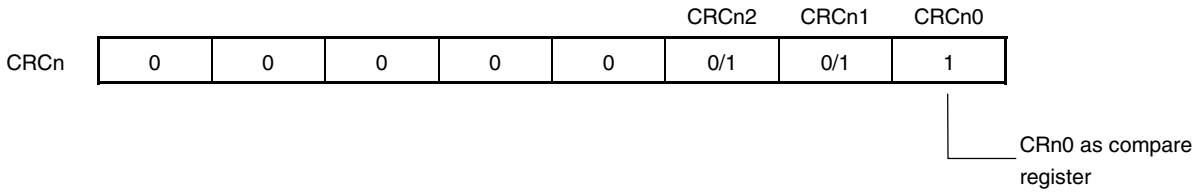
Remark n = 0, 1, 7 to 12

Figure 8-21. Control Register Settings in Square-Wave Output Mode

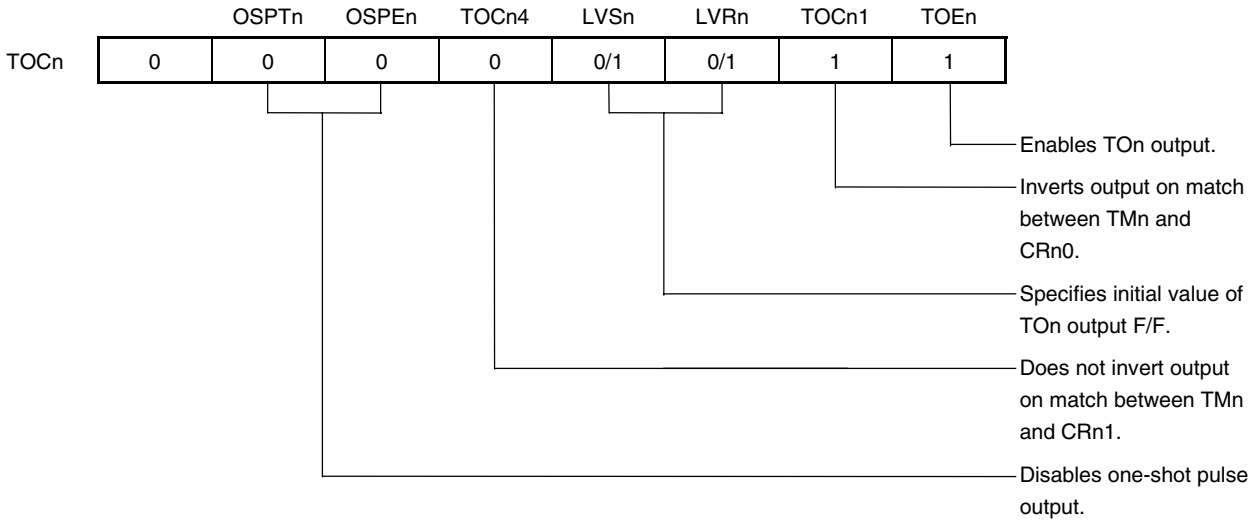
(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



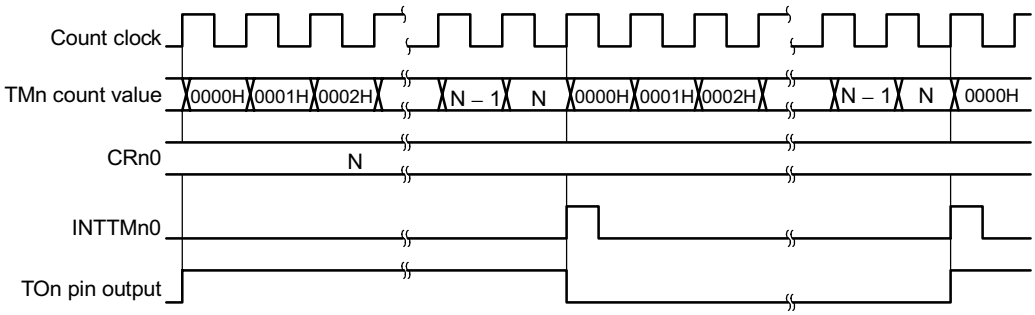
(c) 16-bit timer output control registers 0, 1, 7 to 12 (TOC0, TOC1, TOC7 to TOC12)



Remarks 1. 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the square-wave output function. For details, refer to **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-22. Timing of Square-Wave Output Operation



Remark n = 0, 1, 7 to 12

8.2.6 Operation as one-shot pulse output

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 via 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 8-23, and by setting bit 6 (OSPTn) of TOCn by software.

By setting OSPTn to 1, the 16-bit timer/event counter is cleared and started, and its output is asserted active at the count value (N) preset to 16-bit capture/compare register n1 (CRn1). After that, the output is deasserted inactive at the count value (M) preset 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.

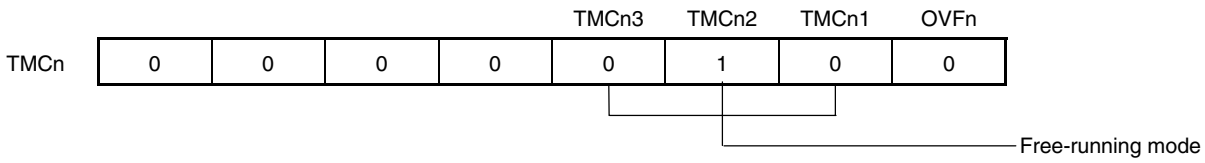
Note This is an example when N < M. When N > M, the output becomes active at the CRn0 value and inactive at the CRn1 value.

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 ends.

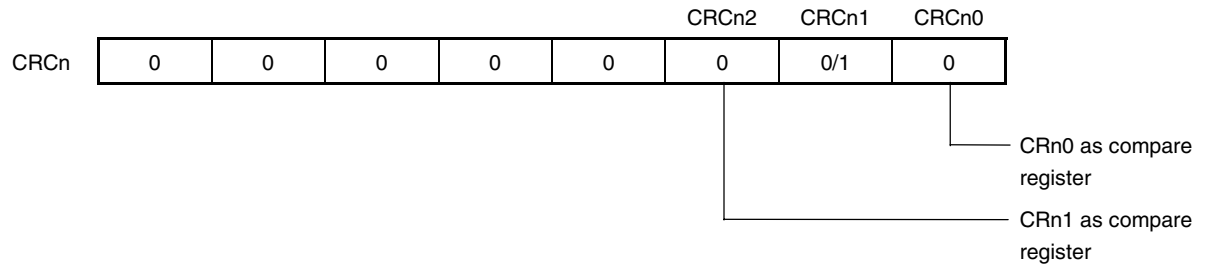
Remark n = 0, 1, 7 to 12

Figure 8-23. Control Register Settings for One-Shot Pulse Output via Software Trigger

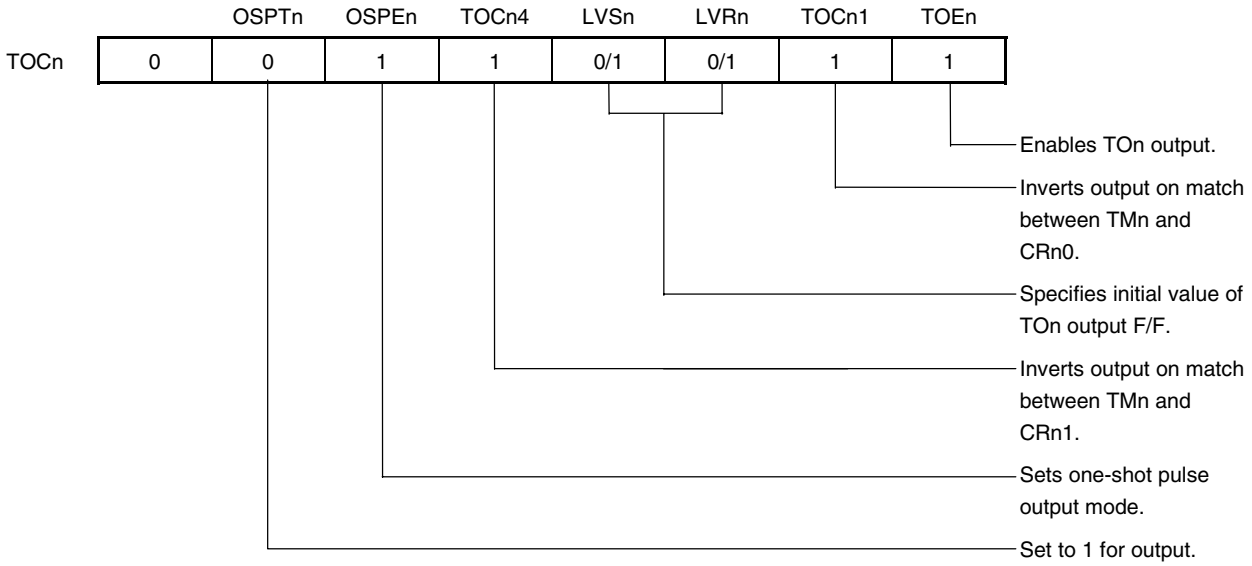
(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



(c) 16-bit timer output control registers 0, 1, 7 to 12 (TOC0, TOC1, TOC7 to TOC12)

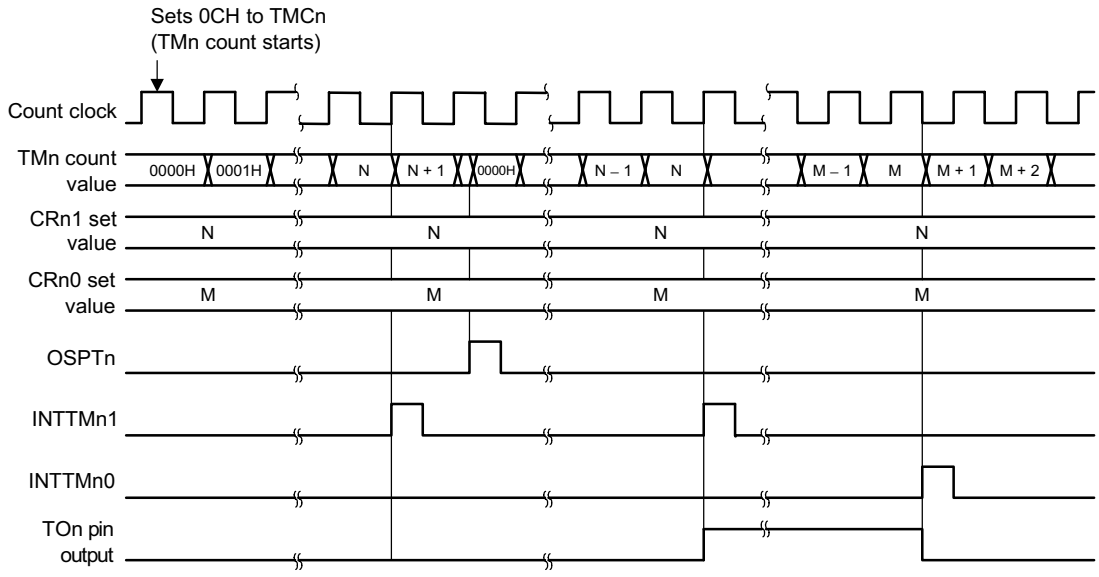


Caution Do not set CRn0 and CRn1 to 0000H.

Remarks 1. 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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-24. Timing of One-Shot Pulse Output Operation via Software Trigger



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

Remark n = 0, 1, 7 to 12
N < M

(2) One-shot pulse output via 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 8-25, 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 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 active at the count value (N) preset to 16-bit capture/compare register n1 (CRn1).

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

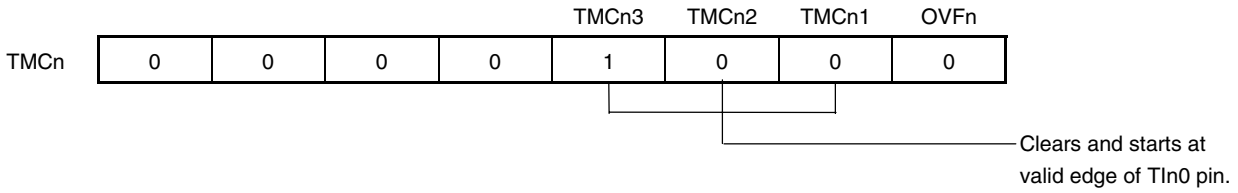
Note This is an example when N < M. When N > M, the output becomes active at the CRn0 value and inactive at the CRn1 value.

★ **Caution If the external trigger occurs while a one-shot pulse is being output, the 16-bit timer/counter clears & starts and the one-shot pulse is output again.**

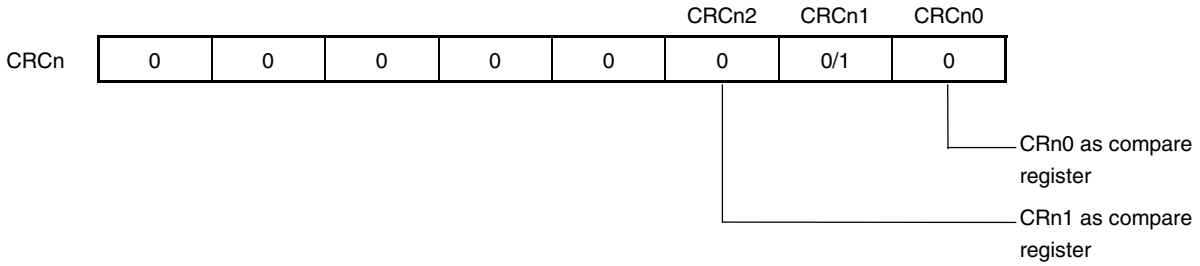
Remark n = 0, 1, 7 to 12

Figure 8-25. Control Register Settings for One-Shot Pulse Output via External Trigger

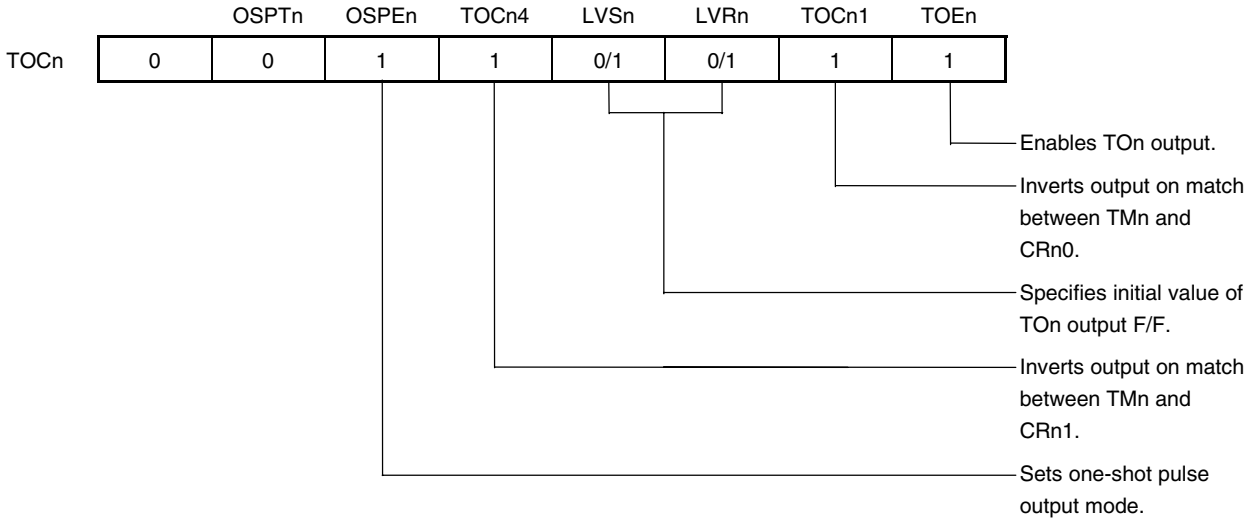
(a) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)



(b) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)



(c) 16-bit timer output control registers 0, 1, 7 to 12 (TOC0, TOC1, TOC7 to TOC12)

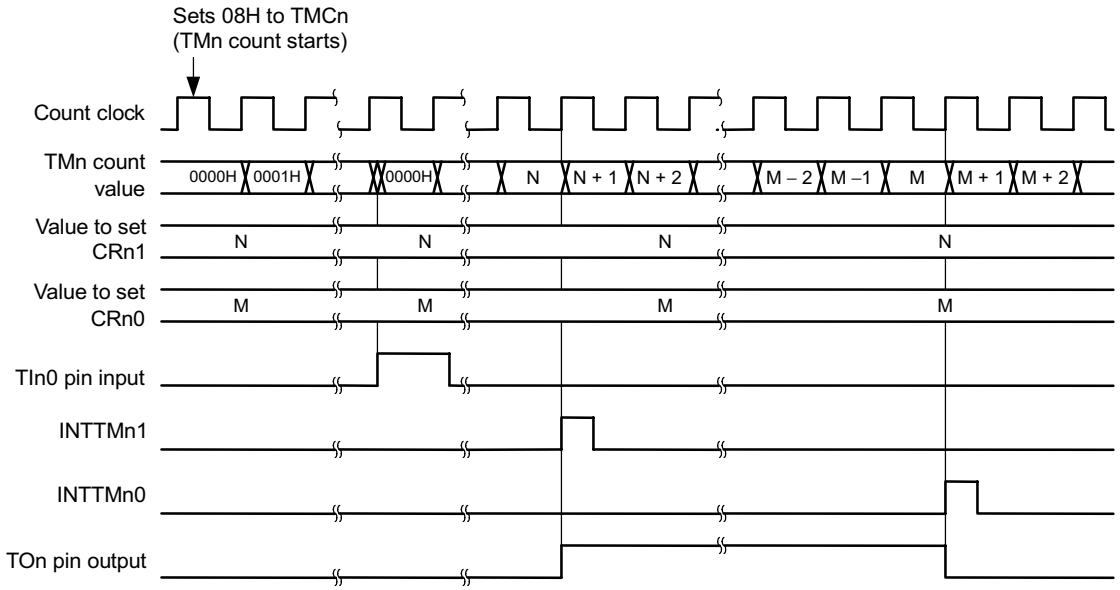


Caution Do not set CRn0 and CRn1 to 0000H.

Remarks 1. 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 **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)** and **(2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**.

2. n = 0, 1, 7 to 12

Figure 8-26. Timing of One-Shot Pulse Output Operation via External Trigger (with Rising Edge Specified)



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

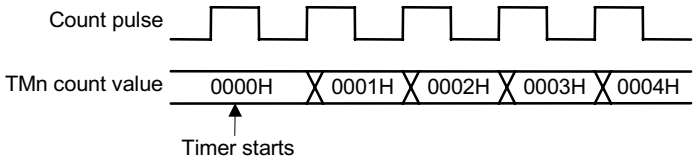
Remark n = 0, 1, 7 to 12
N > M

8.2.7 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 8-27. Start Timing of 16-Bit Timer Register n



Remark n = 0, 1, 7 to 12

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

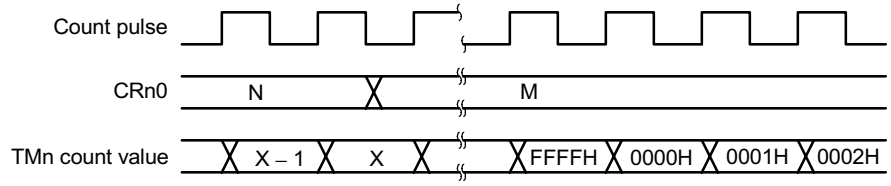
Set 16-bit capture/compare registers n0 and n1 (CRn0, CRn1) to a value other than 0000H (a 1-pulse count operation is disabled when these registers are used as event counters).

(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 restarted after the value of CRn0 has been changed.

Figure 8-28. Timing After Changing Compare Register During Timer Count Operation



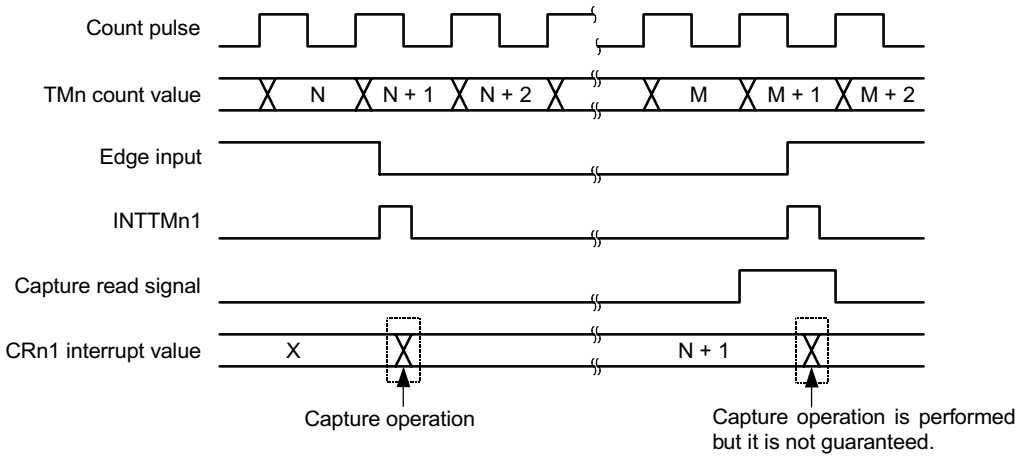
Remarks 1. N > X > M

2. n = 0, 1, 7 to 12

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

Figure 8-29. Data Hold Timing of Capture Register



Remark n = 0, 1, 7 to 12

(5) Setting valid edge

Before setting the valid edge of the TIn0 pin, stop the timer operation by resetting bits 2 and 3 (TMCn2 and TMCn3) of 16-bit timer mode control register n to 0, 0. Set the valid edge by using bits 4 and 5 (ESn00 and ESn01) of prescaler mode register n0 (PRMn0).

Remark n = 0, 1, 7 to 12

(6) Re-triggering one-shot pulse

(a) One-shot pulse output via 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 via external trigger

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

(c) On-shot pulse output function

When using a software trigger for one-shot pulse output of timers 0, 1, and 7 to 12, do not change the level of the TIn0 pin or its alternate-function pin.

The reason for this is that the timer is inadvertently cleared and started at the level of the TIn0 pin pr its alternate-function pin and pulses are output at an unintended timing because the external trigger is valid.

Remark n = 0, 1, 7 to 12

(7) Operation of OVFn flag

(a) OVFn flag set

The OVFn flag is set to 1 in the following case in addition to when TMn register overflows:

Selection of mode in which TM0 is cleared and started on a match between TMn and CRn0.

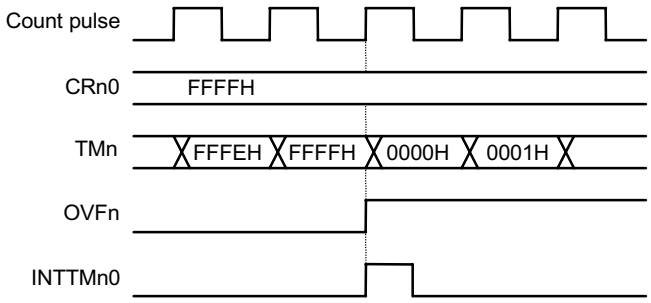
↓

CRn0 is set to FFFFH.

↓

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

Figure 8-30. Operation Timing of OVFn Flag



Remark n = 0, 1, 7 to 12

(b) Clear OVFn flag

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

Remark n = 0, 1, 7 to 12

(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.

Remark n = 0, 1, 7 to 12

(9) Timer operation

(a) CRn1 capture

Even if 16-bit timer register n (TMn) is read, a capture to 16-bit capture/compare register n1 (CRn1) is not performed.

(b) Acknowledgement of TIn0 and TIn1 pins

When the timer is stopped, input signals to the TIn0 and TIn1 pins are not acknowledged, regardless of the CPU operation.

(c) One-shot pulse output

The one-shot pulse output operates correctly only in free-running mode or in clear & start mode entered upon the valid edge of the TIn0 pin. The one-shot pulse cannot be output in the clear & start mode entered on a match of TMn and CRn0 because an overflow does not occur.

Remark n = 0, 1, 7 to 12

(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 both rising and falling edges are selected as the valid edge of TIn0, a capture operation is not performed.

(c) To capture the signals correctly from TIn0 and TIn1

The capture trigger needs a pulse longer than twice the count clock selected by prescaler mode registers n0 and n1 (PRMn0, PRMn1) in order to correctly capture the signals from TIn1 and TIn0.

(d) Interrupt request input

Although a capture operation is performed at the falling edge of the count clock, interrupt request inputs (INTTMn0, INTTMn1) are generated at the rising edge of the next count clock.

Remark n = 0, 1, 7 to 12

(11) Compare operation

(a) When rewriting CRn0 and CRn1 during timer operation

When rewriting 16-bit timer capture/compare registers n0 and n1 (CRn0, CRn1), if the value is close to or larger than the timer value, the match interrupt request generation or clear operation may not be performed correctly.

(b) When CRn0 and CRn1 are set to compare mode

When CRn0 and CRn1 are set to compare mode, they do not perform a capture operation even if a capture trigger is input.

Remark n = 0, 1, 7 to 12

(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.

Remark n = 0, 1, 7 to 12

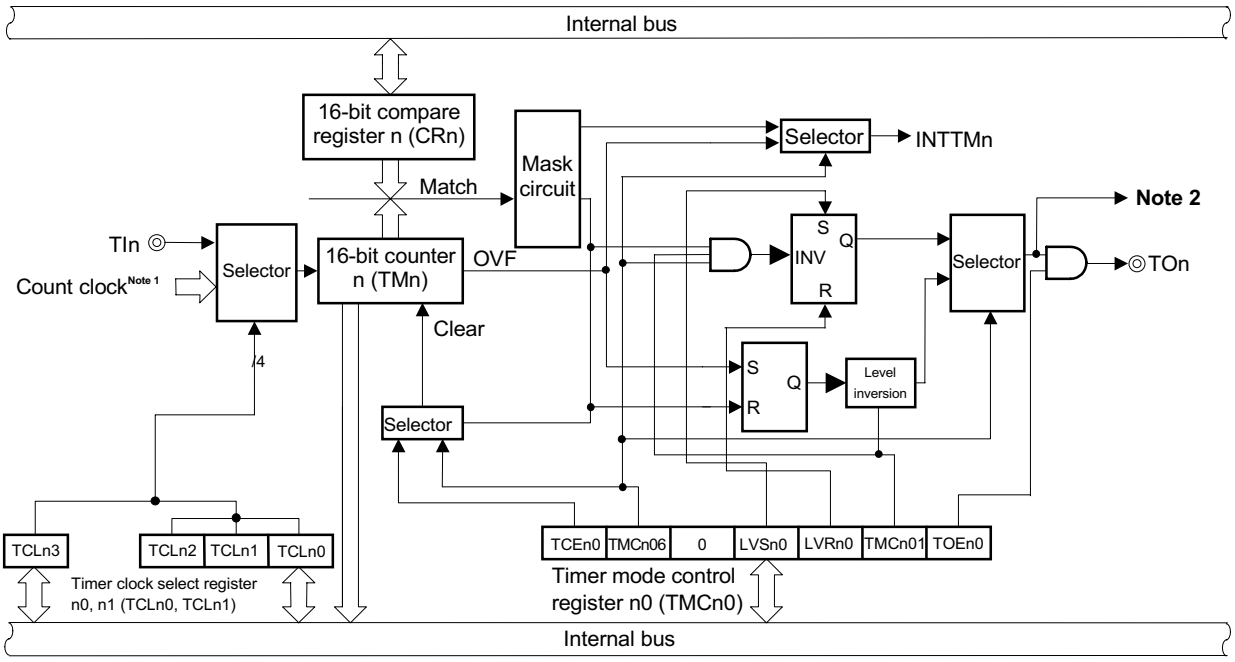
8.3 16-Bit Timer (TM5, TM6)

8.3.1 Functions

TM5 and TM6 have the following functions.

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

Figure 8-31. Block Diagram of TM5 and TM6



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

2. Serial interface clock.

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

2. n = 5, 6

8.3.2 Configuration

Timer n includes the following hardware.

Table 8-5. Configuration of Timers 5 and 6

Item	Configuration
Timer registers	16-bit counters 5, 6 (TM5, TM6)
Registers	16-bit compare registers 5, 6 (CR5, CR6)
Timer outputs	TO5, TO6
Control registers	Timer clock select registers 50, 51, 60, and 61 (TCL50, TCL51, TCL60, and TCL61) 8-bit timer mode control registers 50 and 60 (TMC50, TMC60)

(1) 16-bit counters 5, 6 (TM5, TM6)

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

The counter is incremented in synchronization with the rising edge of the count clock.

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 0000H.

(1) When $\overline{\text{RESET}}$ is input.
(2) When TCEn is cleared.
(3) When TMn and CRn match in the clear & start mode that is entered when TMn and CRn0 match.

Remark n = 5, 6

(2) 16-bit compare registers 5, 6 (CR5, CR6)

The value set in CRn is always compared to the count in 16-bit counter n (TMn). If the two values match, an interrupt request (INTTMn) is generated (except in the PWM mode).

Remark n = 5, 6

8.3.3 Timer n control registers

Timer n is controlled by the following registers.

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

(1) Timer clock select registers 50, 51, 60, 61 (TCL50, TCL51, TCL60, TCL61)

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 sets these registers to 00H.

After reset: 00H R/W Address: FFFFF33EH

	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	0	0	TCL503

After reset: 00H R/W Address: FFFFF334H

	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL503	TCL502	TCL501	TCL500	Count clock selection			
				Count clock	f_{XX}		
					20 MHz	18.87 MHz	16 MHz
0	0	0	0	TI5 falling edge	–	–	–
0	0	0	1	TI5 rising edge	–	–	–
0	0	1	0	$f_{XX}/2$	100 ns	105 ns	125 ns
0	0	1	1	$f_{XX}/4$	200 ns	212 ns	250 ns
0	1	0	0	$f_{XX}/8$	400 ns	424 ns	500 ns
0	1	0	1	$f_{XX}/16$	800 ns	848 ns	1 μs
0	1	1	0	$f_{XX}/64$	3.2 μs	3.4 μs	4 μs
0	1	1	1	f_{XT} (subclock)	30.5 μs	30.5 μs	30.5 μs
1	0	0	0	Setting prohibited	–	–	–
1	0	0	1	Setting prohibited	–	–	–
1	0	1	0	$f_{XX}/32$	1.6 μs	1.7 μs	2 μs
1	0	1	1	$f_{XX}/128$	6.4 μs	6.8 μs	8 μs
1	1	0	0	Setting prohibited	–	–	–
1	1	0	1	Setting prohibited	–	–	–
1	1	1	0	Setting prohibited	–	–	–
1	1	1	1	Setting prohibited	–	–	–

Cautions 1. To overwrite TCL50 and TCL51 with different data, temporarily stop the timer first.

2. Always set bits 3 to 7 of TCL50 and bits 1 to 7 of TCL51 to 0.

After reset: 00H R/W Address: FFFFF28EH

	7	6	5	4	3	2	1	0
TCL61	0	0	0	0	0	0	0	TCL603

After reset: 00H R/W Address: FFFFF284H

	7	6	5	4	3	2	1	0
TCL60	0	0	0	0	0	TCL602	TCL601	TCL600

TCL603	TCL602	TCL601	TCL600	Count clock selection			
				Count clock	fxx		
					20 MHz	18.87 MHz	16 MHz
0	0	0	0	TI6 falling edge	–	–	–
0	0	0	1	TI6 rising edge	–	–	–
0	0	1	0	fxx/2	100 ns	105 ns	125 ns
0	0	1	1	fxx/4	200 ns	212 ns	250 ns
0	1	0	0	fxx/8	400 ns	424 ns	500 ns
0	1	0	1	fxx/16	800 ns	848 ns	1 μs
0	1	1	0	fxx/64	3.2 μs	3.4 μs	4 μs
0	1	1	1	fxx/256	12.8 μs	13.6 μs	16 μs
1	0	0	0	Setting prohibited	–	–	–
1	0	0	1	Setting prohibited	–	–	–
1	0	1	0	fxx/32	1.6 μs	1.7 μs	2 μs
1	0	1	1	fxx/128	6.4 μs	6.8 μs	8 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	–	–	–

Cautions 1. To overwrite TCL60 and TCL61 with different data, temporarily stop the timer first.

2. Always set bits 3 to 7 of TCL60 and bits 1 to 7 of TCL61 to 0.

(2) 16-bit timer mode control registers 50, 60 (TMC50, TMC60)

The TMCn0 register makes the following five settings.

(1) Controls the counting by 16-bit counter n (TMn)
(2) Selects the operating mode of 16-bit counter n (TMn)
(3) Sets the state of the timer output flip-flop
(4) Controls the timer flip-flop or selects the active level in the PWM (free-running) mode
(5) Controls timer output

TMCn0 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 the hardware is initialized to 04H, 00H is readout when reading).

Remark n = 5, 6

After reset: 04H R/W Address: TMC50 FFFFF336H TMC60 FFFFF286H

	<7>	6	5	4	<3>	<2>	1	<0>
TMCn0	TCEn0	TMCn06	0	0	LVSn0	LVRn0	TMCn01	TOEn0

(n = 5, 6)

TCEn0	TMn count operation control
0	Counting is disabled after the counter is cleared to 0 (prescaler disabled)
1	Start count operation

TMCn06	TMn operating mode selection
0	Clear & start mode entered when TMn and CRn match
1	PWM (free-running) mode

LVSn0	LVRn0	Setting state of timer output flip-flop
0	0	Not change
0	1	Reset timer output flip-flop (0)
1	0	Set timer output flip-flop (1)
1	1	Setting prohibited

TMCn01	Other than PWM (free-running) mode (TMCn06 = 0)	PWM (free-running) mode (TMCn06 = 1)
	Controls timer F/F	Selects active level
0	Disable inversion operation	Active high
1	Enable inversion operation	Active low

TOEn0	Timer output control
0	Disable output (port mode)
1	Enable output

Cautions 1. When using the timer output pin (TOn), set the port value to 0 (port mode output). An ORed value (logical OR) of the timer output values is output.

2. Since TOn and TIn are the same alternate-function pin, only one function can be used.

Remarks 1. In the PWM mode, the PWM output is set to the inactive level by TCEn0 = 0.

2. If LVSn0 and LVRn0 are read after setting data, 0 is read.

8.4 16-Bit Timer (TM5, TM6) Operation

8.4.1 Operation as an interval timer

TMn operates as an interval timer that repeatedly generates interrupts at the time interval specified by the count value preset to 16-bit compare register n (CRn).

When the count value of 16-bit counter n (TMn) matches the set value of CRn, the value of TMn is cleared to 0, and the timer continues counting. At the same time, an interrupt request signal (INTTMn) is generated.

The TMn count clock can be selected by bits 0 to 2 (TCLn0 to TCLn2) of timer clock select register n0 (TCLn0) and by bit 0 (TCLn3) of timer clock select register n1 (TCLn1) (n = 5, 6).

Setting method

(1) Set each register.
- TCLn0, TCLn1: Selects the count clock.
- CRn: Compare value
- TMCn0: Selects the clear and start mode entered when TMn and CRn match. (TMCn0 = 0000xxx0B, × = don't care)

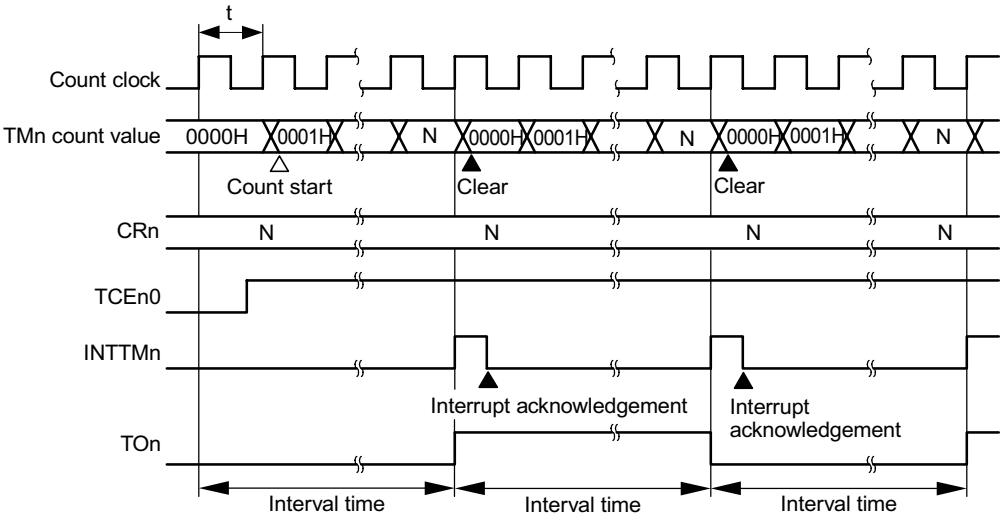
(2) When TCEn0 = 1 is set, counting starts.

(3) When the values of TMn and CRn match, INTTMn is generated (TMn is cleared to 0000H).

(4) INTTMn is then repeatedly generated at the same interval. When counting stops, set TCEn0 = 0.

Figure 8-32. Timing of Interval Timer Operation (1/2)

Basic operation

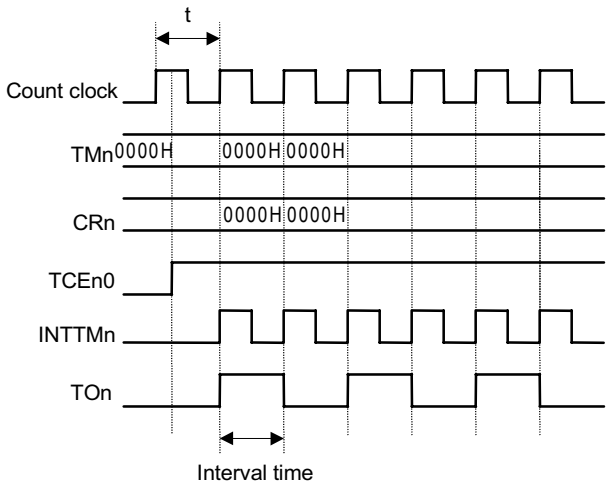


Remarks 1. Interval time = (N + 1) × t; N = 0000H to FFFFH

2. n = 5, 6

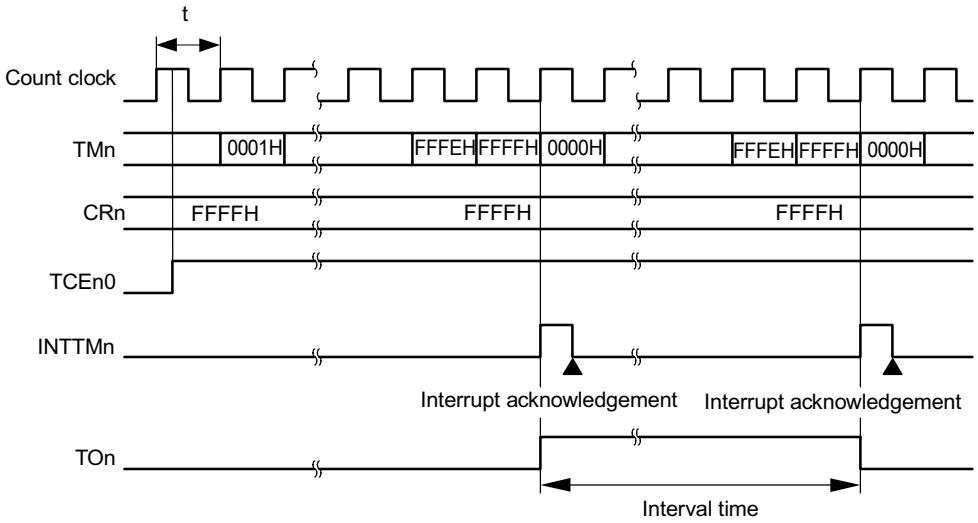
Figure 8-32. Timing of Interval Timer Operation (2/2)

When CRn = 0000H



Remark n = 5, 6

When CRn = FFFFH



Remark n = 5, 6

8.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses that are input to TIn.

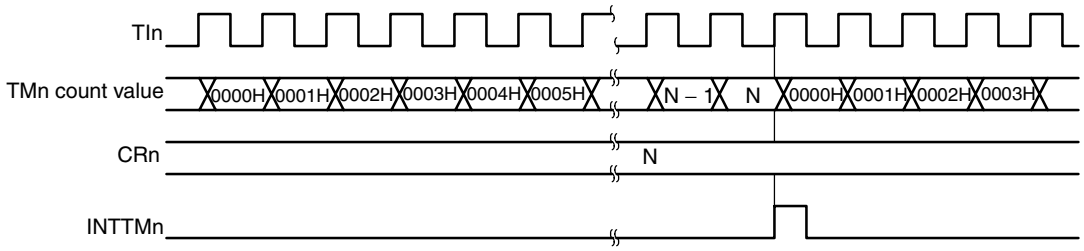
Each time the valid edge specified by timer clock select registers n0 and n1 (TCLn0, TCLn1) is input, TMn is incremented. The edge setting can be selected to be either a rising or falling edge.

If the total value of TMn and the value of 16-bit compare register n (CRn) match, TMn is cleared to 0 and an interrupt request signal (INTTMn) is generated.

INTTMn is generated each time the TMn value matches the CRn value.

Remark n = 5, 6

Figure 8-33. Timing of External Event Counter Operation (with Rising Edge Specified)



Remark n = 5, 6

8.4.3 Operation as square-wave output

A square-wave with any frequency is output at the interval preset to 16-bit compare register n (CRn).

By setting bit 0 (TOEn0) of 16-bit timer mode control register n0 (TMCn0) to 1, the output status of TOn is inverted at an interval specified by the count value preset to CRn. In this way, a square wave of any frequency (duty ratio = 50%) can be output.

Setting method

(1) Set the registers.
- Set the port latch and port mode register to 0
- TCLn0, TCLn1: Selects the count clock
- CRn: Compare value
- TMCn0: Clear and start mode entered when TMn and CRn match

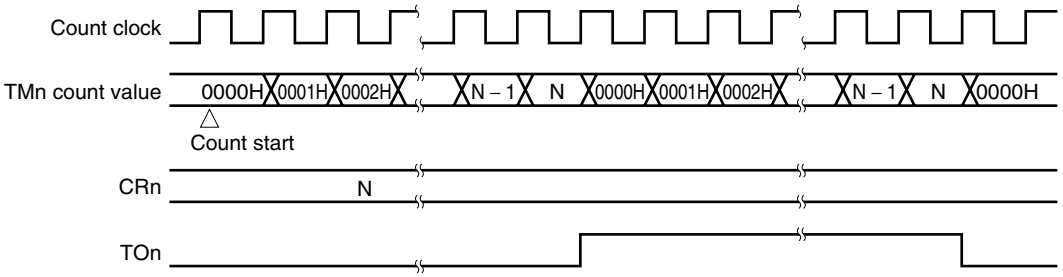
LVSn0	LVRn0	Setting state of timer output flip-flop
1	0	High level output
0	1	Low level output

Inversion of timer output flip-flop enabled
Timer output enabled → TOEn0 = 1

(2) When TCEn0 = 1 is set, the counter starts operating.
(3) If the values of TMn and CRn match, the timer output flip-flop inverts. Also, INTTMn is generated and TMn is cleared to 0000H.
(4) The timer output flip-flop is then inverted at the same interval and a square wave is output from TOn.

Remark n = 5, 6

Figure 8-34. Square-Wave Output Operation Timing



Note The initial value of TOn output can be set with bits 3 and 2 (LVSn0, LVRn0) of the TMCn0 register.

Remark n = 5, 6

8.4.4 Operation as 16-bit PWM output

By setting bit 6 (TMCn6) of 16-bit timer mode control register n0 (TMCn0) to 1, the timer operates as a PWM output.

Pulses with the duty ratio determined by the value set to 16-bit compare register n (CRn) are output from TOn.

Set the width of the active level of the PWM pulse to CRn. The active level can be selected by bit 1 (TMCn01) of TMCn0.

The count clock can be selected by bits 0 to 2 (TCLn0 to TCLn2) of timer clock select register n0 (TCLn0) and by bit 0 (TCLn3) of timer clock select register n1 (TCLn1).

The PWM output can be enabled and disabled by bit 0 (TOEn0) of TMCn0.

Caution CRn can be rewritten only once in one period while in the PWM mode.

Remark n = 5, 6

(1) Basic operation of PWM output

Setting method

(1) Set the port latch and port mode register n to 0.
(2) Set the active level width to 16-bit compare register n (CRn).
(3) Select the count clock using timer clock select register n0, n1 (TCLn0, TCLn1).
(4) Set the active level to bit 1 (TMCn01) of TMCn0.
(5) If bit 7 (TCEn0) of TMCn0 is set to 1, counting starts. To stop counting, set TCEn0 to 0.

PWM output operation

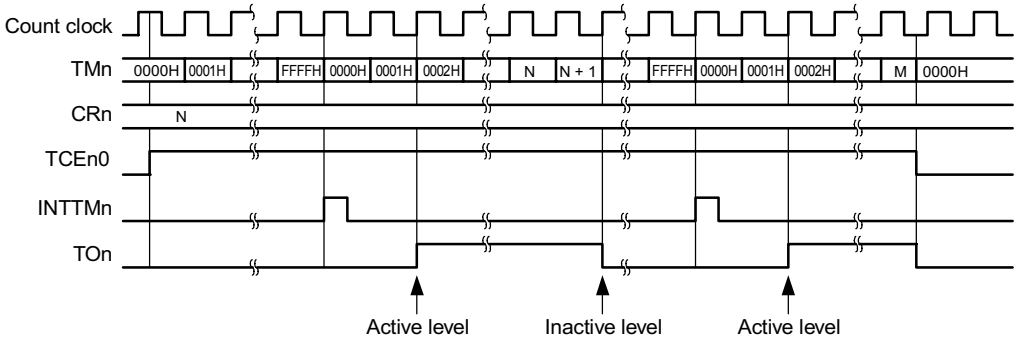
(1) When counting starts, PWM output (output from TOn) outputs the inactive level until an overflow occurs.
(2) When an overflow occurs, the active level specified in step (1) in the setting method is output. The active level is output until CRn and the count value of 16-bit counter n (TMn) match.
(3) PWM output after the CRn and count values match is at the inactive level until an overflow occurs again.
(4) Steps (2) and (3) repeat until counting stops.
(5) If counting is stopped by TCEn0 = 0, PWM output goes to the inactive level.

Remark n = 5, 6

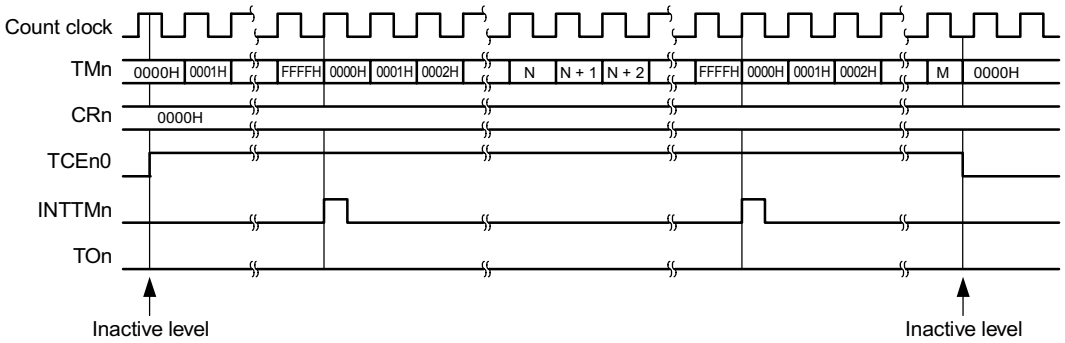
(a) Basic operation of PWM output

Figure 8-35. Timing of PWM Output

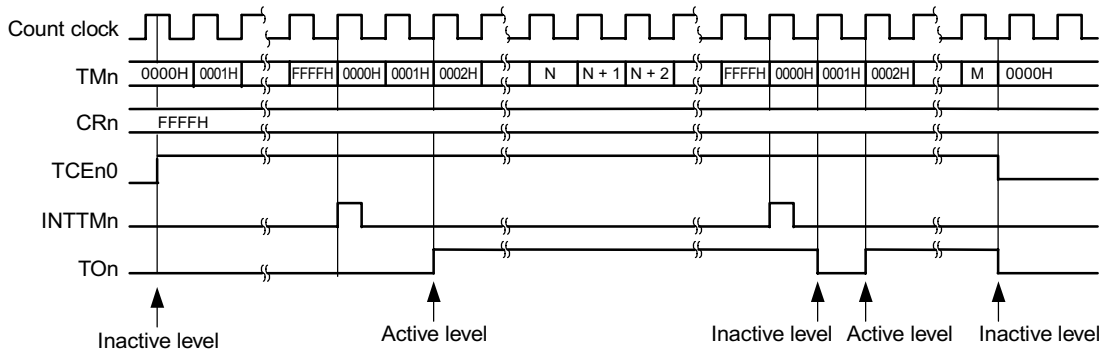
Basic operation (active level = H)



When CRn = 0



When CRn = FFFFH



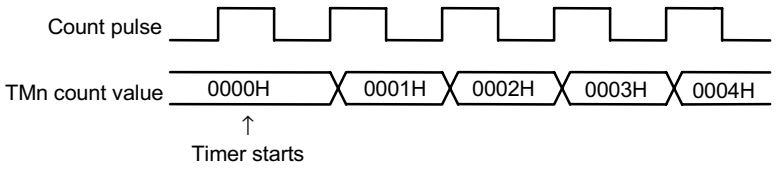
Remark n = 5, 6

8.4.5 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 counter n (TMn) is started asynchronously to the count pulse.

Figure 8-36. Start Timing of Timer n



Remark n = 5, 6

(2) TMn readout during timer operation

Since reading out TMn during operation occurs while the selected clock is temporarily stopped, be sure to select a high- or low-level waveform that is longer than the selected clock (n = 5, 6).

CHAPTER 9 WATCH TIMER FUNCTION

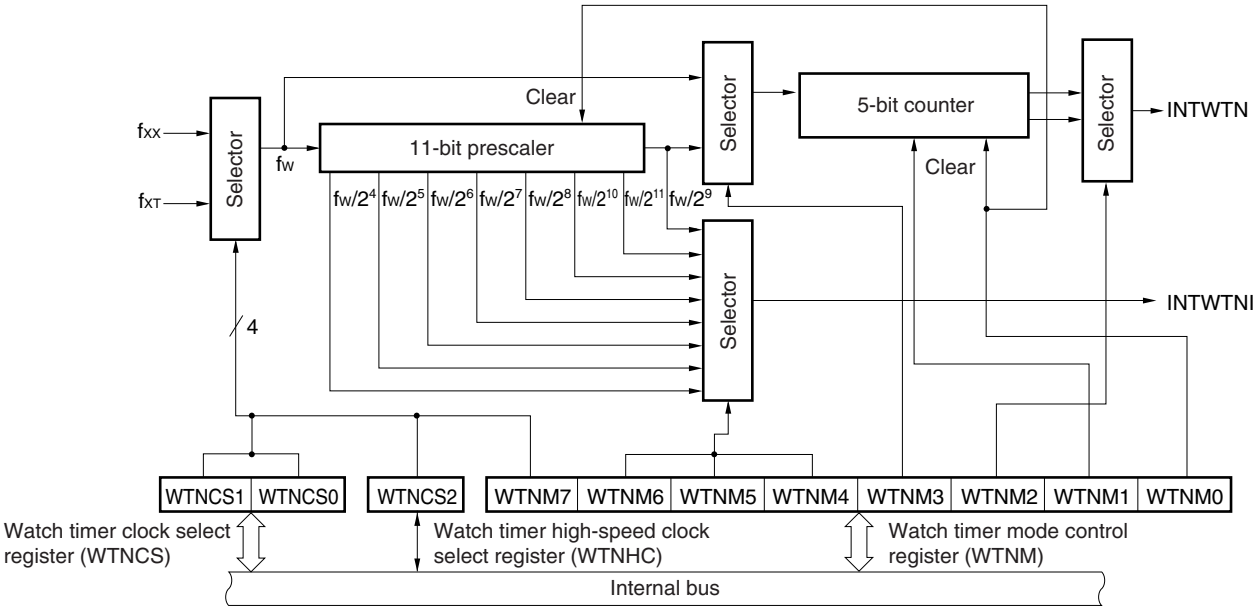
9.1 Function

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 9-1. Block Diagram of Watch Timer



Remark fXX: Main clock frequency
fXT: Subclock frequency
fw: Watch timer clock frequency

(1) Watch timer

The watch timer generates an interrupt request (INTWTN) at time intervals of 0.5 or 0.25 second 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 9-1. Interval Time of Interval Timer

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

Remark fw: Watch timer clock frequency

9.2 Configuration

The watch timer includes the following hardware.

Table 9-2. 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 select register (WTNHC) Watch timer clock select register (WTNCS)

9.3 Watch Timer Control Register

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

(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 interrupt time of the watch timer.

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	Selects 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	Selects interrupt time of watch timer
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	Controls operation of 5-bit counter
0	Clears after operation stops
1	Starts

WTNM0	Enables operation of watch timer
0	Stops operation (clears both prescaler and 5-bit counter)
1	Enables operation

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 **9.3 (3) Watch timer clock select register (WTNCS)**.

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

This register selects the count clock of the watch timer.

The count clock is determined using WTNM7 bit of WTNM register in combination with WTNCS1 and WTNCS0 bits of the watch timer clock select 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

Remark For the settings of WTNCS2, refer to **9.3 (3) Watch timer clock select register (WTNCS)**.

(3) Watch timer clock select 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, interrupt time for watch timer, 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	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	–

Remark WTNM7 is bit 7 of the WTNM register.
WTNCS2 is bit 0 of the WTNHC register.

9.4 Operation

9.4.1 Operation as watch timer

The watch timer operates with time intervals of 0.5 second 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.

Setting the WTNM1 bit to 0 can clear the 5-bit counter of the watch timer. An error of up to 15.6 ms may occur at this time.

Setting the WTNM0 bit to 0 can clear the interval timer. However, an error up to 0.5 s may occur after a watch timer overflow (INTWTN) because the 5-bit counter is also cleared.

9.4.2 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 preset count value.

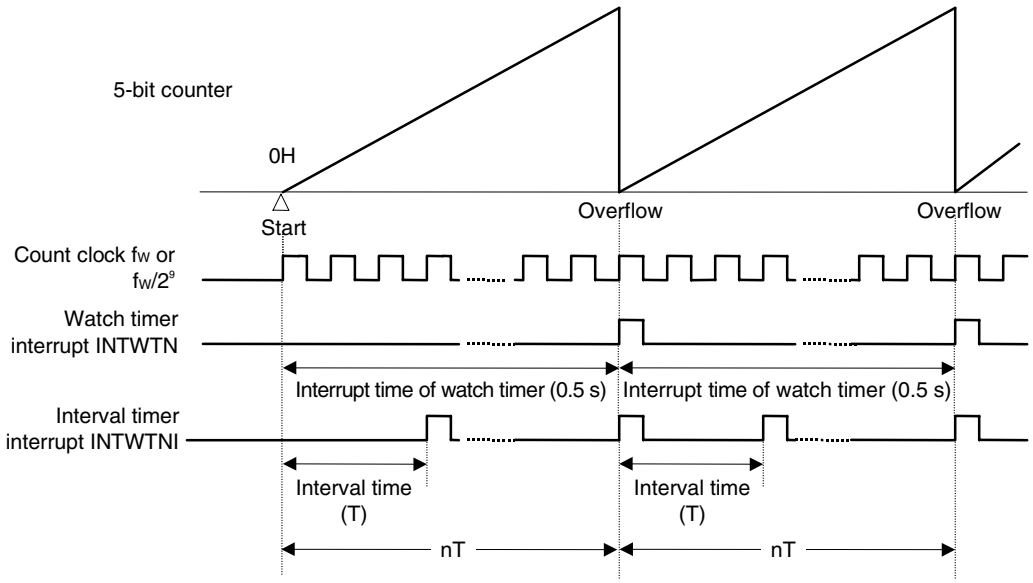
The interval time can be selected by bits 4 to 6 (WTNM4 to WTNM6) of the watch timer mode control register (WTNM).

Table 9-3. Interval Time of Interval Timer

WTNM6	WTNM5	WTNM4	Interval Time	fw = 32.768 kHz
0	0	0	$2^4 \times 1/fw$	488 μs
0	0	1	$2^5 \times 1/fw$	977 μs
0	1	0	$2^6 \times 1/fw$	1.95 ms
0	1	1	$2^7 \times 1/fw$	3.91 ms
1	0	0	$2^8 \times 1/fw$	7.81 ms
1	0	1	$2^9 \times 1/fw$	15.6 ms
1	1	0	$2^{10} \times 1/fw$	31.2 ms
1	1	1	$2^{11} \times 1/fw$	62.4 ms

Remark fw: Watch timer clock frequency

Figure 9-2. Operation Timing of Watch Timer/Interval Timer



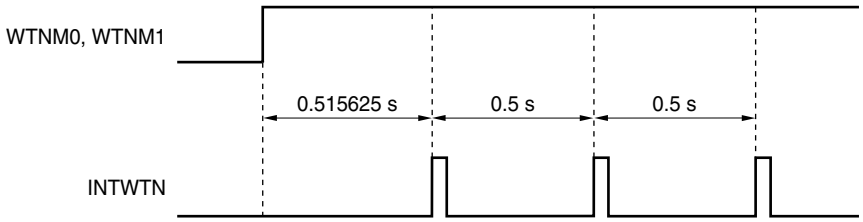
Remark fw: Watch timer clock frequency
( ): fw = 32.768 kHz
n: Number of interval timer operations

9.4.3 Cautions

It takes some time to generate the first watch timer interrupt request (INTWTN) after operation is enabled (WTNM1 and WTNM0 bits of WTNM register = 1).

Figure 9-3. 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.



CHAPTER 10 WATCHDOG TIMER FUNCTION

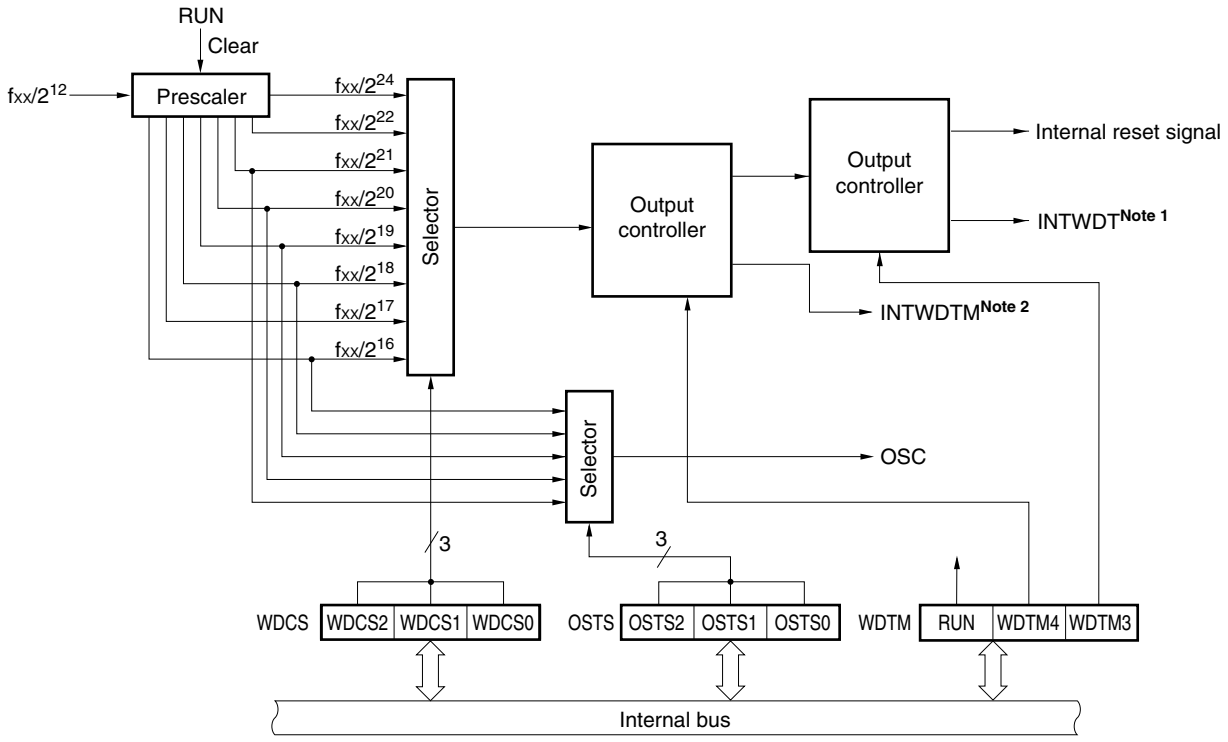
10.1 Functions

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 10-1. 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 a loop is detected, a non-maskable interrupt can be generated.

Table 10-1. Loop Detection Time of Watchdog Timer

Clock	Loop Detection Time		
	f_{XX} = 20 MHz	f_{XX} = 18.87 MHz	f_{XX} = 16 MHz
$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
$2^{17}/f_{XX}$	6.6 ms	6.9 ms	8.2 ms
$2^{18}/f_{XX}$	13.1 ms	13.9 ms	16.4 ms
$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
$2^{22}/f_{XX}$	209.7 ms	222.3 ms	262.1 ms
$2^{24}/f_{XX}$	838.9 ms	889.1 ms	1.05 s

(2) Interval timer mode

Interrupts are generated at a preset time interval.

Table 10-2. Interval Time of Interval Timer

Clock	Interval Time		
	f_{XX} = 20 MHz	f_{XX} = 18.87 MHz	f_{XX} = 16 MHz
$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
$2^{17}/f_{XX}$	6.6 ms	6.9 ms	8.2 ms
$2^{18}/f_{XX}$	13.1 ms	13.9 ms	16.4 ms
$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
$2^{22}/f_{XX}$	209.7 ms	222.3 ms	262.1 ms
$2^{24}/f_{XX}$	838.9 ms	889.1 ms	1.05 s

10.2 Configuration

The watchdog timer includes the following hardware.

Table 10-3. Watchdog Timer Configuration

Item	Configuration
Control registers	Oscillation stabilization time select register (OSTS) Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM)

10.3 Watchdog Timer Control Registers

The watchdog timer is controlled by the following registers.

- Oscillation stabilization time select register (OSTS)
- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Oscillation stabilization time select register (OSTS)

This register selects the oscillation stabilization time after a reset is applied or the 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 01H.

After reset: 01H 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	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
0	0	1	$2^{18}/f_{XX}$ (after reset)	13.1 ms	13.9 ms	16.4 ms
0	1	0	$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
0	1	1	$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
1	0	0	$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
Other than above			Setting prohibited			

(2) Watchdog timer clock select register (WDCS)

This register selects the overflow times of the watchdog timer and the interval timer.

WDCS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets WDCS to 00H.

After reset: 00H R/W Address: FFFFF382H

	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Watchdog timer/interval timer overflow time			
			Clock	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
0	0	1	$2^{17}/f_{XX}$	6.6 ms	6.9 ms	8.2 ms
0	1	0	$2^{18}/f_{XX}$	13.1 ms	13.9 ms	16.4 ms
0	1	1	$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
1	0	0	$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
1	0	1	$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
1	1	0	$2^{22}/f_{XX}$	209.7 ms	222.3 ms	262.1 ms
1	1	1	$2^{24}/f_{XX}$	838.9 ms	889.1 ms	1.05 s

★ **Caution Always set bits 7 to 3 to 0.**

(3) Watchdog timer mode register (WDTM)

This register sets the operating mode of the watchdog timer, enables and disables counting, and generates internal reset signals.

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

$\overline{\text{RESET}}$ input sets WDTM to 00H.

After reset: 00H R/W Address: FFFFF384H

	<7>	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	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.)

WDTM3	Internal reset signal generation selection[Note 2]
0	Internal reset signal not generated when overflow
1	Internal reset signal generated when overflow

Notes 1. Once RUN is set (1), the register cannot be cleared (0) by software. Therefore, when the count starts, the count cannot be stopped except by $\overline{\text{RESET}}$ input.

2. Once WDTM3 and WDTM4 are set (1), the registers 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 $2^{12}/f_{XX}$ seconds shorter than the set time.

10.4 Operation

10.4.1 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 inadvertent program looping.

Setting bit 7 (RUN) of WDTM to 1 starts the count. After counting starts, if RUN is set to 1 again within the set time interval for loop detection, the watchdog timer is cleared and counting starts again.

If RUN is not set to 1 and the loop detection time has elapsed, a non-maskable interrupt (INTWDT) is generated (no reset functions).

The watchdog timer stops running in the IDLE and STOP modes. Consequently, set RUN to 1 and clear the watchdog timer before entering the IDLE or STOP mode. Do not set the watchdog timer when using the HALT mode since the watchdog timer continues to operate in the HALT mode.

Cautions 1. The actual loop detection time may be up to $2^{12}/f_{XX}$ seconds shorter than the set time.

2. When the subclock is selected for the CPU clock, the watchdog timer stops counting (pauses).

Table 10-4. Loop Detection Time of Watchdog Timer

Clock	Loop Detection Time		
	f_{XX} = 20 MHz	f_{XX} = 18.87 MHz	f_{XX} = 16 MHz
$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
$2^{17}/f_{XX}$	6.6 ms	6.9 ms	8.2 ms
$2^{18}/f_{XX}$	13.1 ms	13.9 ms	16.4 ms
$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
$2^{22}/f_{XX}$	209.7 ms	222.3 ms	262.1 ms
$2^{24}/f_{XX}$	838.9 ms	889.1 ms	1.05 s

10.4.2 Operation as interval timer

Set bit 4 (WDTM4) of the watchdog timer mode register (WDTM) to 0 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 flags (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 and STOP modes. Therefore, before entering the IDLE/STOP mode set the RUN bit of the WDTM register to 1 to clear the interval timer, and then set to the IDLE or 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 $2^{12}/f_{XX}$ seconds shorter than the set time.

3. When the subclock is selected for the CPU clock, the watchdog timer stops counting (pauses).

Table 10-5. Interval Time of Interval Timer

Clock	Interval Time		
	f_{XX} = 20 MHz	f_{XX} = 18.87 MHz	f_{XX} = 16 MHz
$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
$2^{17}/f_{XX}$	6.6 ms	6.9 ms	8.2 ms
$2^{18}/f_{XX}$	13.1 ms	13.9 ms	16.4 ms
$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
$2^{22}/f_{XX}$	209.7 ms	222.3 ms	262.1 ms
$2^{24}/f_{XX}$	838.9 ms	889.1 ms	1.05 s

10.5 Standby Function Control Register

(1) Oscillation stabilization time select register (OSTS)

The wait time from when the stop mode is cancelled until the oscillation stabilizes is controlled by the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

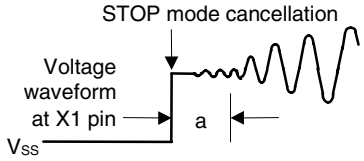
$\overline{\text{RESET}}$ input sets OSTS to 01H.

After reset: 01H 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	f_{XX}		
				20 MHz	18.87 MHz	16 MHz
0	0	0	$2^{16}/f_{XX}$	3.3 ms	3.5 ms	4.1 ms
0	0	1	$2^{18}/f_{XX}$ (after reset)	13.1 ms	13.9 ms	16.4 ms
0	1	0	$2^{19}/f_{XX}$	26.2 ms	27.8 ms	32.8 ms
0	1	1	$2^{20}/f_{XX}$	52.4 ms	55.6 ms	65.5 ms
1	0	0	$2^{21}/f_{XX}$	104.9 ms	111.1 ms	131.1 ms
Other than above			Setting prohibited			

Caution The wait time at the cancellation of the STOP mode does not include the time ("a" in the figure below) until clock oscillation starts after STOP mode is cancelled by $\overline{\text{RESET}}$ input or interrupt generation.



CHAPTER 11 SERIAL INTERFACE FUNCTION

11.1 Overview

The V850/SC1, V850/SC2, and V850/SC3 incorporate the following serial interfaces.

- Channel 0: 3-wire serial I/O (CSI0)/I^2C0[Note]
- Channel 2: 3-wire serial I/O (CSI2)/I^2C1[Note]
- Channel 3: 3-wire serial I/O (CSI3)/asynchronous serial interface (UART1)
- Channel 4: 3-wire serial I/O (CSI4)/asynchronous serial interface (UART0)
- Channel 5: 3-wire serial I/O (CSI5)
- Channel 6: 3-wire serial I/O (CSI6)
- Channel 7: Asynchronous serial interface (UART2)
- Channel 8: Asynchronous serial interface (UART3)

Note I^2C0 and I^2C1 support multimasters.
Either 3-wire serial I/O or I^2C can be used as a serial interface.

11.2 3-Wire Serial I/O (CSI0, CSI2, CSI3): 8 Bits

CSIn (n = 0, 2, 3) has the following two modes.

(1) Operation stopped mode

This mode is used when serial transfers are not performed.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{SCKn}$), a serial output line (SOn), and a serial input line (SIn).

Since data can be transmitted and received simultaneously 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 as the MSB.

★ The $\overline{SCK0}$ and $\overline{SCK2}$ pins can be used to select normal output and N-ch open-drain output, respectively, by setting the port 1 function register and port 2 function register (PF1, PF2).

3-wire serial I/O mode is useful for connection to devices such as peripheral I/O that include a clocked serial interface, and display controllers.

11.2.1 Configuration

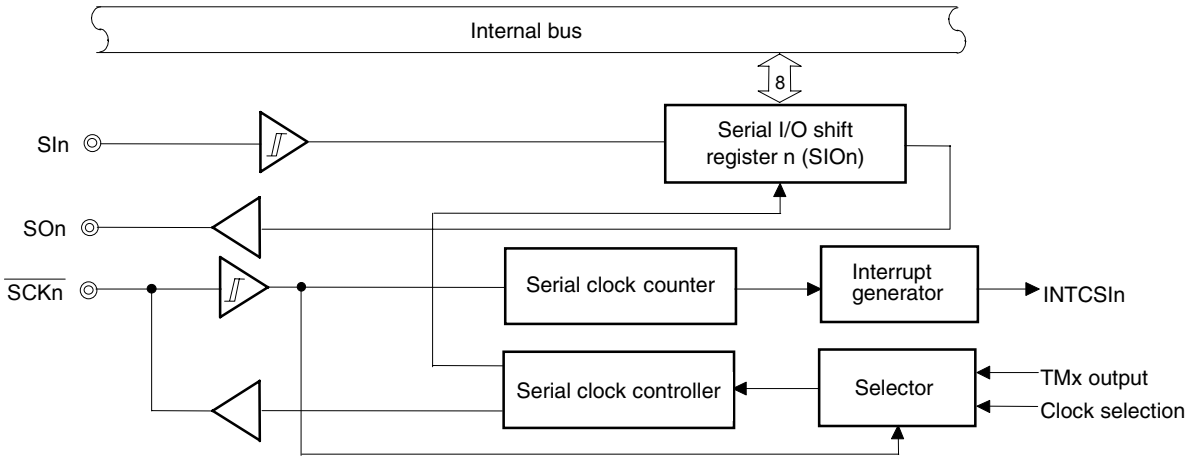
CSIn includes the following hardware.

Table 11-1. Configuration of CSIn

Item	Configuration
Registers	Serial I/O shift register n (SIOn)
Control registers	Serial operation mode register n (CSIMn)
	Serial clock select register n (CSISn)

Remark n = 0, 2, 3

Figure 11-1. Block Diagram of 3-Wire Serial I/O (CSI0, CSI2, CSI3)



Remarks 1. n = 0, 2, 3

2. TMx output is as shown below.

When n = 0, 3: TM5 output

When n = 2: TM6 output

(1) Serial I/O shift register n (SIOn)

SIOn is an 8-bit register that performs parallel-serial conversion and serial transmission/reception (shift operations) in synchronization with the serial clock.

SIOn is set by an 8-bit memory manipulation instruction.

When bit 7 (CSIEn) of serial operation mode register n (CSIMn) is set to 1, 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 sets these registers to 00H.

Caution Do not access SIOn except by the transfer start trigger during a transfer operation (read is disabled when MODEn = 0 and write is disabled when MODEn = 1).

Remark n = 0, 2, 3

11.2.2 CSIn control registers

CSIn is controlled by the following registers.

- Serial operation mode register n (CSIMn)
- Serial clock select register n (CSISn)

(1) Serial clock select register n (CSISn) and serial operation mode register n (CSIMn)

CSISn is used to set the serial clock of serial interface channel n.

CSISn can be set by an 8-bit memory manipulation instruction (n = 0, 2, 3).

$\overline{\text{RESET}}$ input sets CSISn to 00H.

CSIMn is used to enable or disable the serial clock, operation modes, and specific operations of serial interface channel n.

CSIMn can be set by an 8-bit or 1-bit memory manipulation instruction (n = 0, 2, 3).

$\overline{\text{RESET}}$ input sets CSIMn to 00H.

After reset: 00H R/W Address: CSIS0 FFFFF2A4H
CSIS2 FFFFF2C4H
CSIS3 FFFFF2D4H

	7	6	5	4	3	2	1	0
CSISn	0	0	0	0	0	0	0	SCLn2

(n = 0, 2, 3)

After reset: 00H R/W Address: CSIM0 FFFFF2A2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	<7>	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0, 2, 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 ($\overline{SCKn}$)
0	0	1	at n = 0, 3: TM5 output at n = 2: TM6 output
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	$f_{XX}/32$
1	1	1	$f_{XX}/64$

Notes 1. The SIn, SOn, and $\overline{SCKn}$ pins are used as port function pins when CSIEn = 0 (SIOn operation stopped state).

2. When CSIEn = 1 (SIOn operation enabled state), 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 on SCLn1 and SCLn0.

★ **2. Always set bits 6 to 3 to CSIMn to 0.**

Remark If the selected clock is specified as a timer output, the P17/T05/TI5 and P30/T06/TI6 pins do not need to be in timer output mode.

11.2.3 Operations

CSIn has the following two operation modes.

- Operation stopped mode
- 3-wire serial I/O mode

(1) Operation stopped mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

In operation stopped mode, the SIn, SOn, and $\overline{\text{SCKn}}$ pins can be used as normal I/O port pins.

(a) Register settings

Operation stopped mode is set via the CSIEn bit of serial operation mode register n (CSIMn).

Figure 11-2. CSIMn Setting (Operation Stopped Mode)

After reset: 00H R/W Address: CSIM0 FFFFF2A2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	<7>	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0, 2, 3)

CSIEn	SIOn operation enable/disable specification		
	Shift register operation	Serial counter	Port
0	Operation disabled	Cleared	Port function

(2) 3-wire serial I/O mode

3-wire serial I/O mode is useful when connecting to devices such as peripheral I/O that include a clocked serial interface, and display controllers.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCKn}$), a serial output line (SOn), and a serial input line (SIn).

(a) Register settings

3-wire serial I/O mode is set by serial operation mode register n (CSIMn).

Figure 11-3. CSIMn Setting (3-Wire Serial I/O Mode)

After reset: 00H R/W Address: CSIM0 FFFFF2A2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	<7>	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0, 2, 3)

CSIEn	SIOn operation enable/disable specification		
	Shift register operation	Serial counter	Port
1	Operation enabled	Count operation enabled	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: TM5 output when n = 2: TM6 output
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	$f_{XX}/32$
1	1	1	$f_{XX}/64$

Remarks 1. Refer to **11.2.2 (1) Serial clock select register n (CSISn) and serial operation mode register n (CSIMn)** for the SCLn2 bit.

2. If the selected clock is specified as a timer output, the P17/T05/TI5 and P30/T06/TI6 pins do not need to be in timer output mode.

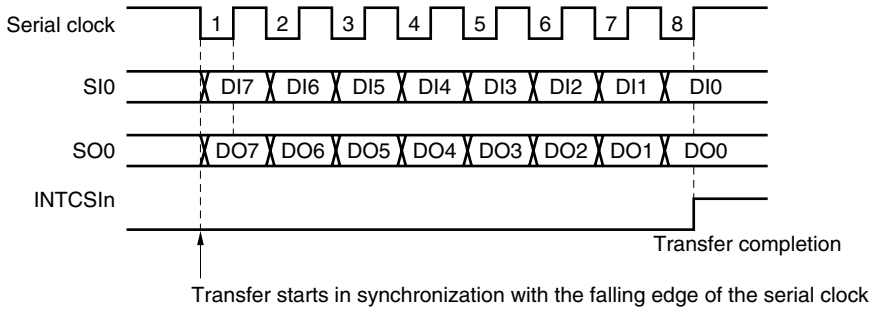
(b) Communication operations

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. Transmit 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 11-4. Timing of 3-Wire Serial I/O Mode



(c) Transfer start

A serial transfer starts when the following two conditions have been satisfied and transfer data has been set to serial I/O shift register n (SIOn).

- The SIOn operation control bit (CSIEn) = 1
- After an 8-bit serial transfer, the internal serial clock is either stopped or is set to high level.

The transfer data is set to SIOn as follows.

- Transmit/receive mode
  When CSIEn = 1 and MODEn = 0, transfer starts when writing to SIOn.
- Receive-only mode
  When CSIEn = 1 and MODEn = 1, transfer starts when reading from SIOn.

Caution After data has been written to SIOn, transfer will not start even if the CSIEn bit value is set to 1.

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets the interrupt request flag (INTCSIn).

11.3 3-Wire Serial I/O (CSI4): 8 to 16 Bits Variable

CSI4 has the following two operation modes.

(1) Operation stopped mode

This mode is used when serial transfers are not performed.

(2) 3-wire serial I/O mode (MSB/LSB-first switchable)

This mode transfers variable data of 8 to 16 bits via three lines: a serial clock line ($\overline{SCK4}$), a serial output line (SO4), and a serial input line (SI4).

Since data can be transmitted and received simultaneously in 3-wire serial I/O mode, the processing time for data transfer is reduced.

The first bit of serial transfer data can be switched between MSB and LSB.

3-wire serial I/O mode is useful for connection to devices such as peripheral I/O that include a clocked serial interface, and display controllers.

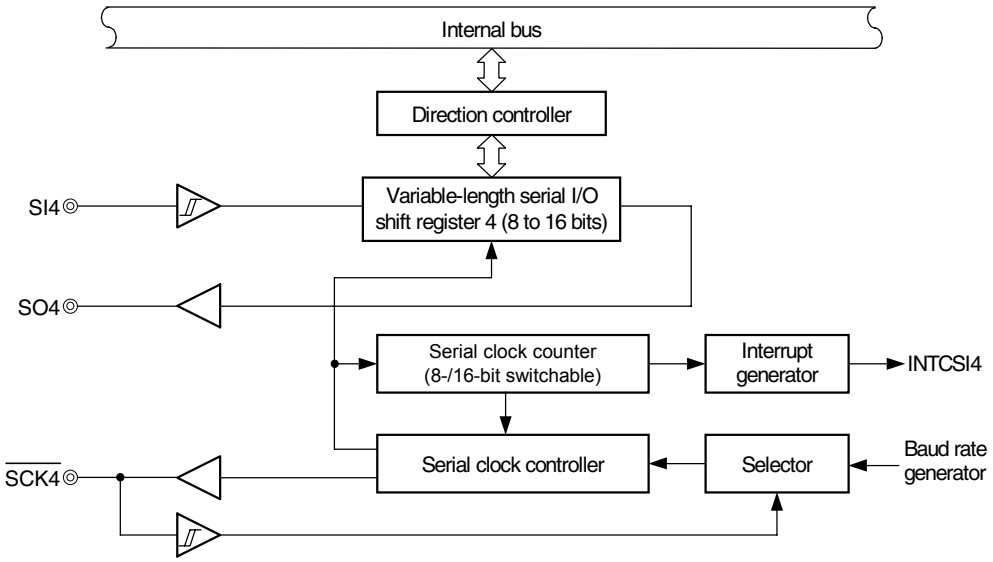
11.3.1 Configuration

CSI4 includes the following hardware.

Table 11-2. Configuration of CSI4

Item	Configuration
Registers	Variable-length serial IO 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 select register 4 (BRGCN4) Baud rate generator output clock select register 4 (BRGCK4)

Figure 11-5. Block Diagram of 3-Wire Serial I/O (CSI4)



(1) Variable-length serial I/O shift register 4 (SIO4)

SIO4 is a 16-bit variable register that performs parallel-serial conversion and transmission/reception (shift operations) in synchronization with the serial clock.

SIO4 is set by a 16-bit memory manipulation instruction.

When bit 7 (CSIE4) of variable-length serial control register 4 (CSIM4) is set to 1, a serial operation can be started by writing data to or reading data from SIO4.

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.

$\overline{\text{RESET}}$ input sets SIO4 to 0000H.

Caution Do not access SIO4 except by the transfer start trigger during a transfer operation (read is disabled when MODE4 = 0 and write is disabled when MODE4 = 1).

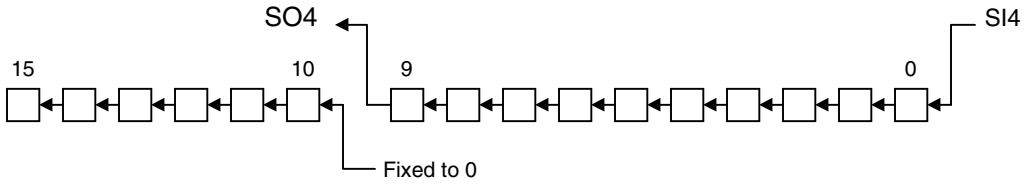
After reset: 0000H R/W Address: FFFFF2E0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIO4																

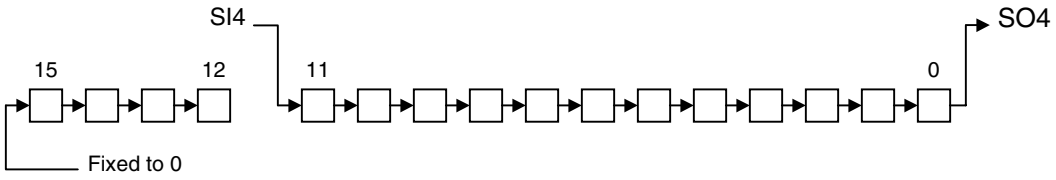
If the transfer bit length is set to other than 16 bits, when setting data to the shift register, be sure to align data from the lowest bit, regardless of whether the first transfer bit is MSB or LSB. Any data can be set to the unused higher bits, but in this case the data received after a serial transfer operation becomes 0.

Figure 11-6. 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



11.3.2 CSI4 control registers

CSI4 is controlled by the following registers.

- Variable-length serial control register 4 (CSIM4)
- Variable-length serial setting register 4 (CSIB4)
- Baud rate generator source clock select register 4 (BRGCN4)
- Baud rate generator output clock select register 4 (BRGCK4)

(1) Variable-length serial control register 4 (CSIM4)

This register is used to enable or disable the serial clock, operation modes, and specific operations of serial interface channel 4.

CSIM4 can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets 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. The SI4, SO4, and $\overline{\text{SCK4}}$ pins are used as port function pins when CSIE4 = 0 (SIO4 operation disabled state).

2. When CSIE4 = 1 (SIO4 operation enable state), the port function is available only for the SI4 pin when only using the transmit function and to SO4 pin when using the receive-only function.

(2) Variable-length serial setting register 4 (CSIB4)

This register 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 when bit 4 (DIR) is 1, and is transferred LSB first when DIR is 0.

CSIB4 can be set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets 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 select register 4 (BRGCN4)

BRGCN4 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets 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 select 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 $1 \leq k \leq 127$>

  $$[\text{Baud rate}] = \frac{f_{XX}}{2^n \times k \times 2} \text{ [Hz]}$$

  f_{XX}: Main clock oscillation frequency

  n: Value set by BRGN2 to BRGN0 ($0 \leq n \leq 7$)

  k: Value set by BRGK6 to BRGK0 ($1 \leq k \leq 127$)

Caution Do not use the baud rate transmit/receive clock of CSI4 with a transfer rate higher than the CPU operation clock. If used with a transfer rate higher than the CPU operation clock, transfer cannot be performed correctly.

11.3.3 Operations

CSI4 has the following two operation modes.

- Operation stopped mode
- 3-wire serial I/O mode

(1) Operation stopped mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

In operation stopped mode, the SI4, SO4, and $\overline{\text{SCK4}}$ pins can be used as normal I/O port pins.

(a) Register settings

Operation stopped mode is set via the CSIE4 bit of variable-length serial control register 4 (CSIM4). When CSIE4 = 0 (SIO4 operation stop state), the pins connected to SI4, SO4, or $\overline{\text{SCK4}}$ function as port pins.

Figure 11-7. CSIM4 Setting (Operation Stopped 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	Cleared	Port function

(2) 3-wire serial I/O mode

3-wire serial I/O mode is useful when connecting to devices such as peripheral I/O that include a clocked serial interface, and display controllers.

This mode executes data transfers via three lines: a serial clock line ($\overline{SCK4}$), a serial output line (SO4), and a serial input line (SI4).

(a) Register settings

3-wire serial I/O mode is set by the variable-length serial control register 4 (CSIM4).

Figure 11-8. CSIM4 Setting (3-Wire 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 when bit 4 (DIR) is 1, and is transferred LSB first when DIR is 0.

Figure 11-9. CSIB4 Setting (3-Wire 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	$\overline{\text{SCK4}}$ 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

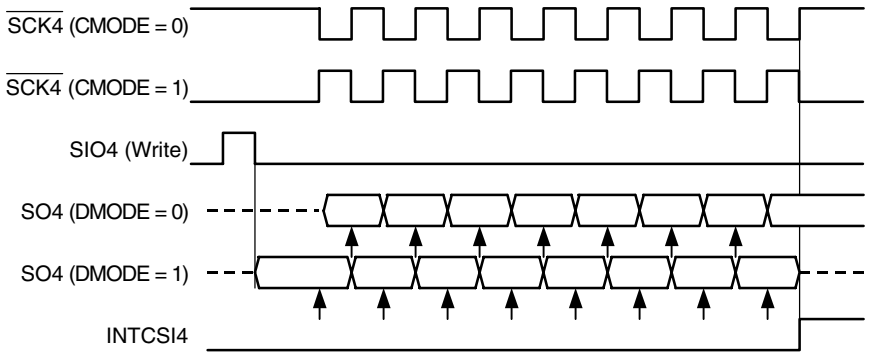
(b) Communication operations

In 3-wire serial I/O mode, data is transmitted and received in 8 to 16-bit units, 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, SIO4 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 be used to change the attribute of the serial clock ($\overline{SCK4}$) and the phases of serial data (SI4 and SO4).

Figure 11-10. Timing of 3-Wire Serial I/O Mode



Remark The arrows show the SI4 data fetch timing.

When CMODE = 0, the serial clock ($\overline{SCK4}$) stops at a high level during the operation stop, and outputs a low level during a data transfer operation. When CMODE = 1, on the other hand, $\overline{SCK4}$ stops at a low level during the operation stop and outputs a 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 ($\overline{SCK4}$), regardless of 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.

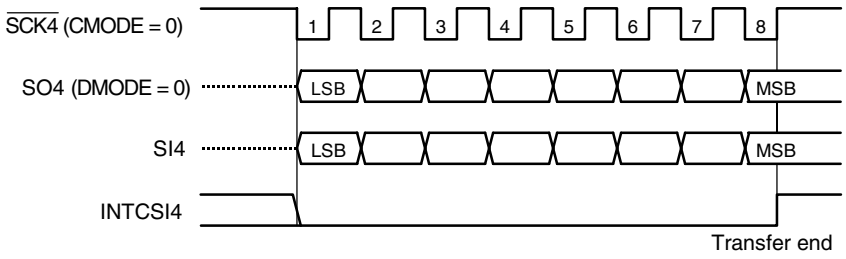
Serial transfer starts when the following operation is performed after the above two conditions have been satisfied.

- Transmit and transmit/receive mode (MODE4 = 0)
  Transfer starts when writing to SIO4.
- Receive-only mode (MODE4 = 1)
  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 11-11. Timing of 3-Wire Serial I/O Mode (When CSIB4 = 08H)



Remark CSIB4 = 08H (CMODE = 0, DMODE = 0, DIR = 0, BSEL3 to BSEL0 = 1000)

11.4 3-Wire Serial I/O (CSI5, CSI6): 8 or 16 Bits

11.4.1 Features

- High-speed transfer: Maximum 4 Mbps
- Half-duplex communications
- Master mode or slave mode can be selected
- Transmit data length: 8 bits or 16 bits can be set
- Transfer data direction can be switched between MSB first and LSB first
- Eight clock signals can be selected (7 master clocks and 1 slave clock)
- 3-wire type SOn: Serial transmit data output
  SIn: Serial receive data input
  $\overline{\text{SCK}}$n: Serial clock I/O
- Interrupt sources: 1 type
  - Transmission/reception completion interrupt (INTCSIn)
- Transmit/receive mode and receive-only mode can be specified
- Two transmit buffers (SOTBFn/SOTBFLn, SOTBn/SOTBLn) and two receive buffers (SIRBn/SIRBLn, SIRBEn/SIRBELn) are provided on chip
- Single transfer mode and repeat transfer mode can be specified

Caution When using P120/$\overline{\text{SCK5}}$, P121/SI5, and P122/SO5 for CSI5 transmission/reception and P123/$\overline{\text{SCK6}}$, P124/SI6, and P125/SO6 for CSI6 transmission/reception, set the CSI5 pin function ($\overline{\text{SCK5}}$, SI5, SO5) and CSI6 pin function ($\overline{\text{SCK6}}$, SI6, SO6) using port alternate-function control register 2 (PAC2) (refer to 5.2.11 (2) (b) Port alternate-function register 2 (PAC2) and 5.3 Setting When Port Pin Is Used for Alternate Function).

Remark n = 5, 6

11.4.2 Configuration

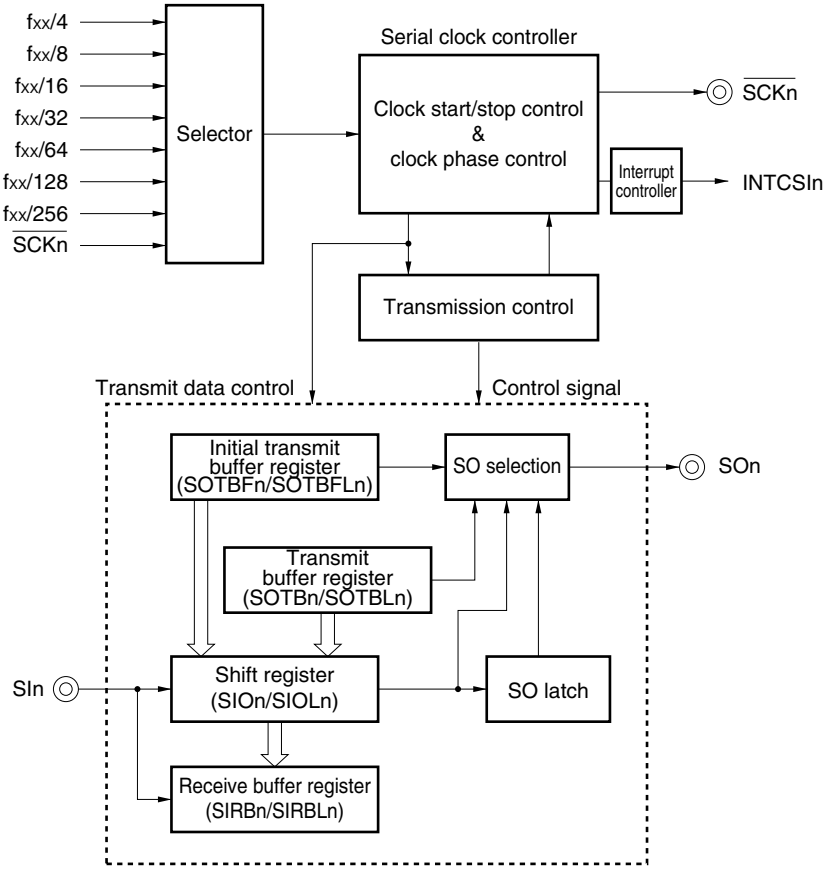
CSIn includes the following hardware (n = 5, 6).

Table 11-3. CSIn Configuration

Item	Configuration
Registers	Serial I/O shift register n, Ln (SIOn, SIOLn)
Control registers	Clocked serial interface mode register n (CSIMn) Clocked serial interface clock select register n (CSICKn) Clocked serial interface receive buffer register n, Ln (SIRBn, SIRBLn) Clocked serial interface read-only receive buffer register n, Ln (SIRBEn, SIRBELn) Clocked serial interface transmit buffer register n , Ln (SOTBn, SOTBLn) Clocked serial interface initial transmit buffer register (SOTBFn, SOTBFLn)

Remark n = 5, 6

Figure 11-12. Block Diagram of 3-Wire Serial I/O (CSI5, CSI6)



Remark n = 5, 6

fxx: Main clock frequency

Transmission/reception of data is performed by the SIOn register (n = 0, 1).

(1) Serial I/O shift register n (SIOn)

The SIOn register is a 16-bit shift register that converts parallel data into serial data, and is used for both transmission and reception.

Data is shifted in (reception) and shifted out (transmission) from the MSB or LSB side.

The actual transmit/receive operations are started up by accessing the buffer register.

Remark n = 5, 6

(2) Serial I/O shift register Ln (SIOLn)

The SIOLn register is an 8-bit shift register that converts parallel data into serial data, and is used for both transmission and reception.

Data is shifted in (reception) and shifted out (transmission) from the MSB or LSB side.

The actual transmit/receive operations are started up by accessing the buffer register.

Remark n = 5, 6

(3) Selector

The selector selects the serial clock to be used.

(4) Serial clock controller

The serial clock controller controls the serial clock supply to the shift register, as well as the clock output to the $\overline{\text{SCKn}}$ pin when the internal clock is used.

(5) Serial clock counter

The serial clock counter counts the serial clock output or input during transmission/reception, and checks whether 8-bit or 16-bit data transmission/reception has been performed.

(6) Interrupt controller

The interrupt controller controls the interrupt request timing.

11.4.3 Control registers

CSIn is controlled by the following registers (n = 5, 6).

- Clocked serial interface mode register n (CSIMn)
- Clocked serial interface clock select register n (CSICKn)
- Clocked serial interface receive buffer register n, Ln (SIRBn, SIRBLn)
- Clocked serial interface read-only receive buffer register n, Ln (SIRBEn, SIRBELn)
- Clocked serial interface transmit buffer register n, Ln (SOTBn, SOTBLn)
- Clocked serial interface initial transmit buffer register n, Ln (SOTBFn, SOTBFLn)

(1) Clocked serial interface mode registers 5, 6 (CSIM5, CSIM6)

These registers control CSIn operations (n = 5, 6).

CSIMn can be set by an 8-bit or 1-bit memory manipulation instruction.

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

Caution The TRMDn, CCLn, DIRn, CSITn, and AUTOn bits of the CSIMn register can be overwritten only when the CSOTn bit = 0. If these bits are overwritten at any other time, the operation cannot be guaranteed.

After reset: 00H R/W Address: FFFFF240H, FFFFF260H

CSIMn (n = 5, 6)

7	6	5	4	3	2	1	0
CSIEn	TRMDn	CCLn	DIRn	CSITn	AUTOn	0	CSOTn[Note]

CSIEn	CSIn operation enable/disable specification
0	CSIn operation enabled
1	CSIn operation disabled
The internal CSIn circuit can be reset asynchronously by setting the CSIEn bit to 0. For the $\overline{SCKn}$ and SOn pin output status when the CSIEn bit = 0, refer to **11.6.5 Output pins**.	

TRMDn	Transmission/reception mode specification
0	Receive-only mode
1	Transmit/receive mode
When the TRMDn bit = 0, receive-only transfer is performed and the SOn pin output is fixed to low level. Data reception is started by reading the SIRBn register. When the TRMDn bit = 1, transmission/reception is started by writing data to the SOTBn register.	

CCLn	Data length specification
0	8 bits
1	16 bits

DIRn	Transfer direction mode (MSB/LSB) specification
0	First bit of transfer data is MSB
1	First bit of transfer data is LSB

CSITn	Delay of interrupt request signal control
0	No delay
1	Delay mode (interrupt request signal is delayed 1/2 cycle).

AUTOn	Single transfer mode or repeat transfer mode specification
0	Single transfer mode
1	Repeat transfer mode

CSOTn	Flag indicating transfer status
0	Idle status
1	Transfer execution status

Note Bit 0 (CSOTn) can only be read.

Cautions 1. The delay mode (CSITn bit = 1) is valid only in the master mode (CKSn2 to CSKn0 bits of the CSICKn register are not 111B). In the slave mode (CKSn2 to CKSn0 bits are 111B), do not set the delay mode.

2. The CSOTn bit is cleared (0) by writing 0 to the CSIEn bit.

(2) Clocked serial interface clock select registers 5, 6 (CSICK5, CSICK6)

The CSICKn register is an 8-bit register that controls the CSIn transfer operation (n = 5, 6).

CSICKn can be set by an 8-bit or 1-bit memory manipulation instruction.

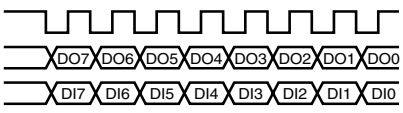
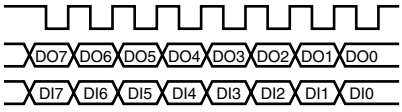
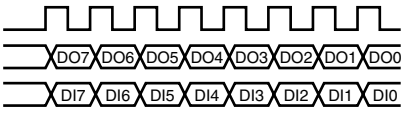
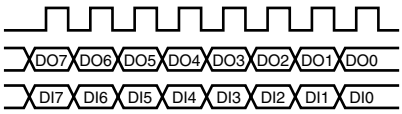
$\overline{\text{RESET}}$ input sets these registers to 00H.

Caution The CSICKn register can be overwritten only when the CSIE bit of the CSIMn register = 0.

After reset: 00H R/W Address: FFFFF242H, FFFFF262H

	7	6	5	4	3	2	1	0
CSICKn	0	0	0	CKPn	DAPn	CKSn2	CKSn1	CKSn0

(n = 5, 6)

CKPn	DAPn	Operation mode
0	0	
0	1	
1	0	
1	1	

CKSn2	CKSn1	CKSn0	Input clock selection	Mode
0	0	0	$f_{XX}/4$	Master mode
0	0	1	$f_{XX}/8$	Master mode
0	1	0	$f_{XX}/16$	Master mode
0	1	1	$f_{XX}/32$	Master mode
1	0	0	$f_{XX}/64$	Master mode
1	0	1	$f_{XX}/128$	Master mode
1	1	0	$f_{XX}/256$	Master mode
1	1	1	External clock ($\overline{\text{SCKn}}$)	Slave mode

Remark f_{XX}: Main clock frequency

(3) Clocked serial interface receive buffer registers 5, 6 (SIRB5, SIRB6)

The SIRBn register is a 16-bit buffer register that stores receive data.

When the receive-only mode is set (TRMDn bit of CSIMn register = 0), reception is started by reading data from the SIRBn register (n = 5, 6).

SIRBn is set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 0000H.

In addition to $\overline{\text{RESET}}$ input, these registers can also be initialized by clearing (0) the CSIEn bit of the CSIMn register.

After reset: 0000H R Address: FFFFF244H, FFFFF264H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRBn	SIRB n15	SIRB n14	SIRB n13	SIRB n12	SIRB n11	SIRB n10	SIRB n9	SIRB n8	SIRB n7	SIRB n6	SIRB n5	SIRB n4	SIRB n3	SIRB n2	SIRB n1	SIRB n0

(n = 5, 6)

Cautions 1. Read the SIRBn register only when a data length of 16 bits has been set (CCLn bit of CSIMn register = 1).

2. When the single transfer mode has been set (AUTOn bit of CSIMn register = 0), perform a read operation only in the idle state (CSOTn bit of CSIMn register = 0). If the SIRBn register is read during data transfer, the data cannot be guaranteed.

(4) Clocked serial interface receive buffer registers L5, L6 (SIRBL5, SIRBL6)

The SIRBLn register is an 8-bit buffer register that stores receive data.

When the receive-only mode is set (TRMDn bit of CSIMn register = 0), reception is started by reading data from the SIRBLn register.

SIRBLn is set by an 8-bit memory manipulation instruction (n = 5, 6).

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

In addition to $\overline{\text{RESET}}$ input, these registers can also be initialized by clearing (0) the CSIEn bit of the CSIMn register.

The SIRBLn register is the same as the lower bytes of the SIRBn register.

After reset: 00H R Address: FFFFF246H, FFFFF266H

	7	6	5	4	3	2	1	0
SIRBLn	SIRBn7	SIRBn6	SIRBn5	SIRBn4	SIRBn3	SIRBn2	SIRBn1	SIRBn0

(n = 5, 6)

Cautions 1. Read the SIRBLn register only when a data length of 8 bits has been set (CCLn bit of CSIMn register = 0).

2. When the single transfer mode is set (AUTOn bit of CSIMn register = 0), perform a read operation only in the idle state (CSOTn bit of CSIMn register = 0). If the SIRBLn register is read during data transfer, the data cannot be guaranteed.

(5) Clocked serial interface read-only receive buffer registers 5, 6 (SIRBE5, SIRBE6)

The SIRBEn register is a 16-bit buffer register that stores receive data.

SIRBEn is set by a 16-bit memory manipulation instruction (n = 5, 6).

$\overline{\text{RESET}}$ input sets these registers to 0000H.

In addition to $\overline{\text{RESET}}$ input, these registers can also be initialized by clearing (0) the CSIEn bit of the CSIMn register.

The SIRBEn register is the same as the SIRBn register. It is used to read the contents of the SIRBn register.

After reset: 0000H R Address: FFFFF24CH, FFFFF26CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRBEn	SIRBEn15	SIRBEn14	SIRBEn13	SIRBEn12	SIRBEn11	SIRBEn10	SIRBEn9	SIRBEn8	SIRBEn7	SIRBEn6	SIRBEn5	SIRBEn4	SIRBEn3	SIRBEn2	SIRBEn1	SIRBEn0

(n = 5, 6)

Cautions 1. A receive operation is not started even if data is read from the SIRBEn register.

2. The SIRBEn register can be read only if a data length of 16 bits is set (CCLn bit of CSIMn register = 1).

(6) Clocked serial interface read-only receive buffer registers L5, L6 (SIRBEL5, SIRBEL6)

The SIRBELn register is an 8-bit buffer register that stores receive data.

★ SIRBELn is set by an 8-bit or 1-bit memory manipulation instruction (n = 5, 6).

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

In addition to $\overline{\text{RESET}}$ input, these registers can also be initialized by clearing (0) the CSIEn bit of the CSIMn register.

The SIRBELn register is the same as the SIRBLn register. It is used to read the contents of the SIRBLn register.

After reset: 00H R Address: FFFFF24EH, FFFFF26EH

	7	6	5	4	3	2	1	0
SIRBELn	SIRBEn7	SIRBEn6	SIRBEn5	SIRBEn4	SIRBEn3	SIRBEn2	SIRBEn1	SIRBEn0

(n = 5, 6)

Cautions 1. A receive operation is not started even if data is read from the SIRBELn register.

2. The SIRBELn register can be read only if a data length of 8 bits has been set (CCLn bit of CSIMn register = 0).

(7) Clocked serial interface transmit buffer registers 5, 6 (SOTB5, SOTB6)

The SOTBn register is a 16-bit buffer register that stores transmit data.

When the transmit/receive mode is set (TRMDn bit of CSIMn register = 1), transmission is started by writing data to the SOTBn register (n = 5, 6).

SOTBn is set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 0000H.

After reset: 0000H R/W Address: FFFFF248H, FFFFF268H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOTBn	SOTBn15	SOTBn14	SOTBn13	SOTBn12	SOTBn11	SOTBn10	SOTBn9	SOTBn8	SOTBn7	SOTBn6	SOTBn5	SOTBn4	SOTBn3	SOTBn2	SOTBn1	SOTBn0

(n = 5, 6)

Cautions 1. Access the SOTBn register only when a data length of 16 bits is set (CCLn bit of CSIMn register = 1).

2. When the single transfer mode is set (AUTOn bit of CSIMn register = 0), perform access only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBn register is accessed during data transfer, the data cannot be guaranteed.

(8) Clocked serial interface transmit buffer registers L5, L6 (SOTBL5, SOTBL6)

The SOTBLn register is an 8-bit buffer register that stores transmit data.

When the transmit/receive mode is set (TRMDn bit of CSIMn register = 1), transmission is started by writing data to the SOTBLn register (n = 5, 6).

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

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

The SOTBLn register is the same as the lower bytes of the SOTBn register.

After reset: 00H R/W Address: FFFFF24AH, FFFFF26AH

	7	6	5	4	3	2	1	0
SOTBLn	SOTBn7	SOTBn6	SOTBn5	SOTBn4	SOTBn3	SOTBn2	SOTBn1	SOTBn0

(n = 5, 6)

Cautions 1. Access the SOTBLn register only when a data length of 8 bits has been set (CCLn bit of CSIMn register = 0).

2. When the single transfer mode is set (AUTOn bit of CSIMn register = 0), perform access only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBLn register is accessed during data transfer, the data cannot be guaranteed.

(9) Clocked serial interface initial transmit buffer registers 5, 6 (SOTBF5, SOTBF6)

The SOTBFn register is a 16-bit buffer register that stores the initial transmit data in the repeat transfer mode.

Transmission is not started even if data is written to the SOTBFn register (n = 5, 6).

SOTBFn can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 0000H.

After reset: 0000H R/W Address: FFFFF250H, FFFFF270H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOTBFn	SOTBF n15	SOTBF n14	SOTBF n13	SOTBF n12	SOTBF n11	SOTBF n10	SOTBF n9	SOTBF n8	SOTBF n7	SOTBF n6	SOTBF n5	SOTBF n4	SOTBF n3	SOTBF n2	SOTBF n1	SOTBF n0

(n = 5, 6)

Caution Access the SOTBFn register only when a data length of 16 bits has been set (CCLn bit of CSIMn register = 1), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBFn register is accessed during data transfer, the data cannot be guaranteed.

(10) Clocked serial interface initial transmit buffer registers L5, L6 (SOTBFL5, SOTBFL6)

The SOTBFLn register is an 8-bit buffer register that stores the initial transmit data in the repeat transfer mode.

Transmission is not started even if data is written to the SOTBFLn register (n = 5, 6).

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

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

The SOTBFLn register is the same as the lower bytes of the SOTBFn register.

After reset: 00H R/W Address: FFFFF252H, FFFFF272H

	7	6	5	4	3	2	1	0
SOTBFLn	SOTBFn7	SOTBFn6	SOTBFn5	SOTBFn4	SOTBFn3	SOTBFn2	SOTBFn1	SOTBFn0

(n = 5, 6)

Caution Access the SOTBFLn register only when a data length of 8 bits has been set (CCLn bit of CSIMn register = 0), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SOTBFLn register is accessed during data transfer, the data cannot be guaranteed.

(11) Serial I/O shift registers 5, 6 (SIO5, SIO6)

The SIOn register is a 16-bit shift register that converts parallel data into serial data.

Data is shifted in (received) or shifted out (transmitted) starting from MSB or LSB.

Transfer is not started even if the SIOn register is read (n = 5, 6).

SIOn is set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to 0000H.

In addition to $\overline{\text{RESET}}$ input, these registers can also be initialized by clearing (0) the CSIEn bit of the CSIMn register.

After reset: 0000H R Address: FFFFF254H, FFFFF274H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIOn	SIO n15	SIO n14	SIO n13	SIO n12	SIO n11	SIO n10	SIO n9	SIO n8	SIO n7	SIO n6	SIO n5	SIO n4	SIO n3	SIO n2	SIO n1	SIO n0

(n = 5, 6)

Caution Access the SIOn register only when a data length of 16 bits has been set (CCLn bit of CSIMn register = 1), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SIOn register is accessed during data transfer, the data cannot be guaranteed.

(12) Serial I/O shift registers L5, L6 (SIOL5, SIOL6)

The SIOLn register is an 8-bit shift register that converts parallel data into serial data.

Data is shifted in (received) or shifted out (transmitted) starting from MSB or LSB.

Transfer is not started even if the SIOLn register is read (n = 5, 6).

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

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

In addition to $\overline{\text{RESET}}$ input, these registers can also be initialized by clearing (0) the CSIEn bit of the CSIMn register.

The SIOLn register is the same as the lower bytes of the SIOn register.

After reset: 00H R Address: FFFFF256H, FFFFF276H

	7	6	5	4	3	2	1	0
SIOLn	SIOn7	SIOn6	SIOn5	SIOn4	SIOn3	SIOn2	SIOn1	SIOn0

(n = 5, 6)

Caution Access the SIOLn register only when a data length of 8 bits has been set (CCLn bit of CSIMn register = 0), and only in the idle state (CSOTn bit of CSIMn register = 0). If the SIOLn register is accessed during data transfer, the data cannot be guaranteed.

11.4.4 Operation

CSIn is has the following two operation modes (n = 5, 6).

- Single transfer mode
- Repeat transfer mode

(1) Single transfer mode

(a) Usage

In the receive-only mode (TRMDn bit of CSIMn register = 0), transfer is started by reading[Note 1] the receive data buffer register (SIRBn/SIRBLn) (n = 5, 6).

In the transmit/receive mode (TRMDn bit of CSIMn register = 1), transfer is started by writing[Note 2] to the transmit data buffer register (SOTBn/SOTBLn).

In the slave mode, operation must be enabled beforehand (CSIEn bit of CSIMn register = 1).

When transfer is started, the value of the CSOTn bit of the CSIMn register becomes 1 (transmission execution status).

Upon transfer completion, the transmission/reception completion interrupt (INTCSIn) is set (1), and the CSOTn bit is cleared (0). The next data transfer request is then waited for.

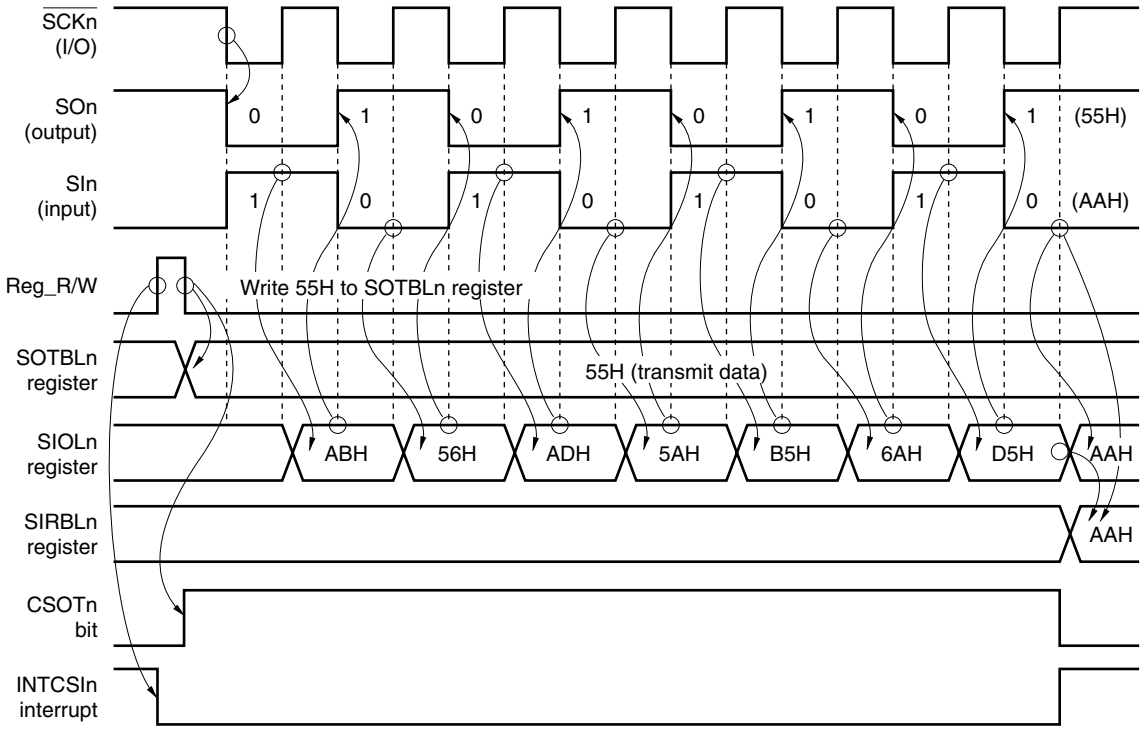
Notes 1. When a data length of 16 bits (CCLn bit of CSIMn register = 1) has been set, read the SIRBn register. When a data length of 8 bits (CCLn bit of CSIMn register = 0) has been set, read the SIRBLn register.

2. When a data length of 16 bits (CCLn bit of CSIMn register = 1) has been set, write to the SOTBn register. When a data length of 8 bits (CCLn bit of CSIMn register = 0) has been set, write to the SOTBLn register.

Caution When the CSOTn bit of the CSIMn register = 1, do not manipulate the CSIn register.

Figure 11-13. Timing Chart in Single Transfer Mode (1/2)

(a) In transmit/receive mode, data length: 8 bits, transfer direction: MSB first, no interrupt delay, single transfer mode, operation mode: CKPn bit = 0, DAPn bit = 0

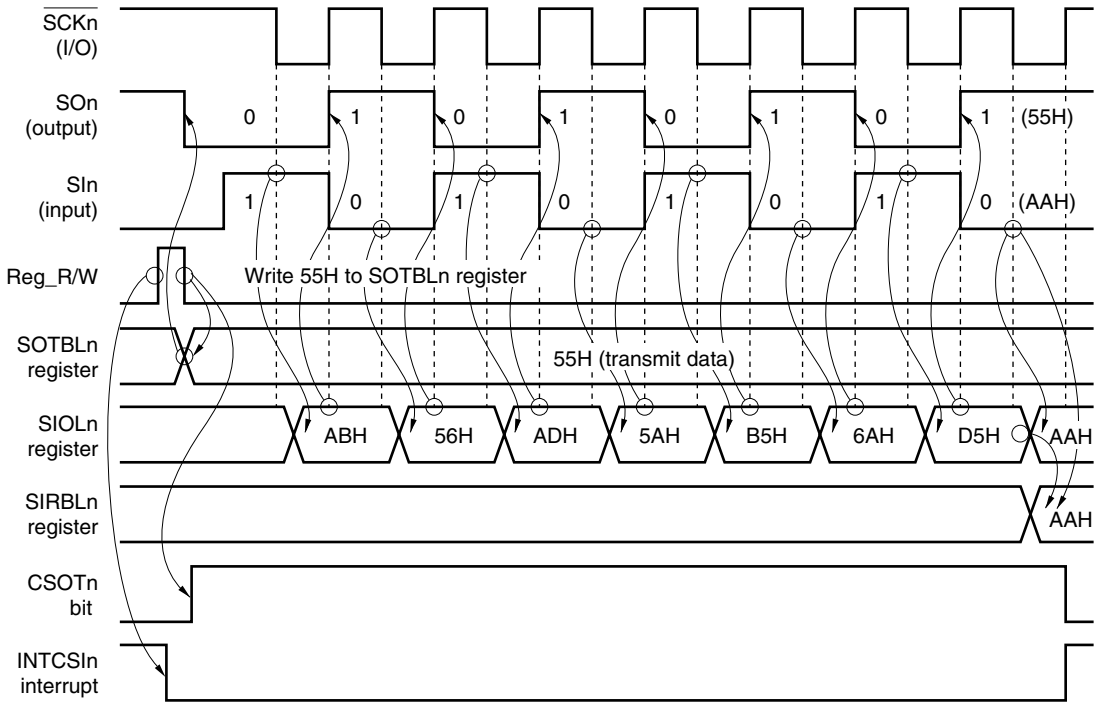


Remarks 1. n = 5, 6

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

Figure 11-13. Timing Chart in Single Transfer Mode (2/2)

(b) In transmit/receive mode, data length: 8 bits, transfer direction: MSB first, no interrupt delay, single transfer mode, operation mode: CKPn bit = 0, DAPn bit = 1



Remarks 1. n = 5, 6

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

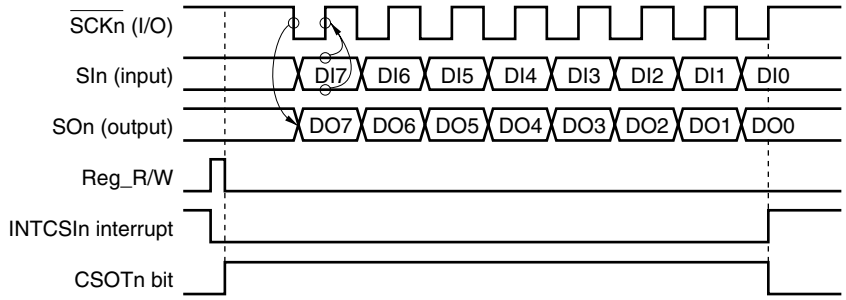
(b) Clock phase selection

The following shows the timing when changing the conditions for clock phase selection (CKPn bit of CSICn register) and data phase selection (DAPn bit of CSICn register) under the following conditions.

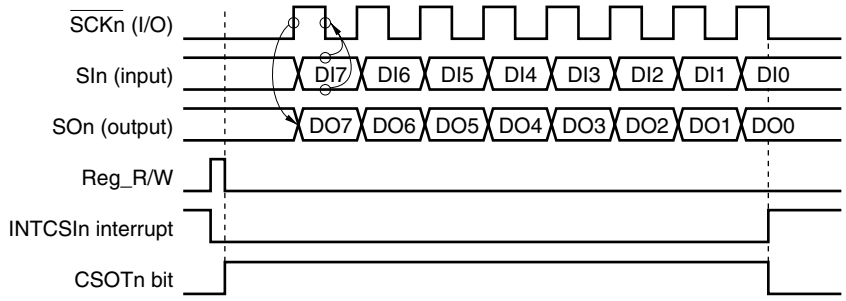
- Data length = 8 bits (CCLn bit of CSIMn register = 0)
- First bit of transfer data = MSB (DIRn bit of CSIMn register = 0)
- No interrupt request signal delay control (CSITn bit of CSIMn register = 0)

Figure 11-14. Timing Chart According to Clock Phase Selection (1/2)

(a) When CKPn bit = 0, DAPn bit = 0



(b) When CKPn bit = 1, DAPn bit = 0

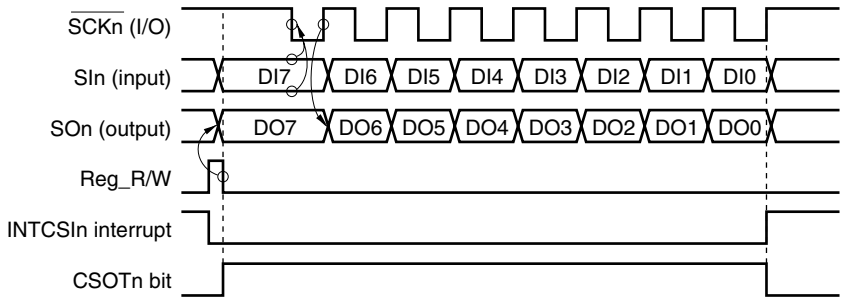


Remarks 1. n = 5, 6

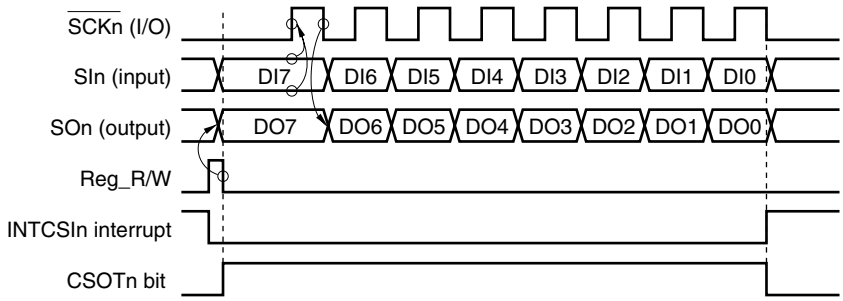
2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

Figure 11-14. Timing Chart According to Clock Phase Selection (2/2)

(c) When CKPn bit = 0, DAPn bit = 1



(d) When CKPn bit = 1, DAPn bit = 1



Remarks 1. n = 5, 6

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

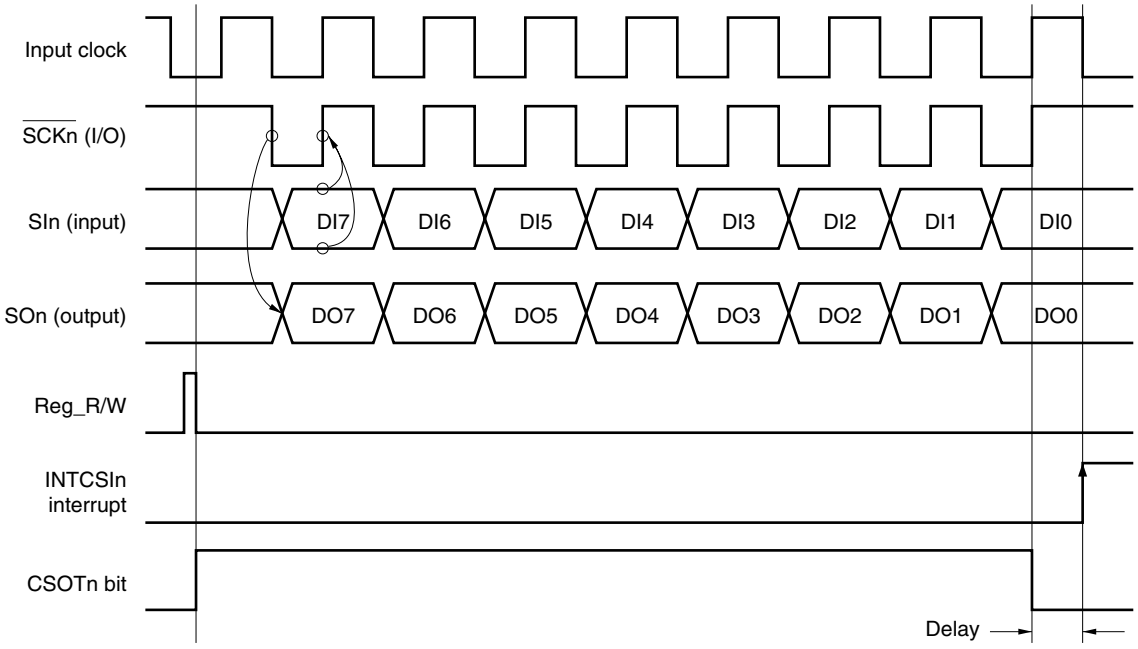
(c) Transmission/reception completion interrupt request signals (INTCSI5, INTCSI6)

INTCSIn is set (1) upon completion of data transmission/reception.

Caution The delay mode (CSITn bit = 1) is valid only in the master mode (bits CKSn2 to CKSn0 of the CSICKn register are not 111B). The delay mode cannot be set when the slave mode is set (bits CKSn2 to CKSn0 = 111B).

Figure 11-15. Timing Chart of Interrupt Request Signal Output in Delay Mode (1/2)

(a) When CKPn bit = 0, DAPn bit = 0

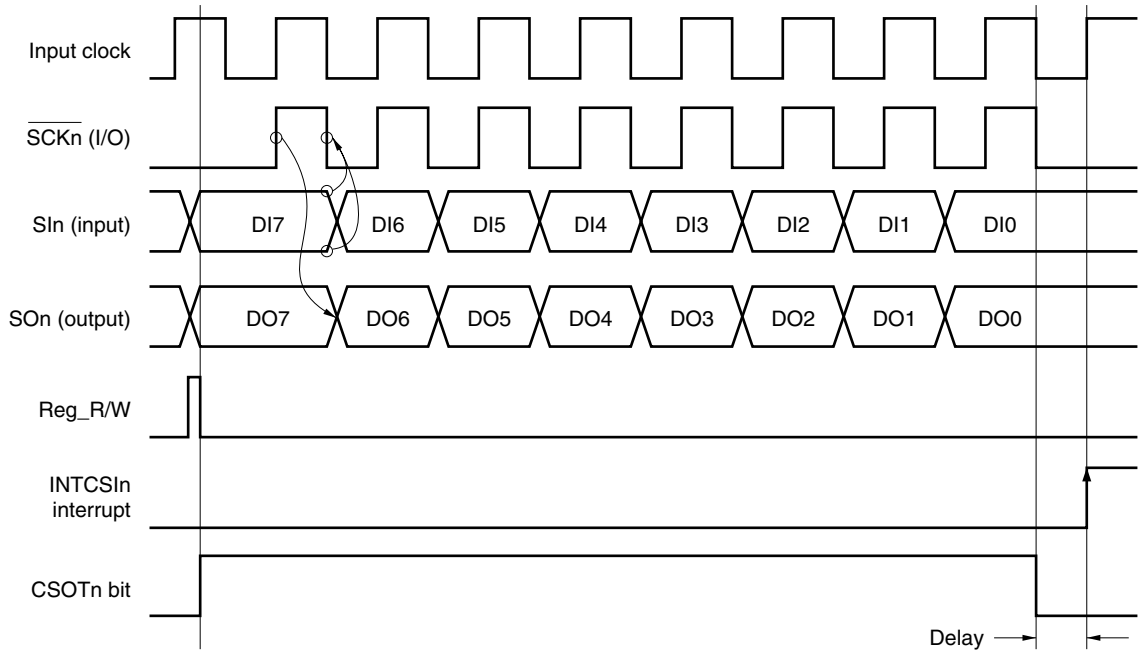


Remarks 1. n = 5, 6

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

Figure 11-15. Timing Chart of Interrupt Request Signal Output in Delay Mode (2/2)

(b) When CKPn bit = 1, DAPn bit = 1



Remarks 1. n = 5, 6

2. Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

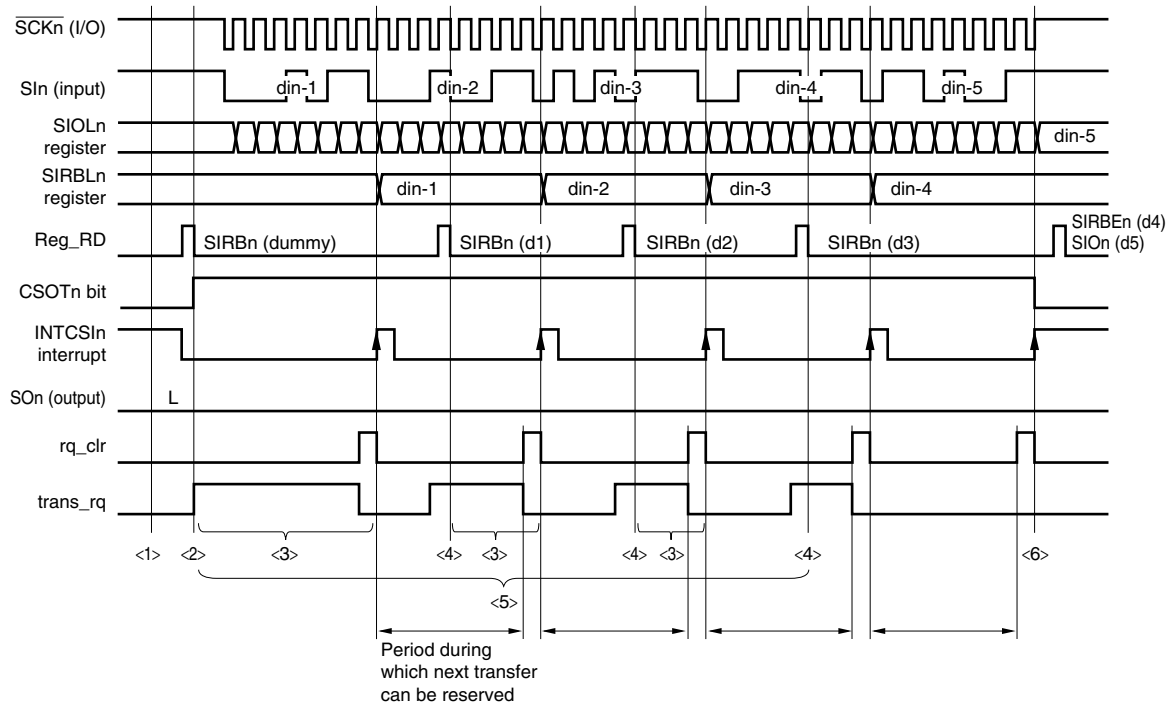
(2) Repeat transfer mode

(a) Usage (receive-only)

<1> Set the repeat transfer mode (AUTOn bit of CSIMn register = 1) and the receive-only mode (TRMDn bit of CSIMn register = 0).
<2> Read SIRBn register (start transfer with dummy read).
<3> Wait for transmission/reception completion interrupt request (INTCSIn).
<4> When the transmission/reception completion interrupt request (INTCSIn) has been set (1), read the SIRBn register[Note] (reserve next transfer).
<5> Repeat steps <3> and <4> (N – 2) times. (N: Number of transfer data)
<6> Following output of the last transmission/reception completion interrupt request (INTCSIn), read the SIRBEn register and the SIOn register[Note].

Note When transferring N number of data, receive data is loaded by reading the SIRBn register from the first data to the (N – 2)th data. The (N – 1)th data is loaded by reading the SIRBEn register, and the Nth (last) data is loaded by reading the SIOn register.

Figure 11-16. Repeat Transfer (Receive-Only) Timing Chart



Remarks 1. n = 5, 6

2. Reg_RD: Internal signal. This signal indicates that the receive data buffer register (SIRBn/SIRBLn) has been read.

rq_clr: Internal signal. Transfer request clear signal.

trans_rq: Internal signal. Transfer request signal.

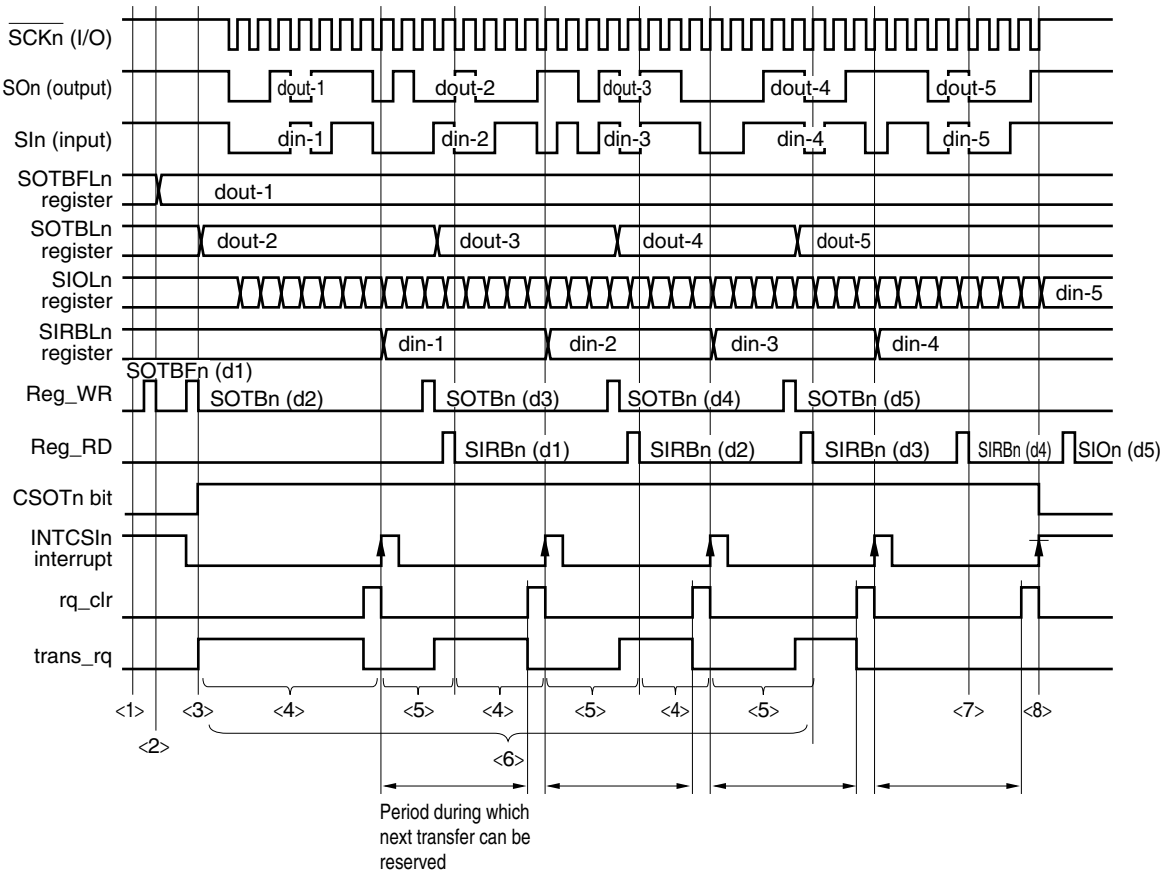
In the case of the repeat transfer mode, two transfer requests are set at the start of the first transfer. Following the transmission/reception completion interrupt request (INTCSIn), transfer is continued if the SIRBn register can be read within the next transfer reservation period. If the SIRBn register cannot be read, transfer ends and the SIRBn register does not receive the new value of the SIOn register.

The last data can be obtained by reading the SIOn register following completion of the transfer.

(b) Usage (transmission/reception)

<1> Set the repeat transfer mode (AUTOn bit of CSIMn register = 1) and the transmission/reception mode (TRMDn bit of CSIMn register = 1)
<2> Write the first data to the SOTBFn register.
<3> Write the 2nd data to the SOTBn register (start transfer).
<4> Wait for a transmission/reception completion interrupt request (INTCSIn).
<5> When the transmission/reception completion interrupt request (INTCSIn) has been set (1), write the next data to the SOTBn register (reserve next transfer), and read the SIRBn register to load the receive data.
<6> Repeat steps <4> and <5> as long as data to be sent remains.
<7> Wait for the INTCSIn interrupt. When the interrupt request signal is set (1), read the SIRBn register to load the (N − 1)th receive data (N: Number of transfer data).
<8> Following the last transmission/reception completion interrupt request (INTCSIn), read the SIOn register to load the Nth (last) receive data.

Figure 11-17. Repeat Transfer (Transmission/Reception) Timing Chart



Remarks 1. n = 5, 6

2. Reg_WR: Internal signal. This signal indicates that the transmit data buffer register (SOTBn/SOTBLn) has been written.

Reg_RD: Internal signal. This signal indicates that the receive data buffer register (SIRBn/SIRBLn) has been read.

rq_clr: Internal signal. Transfer request clear signal.

trans_rq: Internal signal. Transfer request signal.

In the case of the repeat transfer mode, two transfer requests are set at the start of the first transfer. Following the transmission/reception completion interrupt request (INTCSIn), transfer is continued if the SOTBn register can be written within the next transfer reservation period. If the SOTBn register cannot be written, transfer ends and the SIRBn register does not receive the new value of the SIOn register.

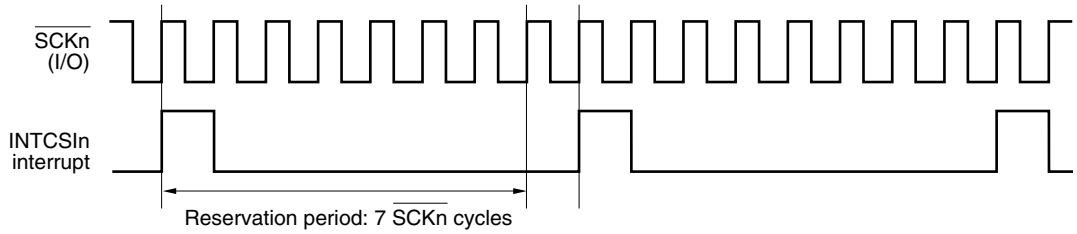
The last receive data can be obtained by reading the SIOn register following completion of the transfer.

(c) Next transfer reservation period

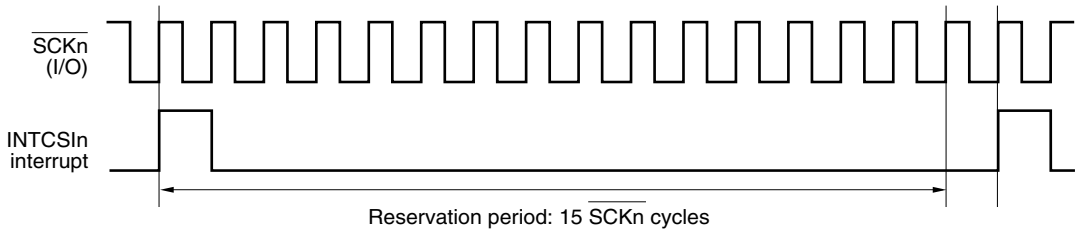
In the repeat transfer mode, the next transfer must be prepared within the period shown below.

Figure 11-18. Timing Chart of Next Transfer Reservation Period (1/2)

(a) When data length is 8 bits, operation mode: CKPn bit = 0, DAPn bit = 0



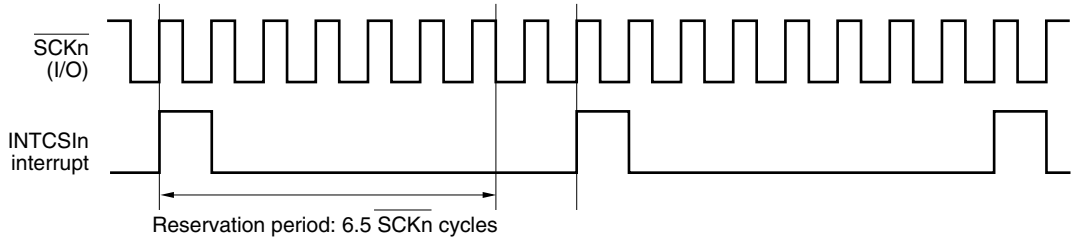
(b) When data length is 16 bits, operation mode: CKPn bit = 0, DAPn bit = 0



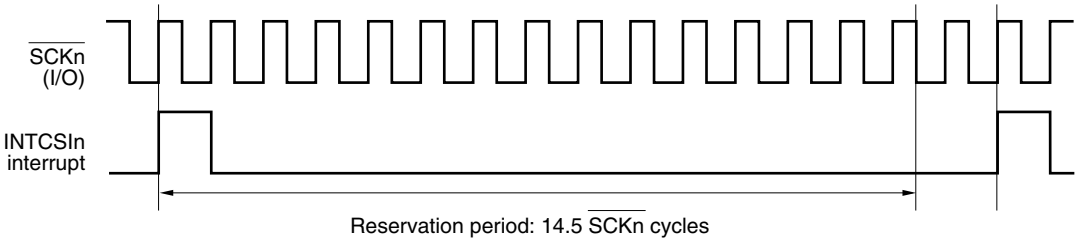
Remark n = 5, 6

Figure 11-18. Timing Chart of Next Transfer Reservation Period (2/2)

(c) When data length is 8 bits, operation mode: CKPn bit = 0, DAPn bit = 1



(d) When data length is 16 bits, operation mode: CKPn bit = 0, DAPn bit = 1



Remark n = 5, 6

(d) Cautions

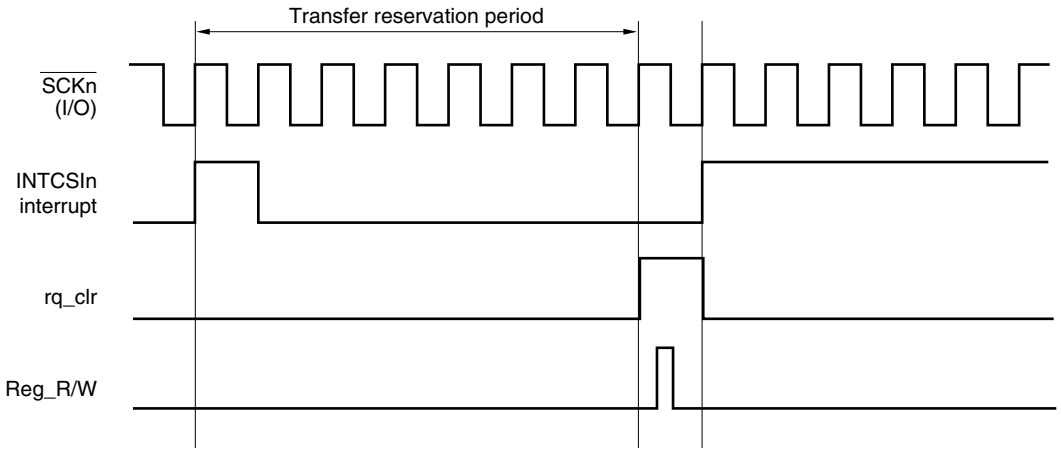
To continue repeat transfers, it is necessary to either read the SIRBn register or write to the SOTBn register during the transfer reservation period.

If the SIRBn register or the SOTBn register is accessed when the transfer reservation period is over, the following occurs.

(i) In case of contention between transfer request clear and register access

Since request cancellation has higher priority, the next transfer request is ignored. Therefore, transfer is interrupted, and normal data transfer cannot be performed.

Figure 11-19. Transfer Request Clear and Register Access Contention



Remarks 1. n = 5, 6

2. rq_clr: Internal signal. Transfer request clear signal.

Reg_R/W: Internal signal. This signal indicates that the receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

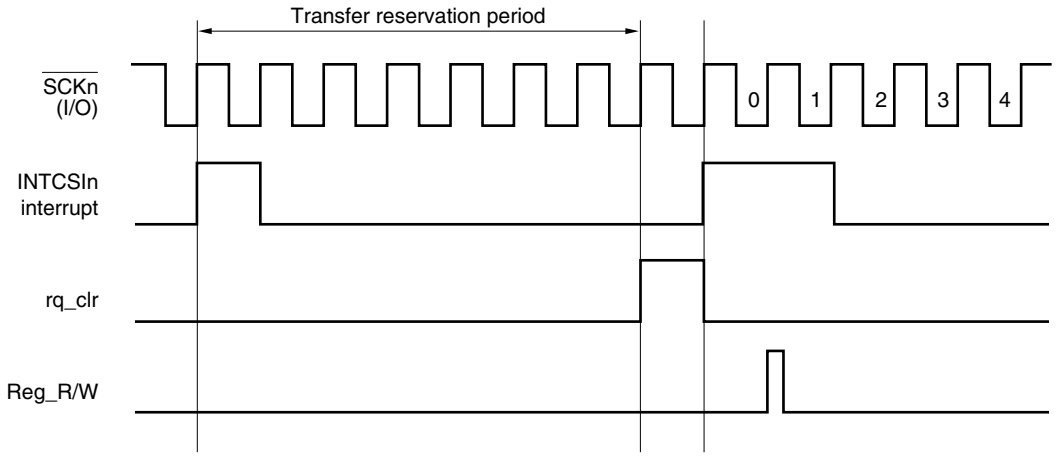
(ii) In case of contention between interrupt request and register access

Since continuous transfer has stopped once, the transfer is executed as a new repeat transfer.

In the slave mode, a bit phase error transfer error results (refer to **Figure 11-20**).

In the transmit/receive mode, the value of the SOTBFn register is retransmitted, and illegal data is sent.

Figure 11-20. Interrupt Request and Register Access Contention



Remarks 1. n = 5, 6

2. rq_clr: Internal signal. Transfer request clear signal.

Reg_R/W: Internal signal. This signal indicates that receive data buffer register (SIRBn/SIRBLn) read or transmit data buffer register (SOTBn/SOTBLn) write was performed.

11.4.5 Output pins

(1) $\overline{\text{SCKn}}$ pin

When CSIn operation is disabled (CSIEn bit of CSIMn register = 0), the $\overline{\text{SCKn}}$ pin output status is as follows (n = 5, 6).

Table 11-4. $\overline{\text{SCKn}}$ Pin Output Status

CKPn	CKSn2	CKSn1	CKSn0	$\overline{\text{SCKn}}$ Pin Output
0	don't care	don't care	don't care	Fixed to high level
1	1	1	1	Fixed to high level
	Other than above			Fixed to low level

Remarks 1. n = 5, 6

2. When any of bits CKPn and CKSn2 to CKSn0 of the CSICKn register is overwritten, the $\overline{\text{SCKn}}$ pin output changes.

(2) SOn pin

When CSIn operation is disabled (CSIEn bit of CSIMn register = 0), the SOn pin output status is as follows (n = 5, 6).

Table 11-5. SOn Pin Output Status

TRMDn	DAPn	AUTOn	CCLn	DIRn	SOn Pin Output
0	don't care	don't care	don't care	don't care	Fixed at low level
1	0	don't care	don't care	don't care	SO latch value (low level)
	1	0	0	0	SOTB7 value
				1	SOTB0 value
			1	0	SOTB15 value
				1	SOTB0 value
		1	0	0	SOTBF7 value
				1	SOTBF0 value
			1	0	SOTBF15 value
				1	SOTBF0 value

Remarks 1. n = 5, 6

2. When any of bits TRMDn, CCLn, DIRn, and AUTOn of the CSIMn register or DAPn bit of the CSICKn register is overwritten, the SOn pin output changes.

3. SOTBm: Bit m of SOTBn register (m = 0, 7, 15)

4. SOTBFm: Bit m of SOTBFn register (m = 0, 7, 15)

11.5 I^2C Bus

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.

I^2C0 and I^2C1 have the following two modes.

- Operation stopped mode
- I^2C (Inter IC) bus mode (multimasters supported)

(1) Operation stopped mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

(2) I^2C bus mode (multimasters support)

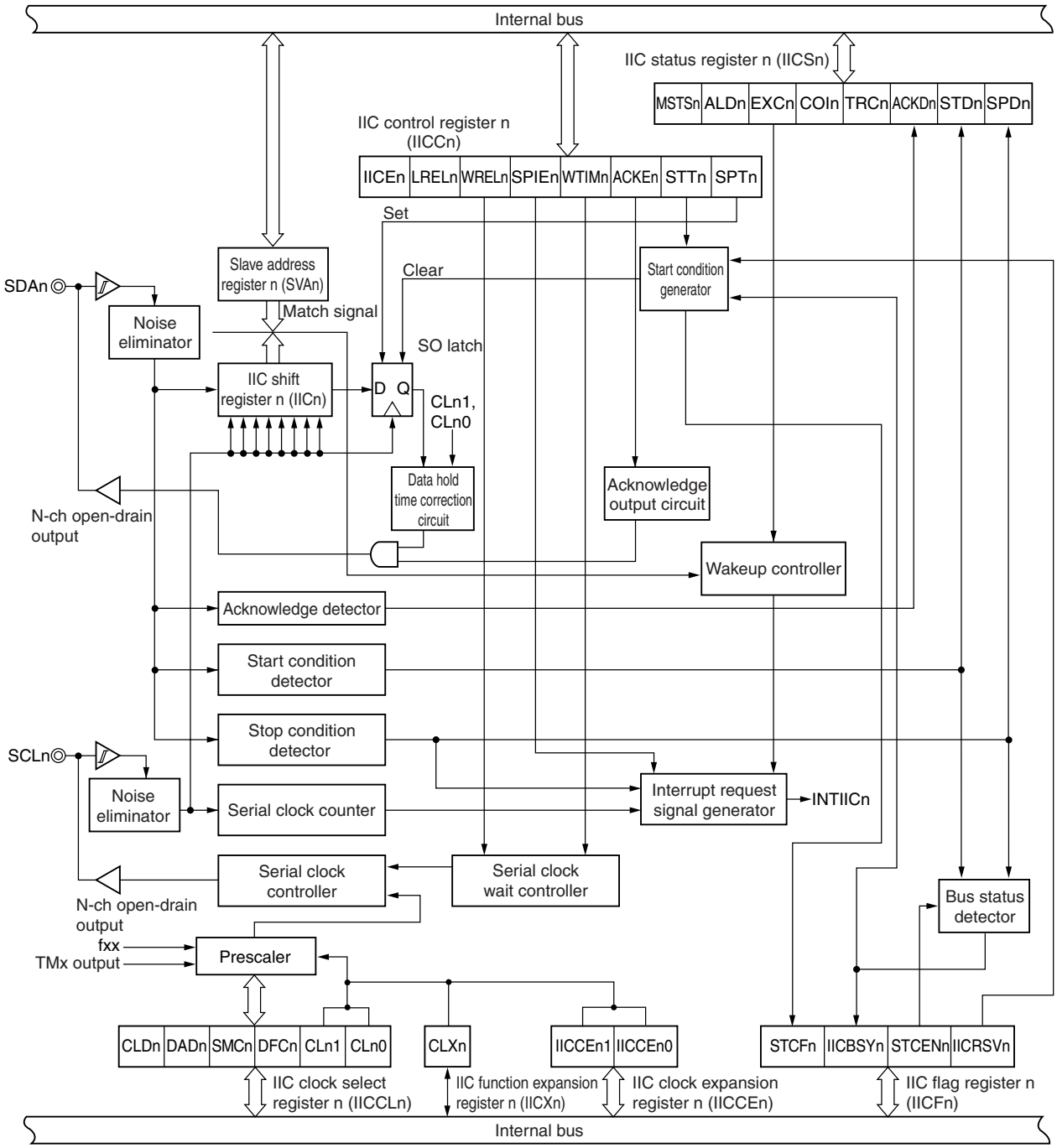
This mode is used for 8-bit data transfers with several devices via two lines: a serial clock line (SCLn) and a serial data bus line (SDAn).

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, I^2Cn requires pull-up resistors for the serial clock line and the serial data bus line.

Remark n = 0, 1

★ **Figure 11-21. Block Diagram of I²Cn**



Remarks 1. n = 0, 1

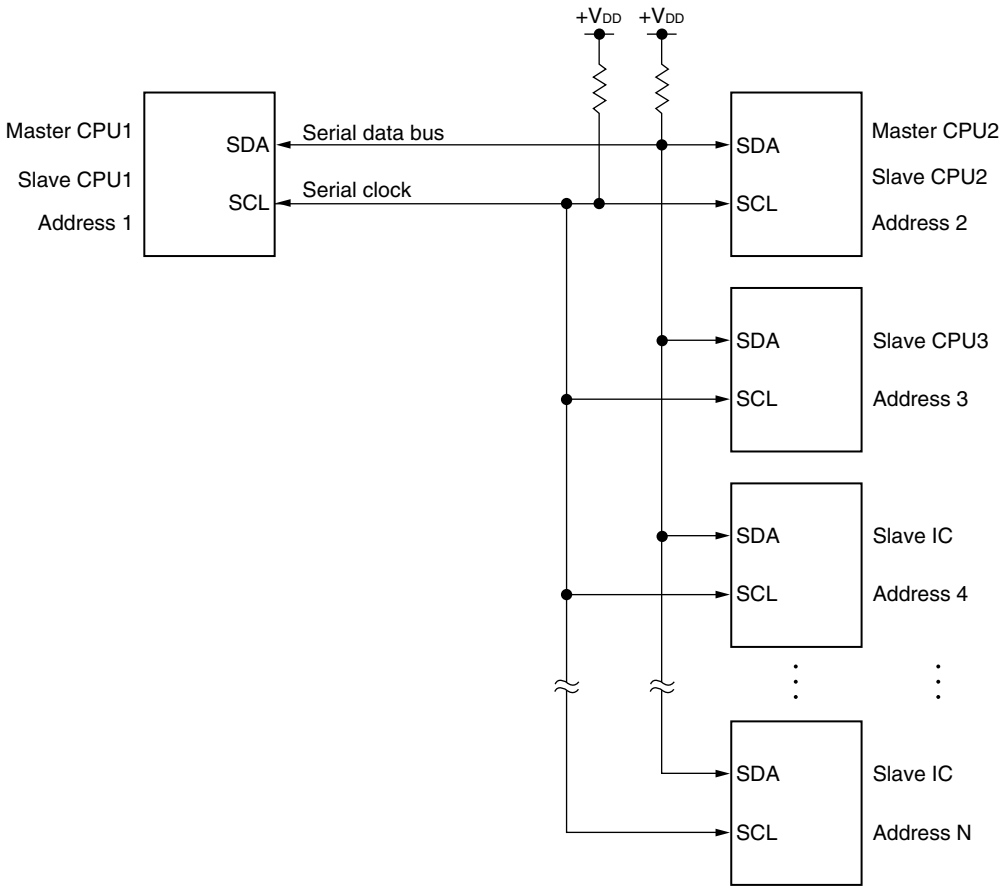
2. TMx output

n = 0: TM5 output

n = 1: TM6 output

A serial bus configuration example is shown below.

Figure 11-22. Serial Bus Configuration Example Using I^2C Bus



11.5.1 Configuration

I^2Cn includes the following hardware (n = 0, 1).

Table 11-6. 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 expansion registers 0 and 1 (IICCE0, IICCE1) IIC function expansion registers 0 and 1 (IICX0, IICX1) IIC clock select registers 0 and 1 (IICCL0, IICCL1)

★

(1) IIC shift registers 0 and 1 (IIC0, IIC1)

These registers are used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data, and 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 sets the IIC0 and IIC1 to 00H.

(2) Slave address registers 0 and 1 (SVA0, SVA1)

This register sets local addresses when in slave mode.

SVAn is set by an 8-bit memory manipulation instruction (n = 0, 1).

$\overline{\text{RESET}}$ input sets the SVA0 and SVA1 to 00H.

(3) SO latch

The SO latch is used to retain the output level of the SDAn pin (n = 0, 1).

(4) Wake-up 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 the IIC control register n (IICCn)
SPIEn bit: bit 4 of the 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**

A start condition is issued when the STTn bit is set.

However, in the communication reservation disabled status (IICRSVn bit = 1), this request is ignored and the STCFn bit is set if the bus is not released (IICBSYn bit = 1).

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**

Whether the bus is released or not is ascertained by detecting a start condition and stop condition.

However, the bus status cannot be detected immediately after operation, so set the initial status by using the STCENn bit.

Remark STCENn bit: Bit 1 of IIC flag register n (IICFn)

11.5.2 I^2C control registers

I^2C0 and I^2C1 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 expansion registers 0, 1 (IICCE0, IICCE1)
- IIC function expansion registers 0, 1 (IICX0, IICX1)
- IIC clock select registers 0, 1 (IICCL0, IICCL1)

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)

These registers are used to enable/disable I^2C operations, set the wait timing, and set other I^2C operations.

IICCn can be set by an 8-bit or 1-bit memory manipulation instruction (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICCn to 00H.

Caution In I^2C0, I^2C1 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^2Cn operation enable/disable specification
0	Operation stopped. IIC status register n (IICSn) preset. Internal operation stopped.
1	Operation enabled.

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 communication operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLn and SDAn lines are set for 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 communication entry conditions are met.
- After a stop condition is detected, restart is in master mode.
- An address match or extension code reception occurs after the start condition.

Condition for clearing (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	Wait not cancelled
1	Wait cancelled. 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	Disabled
1	Enabled

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. 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)

ACKEn	Acknowledge control
0	Acknowledgement disabled.
1	Acknowledgement enabled. During the ninth clock period, the SDA line is set to low level. However, the $\overline{ACK}$ 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 $\overline{RESET}$ is input	• Set by instruction

STTn	Start condition trigger
0	Start condition is not generated.
1	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: If the communication reservation function is enabled (IICRSVn = 0) • This trigger functions as a start condition reserve flag. When set, it releases the bus and then automatically generates a start condition. If the communication reservation function is disabled (IICRSVn = 1) • The STCFn bit is set. This trigger does not generate a start condition. In the wait state (when master device) Generates a restart condition after releasing the wait.

Cautions concerning set timing

- For master reception: Cannot be set during transfer. Can be set only when 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: Even when the communication reservation function is disabled (IICRSVn = 1), the communication reservation status is entered.
- Cannot be set at the same time as SPTn

Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by loss in arbitration • Cleared after start condition is generated by master device • When LRELn = 1 • When $\overline{IICEn}$ = 0 • When $\overline{RESET}$ is input	• Set by instruction

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)

SPTn	Stop condition trigger
0	Stop condition is not generated.
1	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.

Cautions concerning set 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.

Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When LRELn = 1 • When IICEn = 0 • When RESET is input	• Set by instruction

★ **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 the operation enabled status. For details, see **11.5.13 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)

These registers indicate 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 $\overline{\text{IICEn}}$ 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 $\overline{\text{IICEn}}$ 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 RESET is input	• When the high-order 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 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 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 becomes high impedance when bit 5 (WRELn) of IIC control register n (IICCn) is set and the wait state is cancelled at the ninth clock by 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 ACK
0	ACK was not detected.
1	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 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 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 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)**

These registers set the I^2Cn operation mode and indicate the I^2C bus status (n = 0, 1).

IICFn can be set by an 8-bit or 1-bit memory manipulation instruction.

IICRSVn enables/disables the communication reservation function (see **11.5.12 Communication reservation**).

The initial value of the IICBSYn bit is set by using the STCENn bit (see **11.5.13 Cautions**).

The IICRSVn and STCENn bits can be written only when operation of I^2Cn is disabled (IICEn of IIC control register n (IICCn) = 0). After operation is enabled (IICEn = 1), IICFn can be read.

(1/2)

After reset: 00H R/W[Note] Address: FFFFF368H, FFFFF36AH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

(n = 0, 1)

STCFn	STTn clear flag
0	Start condition issued
1	STTn flag cleared

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Cleared by STTn = 1 • When RESET is input	• Clearance of STTn when communication reservation disabled (IICRSVn = 1)

IICBSYn	I^2Cn bus status flag
0	Bus released status
1	Bus communication status

Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
• When stop condition is detected • When RESET is input	• When start condition is detected • By setting 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	Start conditions cannot be generated until a stop condition is detected following operation enable (IICEn = 1).
1	Start conditions can be generated even if a stop condition is not detected following operation enable (IICEn = 1).

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• When start condition is detected • When RESET is input	• Setting by instruction

IICRSVn	Communication reservation function disable bit
0	Communication reservation enabled
1	Communication reservation disabled

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Clearing by instruction • When RESET is input	• Setting by instruction

Cautions 1. Write STCENn and IICRSVn only when operation is stopped (IICEn = 0).

2. When STCENn = 1, the bus released status (IICBSYn = 0) is recognized regardless of the actual bus status immediately after the I^2Cn bus operation is enabled. Therefore, to issue the first start condition (STTn = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

Remarks 1. STTn: Bit 1 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

2. n = 0, 1

(4) IIC clock expansion registers 0, 1 (IICCE0, IICCE1), IIC function expansion registers 0, 1 (IICX0, IICX1), IIC clock select registers 0, 1 (IICCL0, IICCL1)

These registers are used to set the transfer clock for the I^2Cn bus.

IICCEn can be set by an 8-bit memory manipulation instruction, and IICXn and IICCLn can be set by an 8-bit or 1-bit memory manipulation instruction (n = 0, 1).

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

(1/2)

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)

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)

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 $\overline{\text{IICEn}}$ = 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 $\overline{\text{IICEn}}$ = 0 • When $\overline{\text{RESET}}$ is input	• When the SDAn line is at high level

Note Bits 4 and 5 of IICCLn are read-only bits.

★ **Caution Always set bits 7 and 6 of IICCLn to 0.**

Remark IICEn: Bit 7 of IIC control register n (IICCn)

(2/2)

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).	

IICCEn1	IICCEn0	CLXn	SMCn	CLn1	CLn0	Selection clock (f_{XX}/m)		Settable main clock frequency (f_{XX}) range	Operation mode
x	x	1	1	0	x	f_{XX}/12		4.0 MHz to 4.19 MHz	High-speed mode
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	
0	0	0	1	1	1	n = 0	TM5 output/18	TM5 setting	
						n = 1	TM6 output/18	TM6 setting	
x	x	0	0	0	0	f_{XX}/44		4.0 MHz to 4.19 MHz	Normal mode
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	
0	0	0	0	1	1	n = 0	TM5 output/66	TM5 setting	
						n = 1	TM6 output/66	TM6 setting	
Other than above						Setting prohibited			

Remarks 1. n = 0, 1

2. x: don't care

3. If the selected clock is specified as a timer output, the P17/T05/TI5 and P30/T06/TI6 pins do not need to be in timer output mode.

(a) 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 the descriptions for bits IICCEn1, IICCEn0, CLXn, SMCn, CLn1, and CLn0 in **11.5.2 (4)**.)

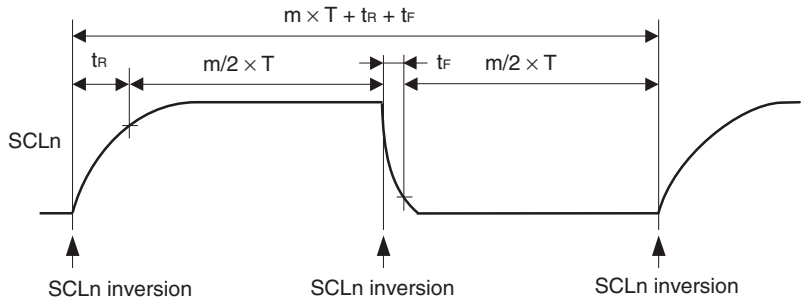
T: $1/f_{XX}$

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.

$$f_{SCL} = 1/(198 \times 50 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 98.5 \text{ kHz}$$



(5) IIC shift registers 0, 1 (IIC0, IIC1)

IICn is used for serial transmission/reception (shift operations) 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)

(6) 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 to 0.

After reset: 00H R/W Address: FFFFF346H, FFFFF356H

	7	6	5	4	3	2	1	0
SVAn								0

(n = 0, 1)

11.5.3 I^2C bus mode functions

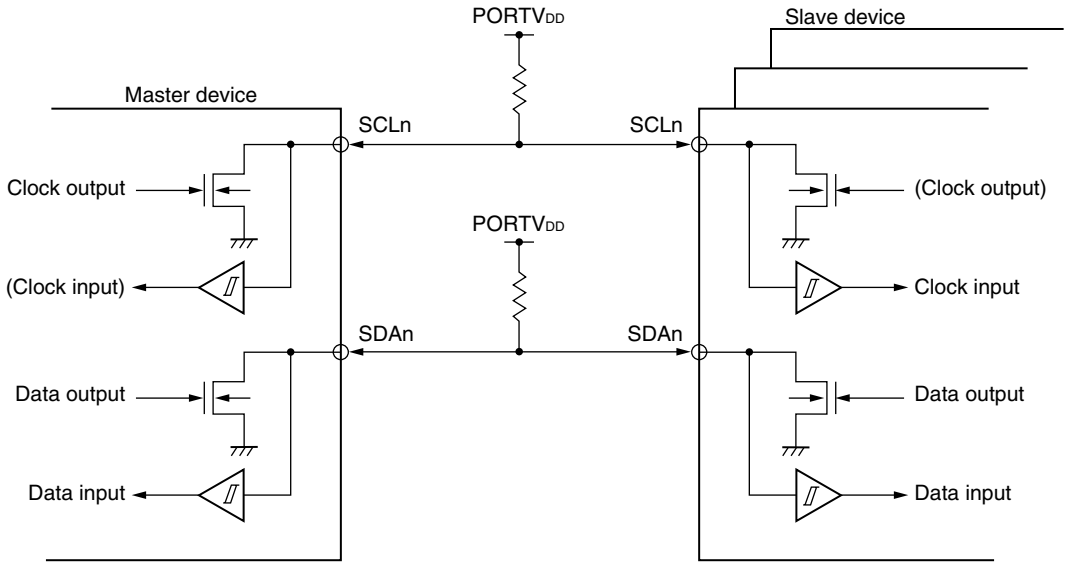
(1) Pin configuration

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.

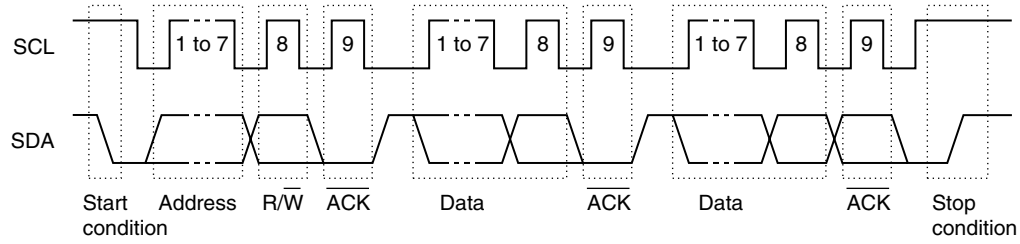
Figure 11-23. Pin Configuration Diagram



11.5.4 I^2C bus definitions and control methods

The following section describes the I^2C bus's serial data communication format and the signals used by the I^2C bus. The transfer timing for the "start condition", "data", and "stop condition" output via the I^2C bus's serial data bus is shown below.

Figure 11-24. I^2C Bus Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

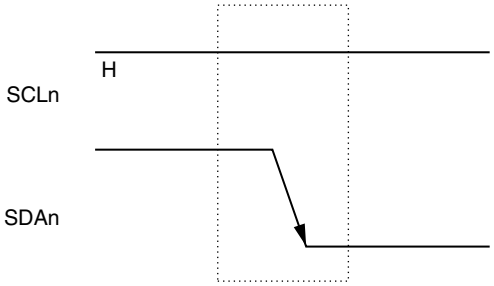
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

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 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 11-25. Start Conditions



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 IIC status register n (IICSn)). When a start condition is detected, IICSn's bit 1 (STDn) is set to 1 (n = 0, 1).

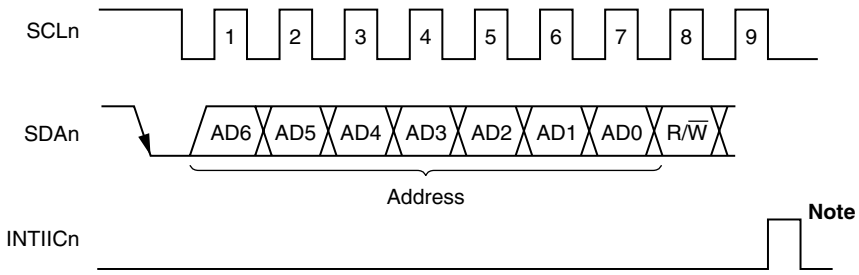
(2) Addresses

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 11-26. Address



Note INTIICn is generated if a local address or extension code is received during slave device operation.

Remark n = 0, 1

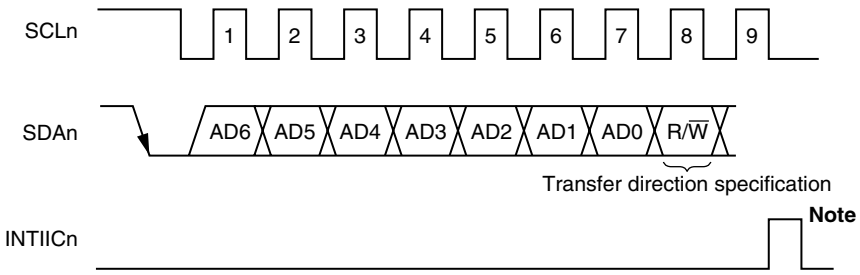
The slave address and the eighth bit, which specifies the transfer direction as described in **(3) Transfer direction specification** below, are written together to the IIC shift register (IICn) and then output. Received addresses are written to IICn (n = 0, 1).

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 11-27. 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 each 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).

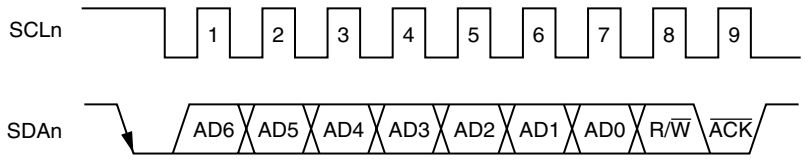
When bit 2 (ACKEn) of IIC control register n (IICCn) is set to 1, automatic $\overline{ACK}$ signal generation is enabled (n = 0, 1).

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 11-28. $\overline{ACK}$ 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.

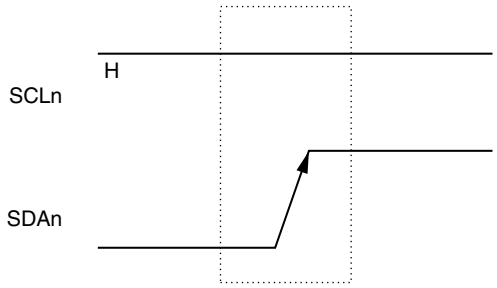
When 9-clock wait is selected: $\overline{ACK}$ signal is automatically output at the falling edge of the SCLn's eighth clock if ACKEn has already been set to 1.

(5) Stop condition

When the SCLn pin is 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 11-29. 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 ($\overline{\text{WAIT}}$)

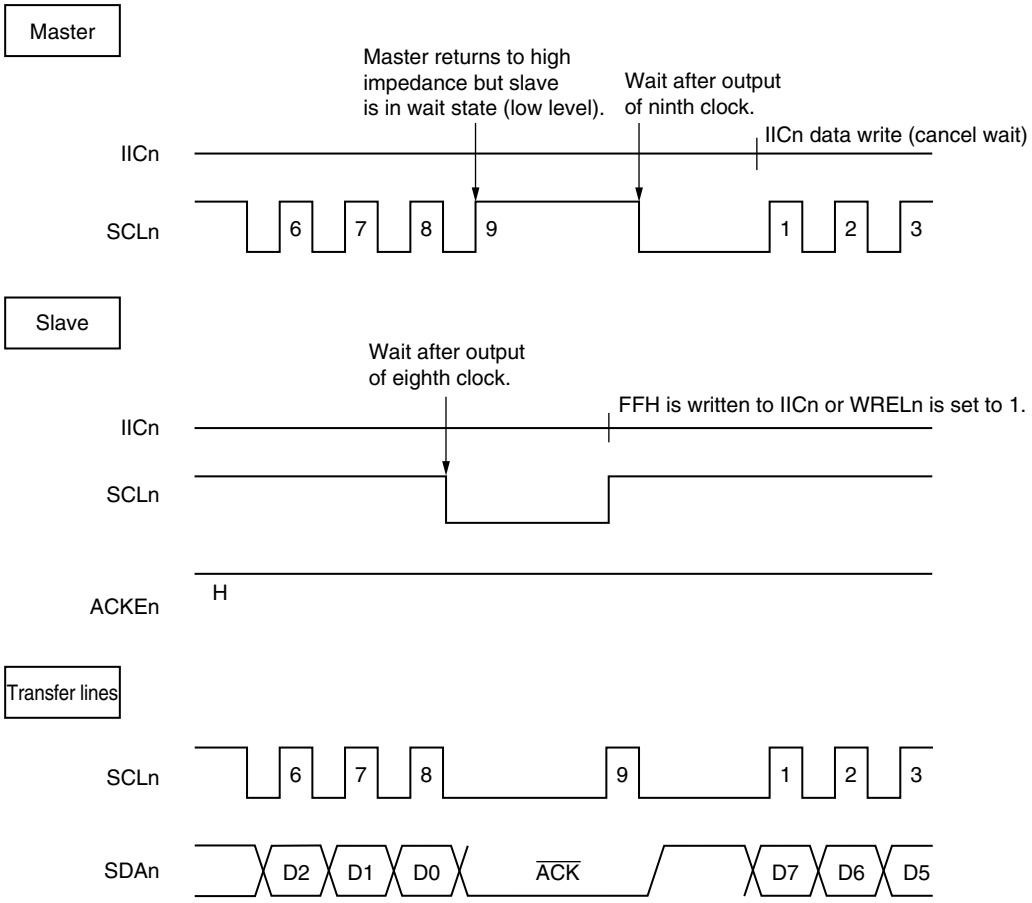
The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

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 11-30. Wait Signal (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait

(master: transmission, slave: reception, and ACKEn = 1)

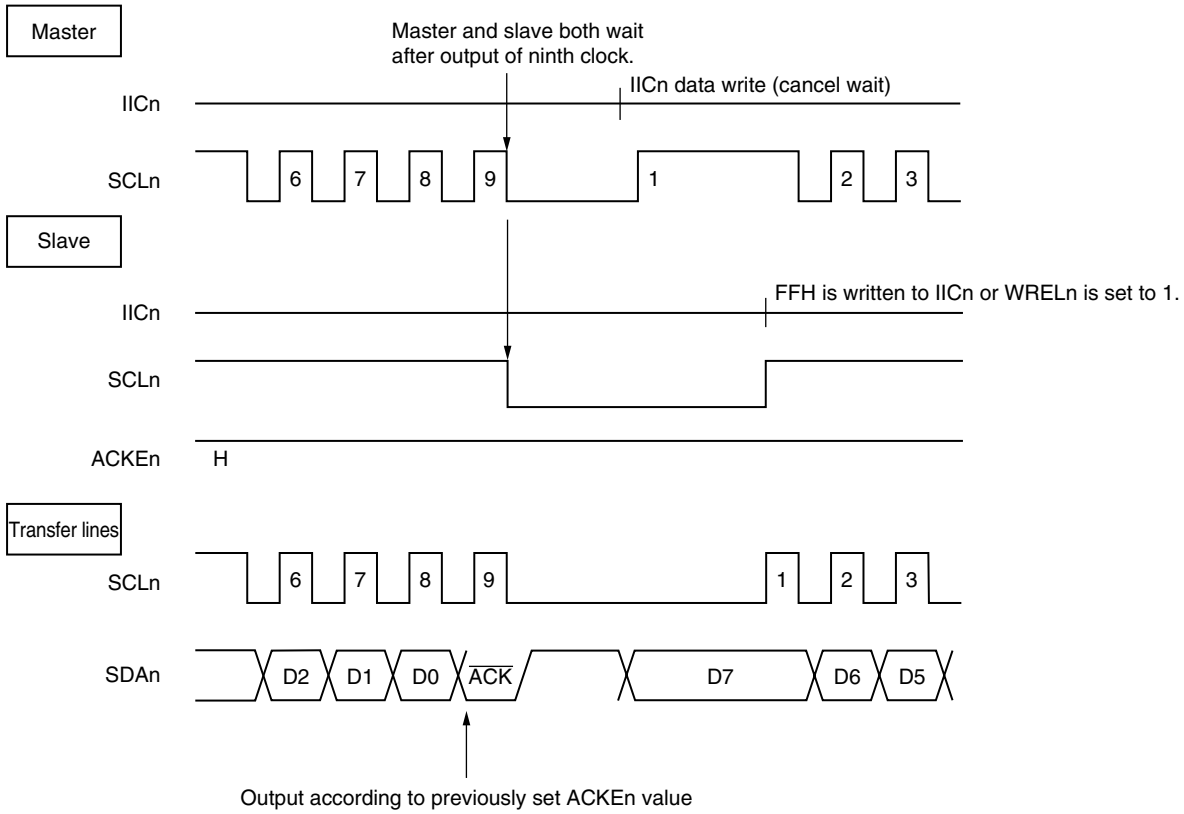


Remark n = 0, 1

Figure 11-30. Wait Signal (2/2)

(2) 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 of 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) on the receiving side, 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

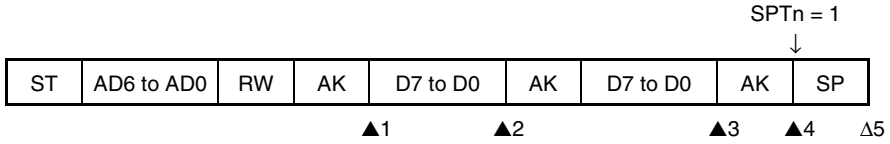
11.5.5 I^2C interrupt requests (INTIICn)

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



▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX000B

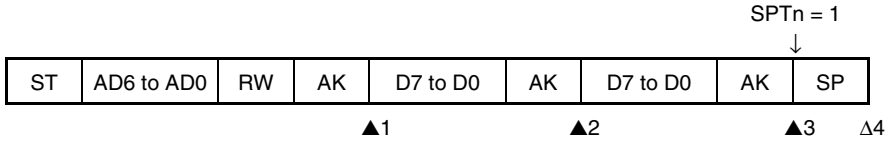
▲3: IICSn = 10XXX000B (WTIMn = 1)

▲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



▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX100B

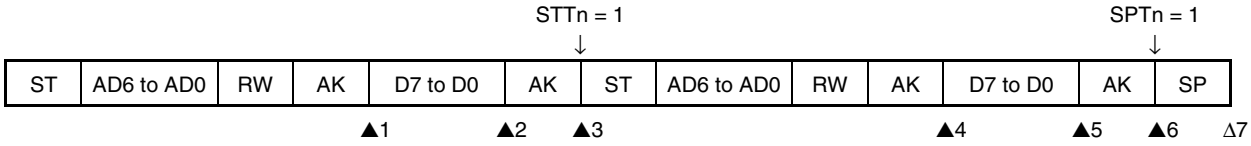
▲3: IICSn = 10XXXX00B

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



▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXX000B (WTIMn = 1)

▲3: IICSn = 10XXXX00B (WTIMn = 0)

▲4: IICSn = 10XXX110B (WTIMn = 0)

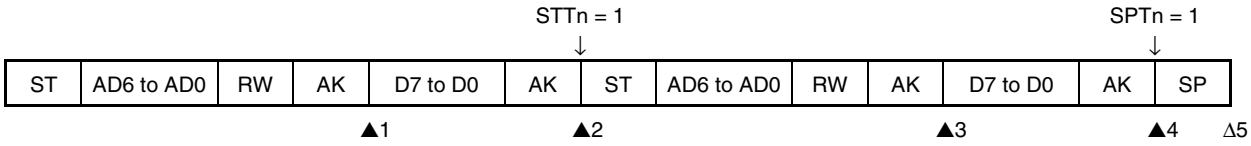
▲5: IICSn = 10XXX000B (WTIMn = 1)

▲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



▲1: IICSn = 10XXX110B

▲2: IICSn = 10XXXX00B

▲3: IICSn = 10XXX110B

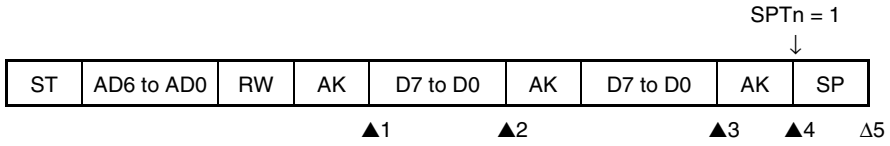
▲4: IICSn = 10XXXX00B

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



▲1: IICSn = 1010X110B

▲2: IICSn = 1010X000B

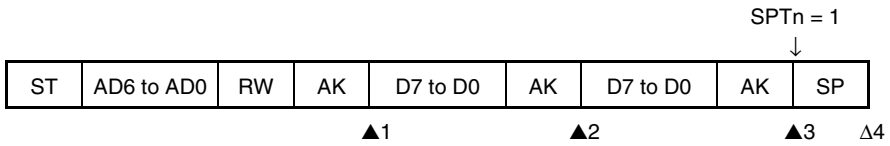
▲3: IICSn = 1010X000B (WTIMn = 1)

▲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



▲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 (matches 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

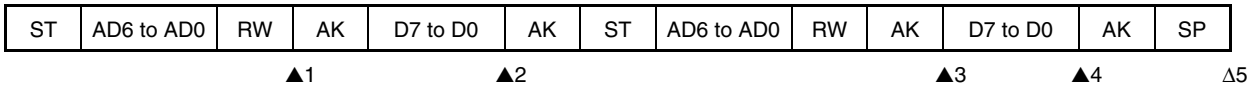
▲4: IICSn = 0010XX00B

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)



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

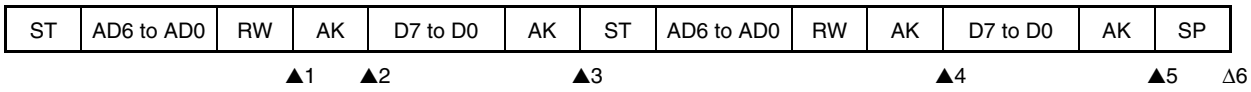
▲3: IICSn = 0001X110B

▲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)



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0001X110B

▲5: IICSn = 0001XX00B

Δ 6: IICSn = 00000001B

Remark ▲: Always generated
Δ: 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

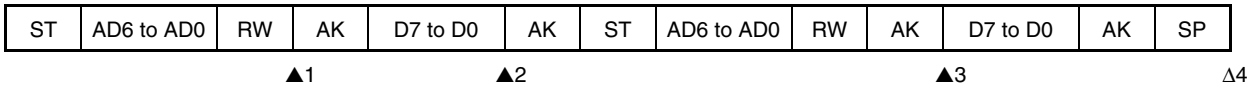
Δ: Generated only when SPIEn = 1

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))



▲1: IICSn = 0010X010B

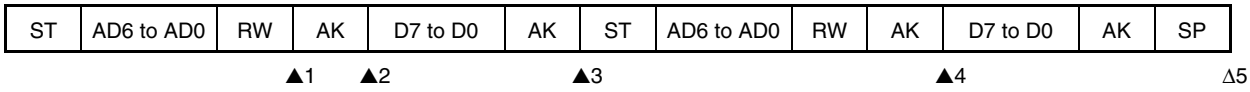
▲2: IICSn = 0010X000B

▲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))



▲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

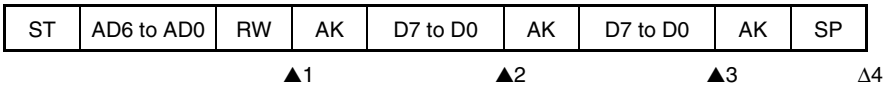
▲3: IICSn = 0001XX00B

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



▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

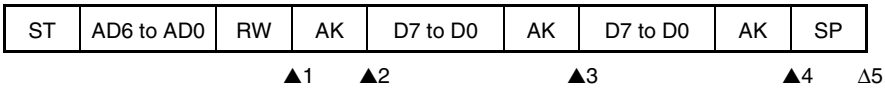
▲2: IICSn = 0010X000B

▲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



▲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)

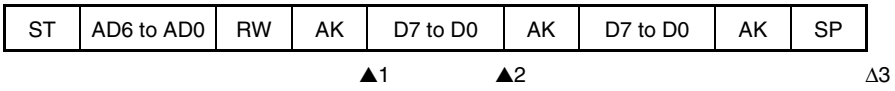
IICCn's LRELn is set to 1 by software

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



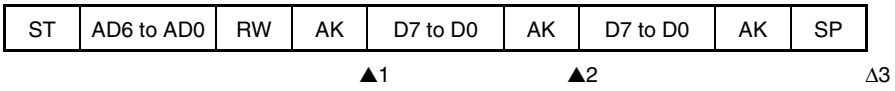
▲1: IICSn = 10001110B

▲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



▲1: IICSn = 10001110B

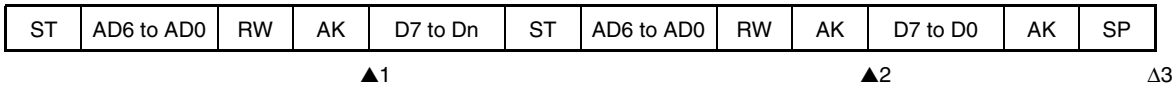
▲2: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
n = 0, 1

(d) When arbitration loss occurs due to restart condition during data transfer

<1> Not extension code (Example: mismatches with SVAn)



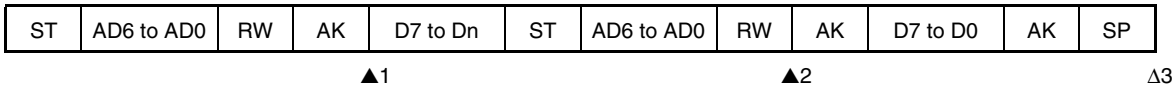
▲1: IICSn = 1000X110B

▲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



▲1: IICSn = 1000X110B

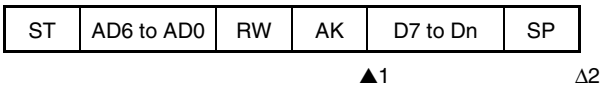
▲2: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

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 arbitration loss occurs due to stop condition during data transfer



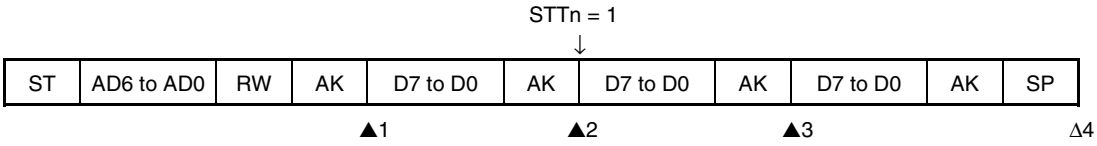
▲1: IICSn = 1000X110B

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



▲1: IICSn = 1000X110B

▲2: IICSn = 1000XX00B

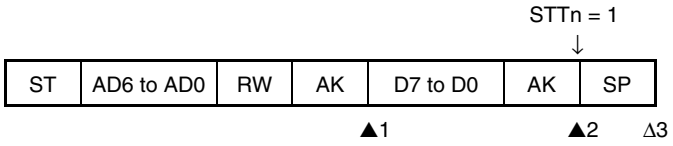
▲3: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

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



▲1: IICSn = 1000X110B

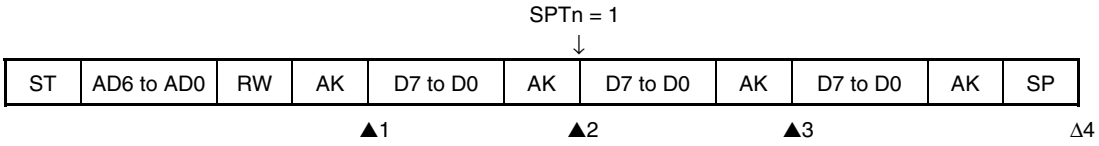
▲2: IICSn = 1000XX00B

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



▲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

11.5.6 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 11-7. 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, $\overline{ACK}$ 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 the 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 the 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 $\overline{ACK}$ must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICn is generated when a stop condition is detected.

Remark n = 0, 1

11.5.7 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).

11.5.8 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).

11.5.9 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 four bits of data match: EXCn = 1[Note]
- Seven 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 11-8. 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

11.5.10 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 while the number of clocks are being adjusted until the data differs. This kind of operation is called arbitration (n = 0, 1).

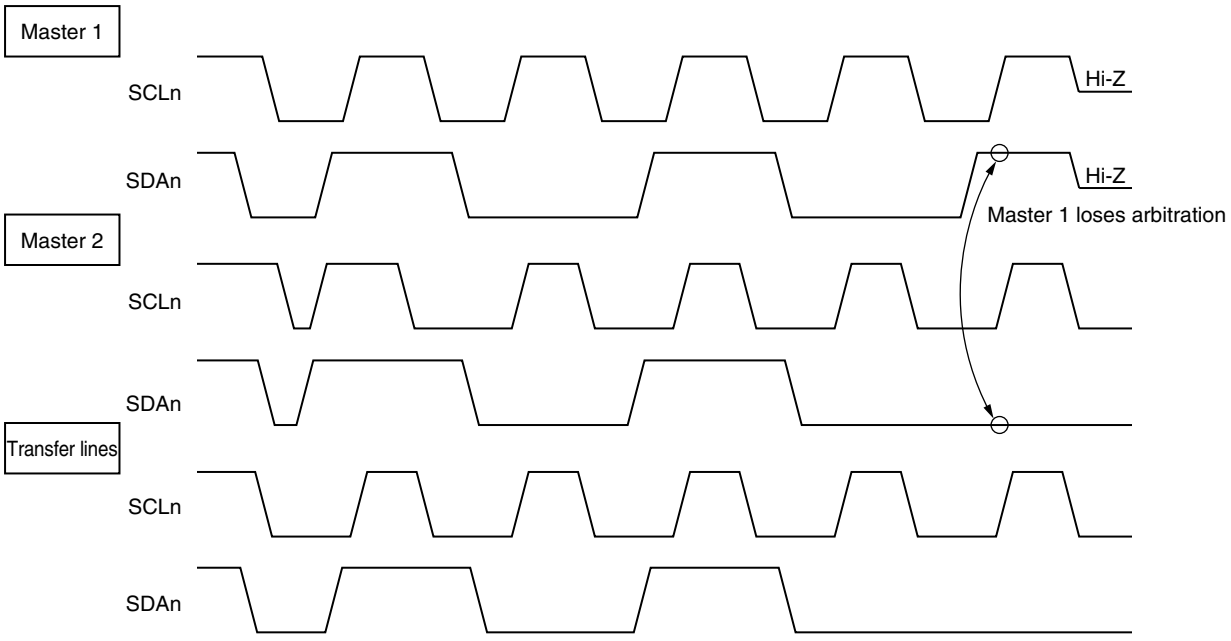
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in IIC status register n (IICSn) is set via the timing by which the arbitration loss occurred, and the SCLn and SDAn lines are both set to high impedance, which releases the bus (n = 0, 1).

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 **11.5.5 I^2C interrupt requests (INTIICn)**.

Note STDn: Bit 1 of IIC status register n (IICSn)
STTn: Bit 1 of IIC control register n (IICCn)

Figure 11-31. Arbitration Timing Example



Remark n = 0, 1

Table 11-9. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{ACK}$ signal transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when 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 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 the IIC control register n (IICCn)

11.5.11 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, wake-up standby mode is set. This wake-up 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 wake up function, and this determines whether interrupt requests are enabled or disabled (n = 0, 1).

11.5.12 Communication reservation

(1) When communication reservation function is enabled (IICRSVn of IICFn = 0)

To start master device communications when not currently using the bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when bit 6 (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 MSTSn (bit 7 of the IIC status register n (IICSn)) (n = 0, 1).
The wait periods, which should be set via software, are listed in Table 11-10. These wait periods can be set by bits 3, 1, and 0 (SMCn, CLn1, and CLn0) of IIC clock select register n (IICCLn) (n = 0, 1).

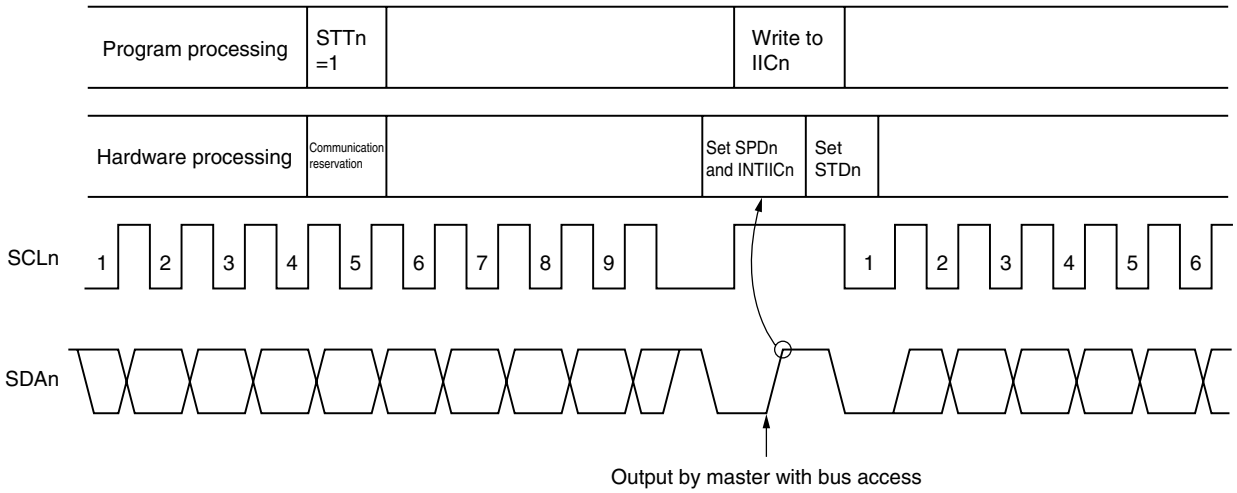
Table 11-10. Wait Periods

SMCn	CLn1	CLn0	Wait Period
0	0	0	26 clocks
0	0	1	46 clocks
0	1	0	92 clocks
0	1	1	37 clocks
1	0	0	16 clocks
1	0	1	
1	1	0	32 clocks
1	1	1	13 clocks

Remark n = 0, 1

The communication reservation timing is shown below.

Figure 11-32. Communication Reservation Timing

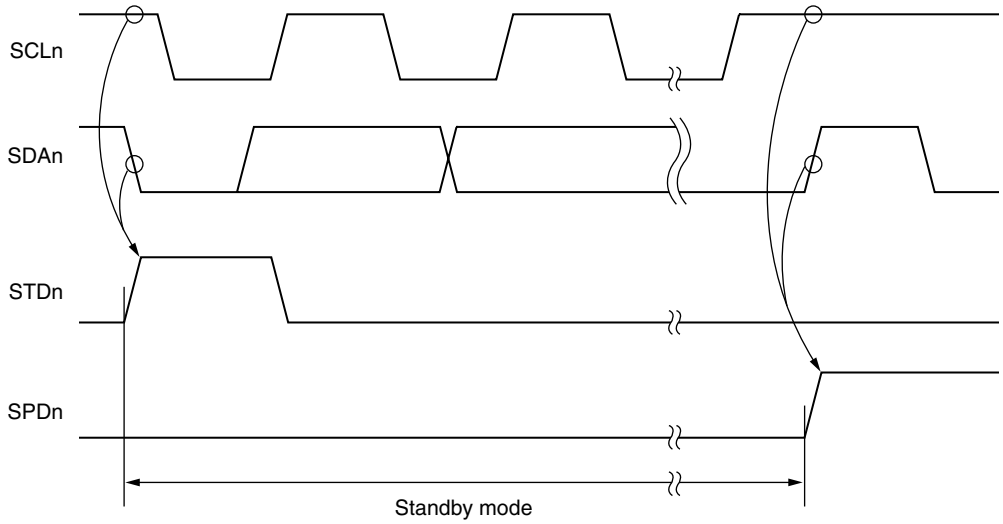


IICn: IIC shift register n
STTn: Bit 1 of IIC control register n (IICCn)
STDn: Bit 1 of IIC status register n (IICSn)
SPDn: Bit 0 of IIC status register n (IICSn)

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).

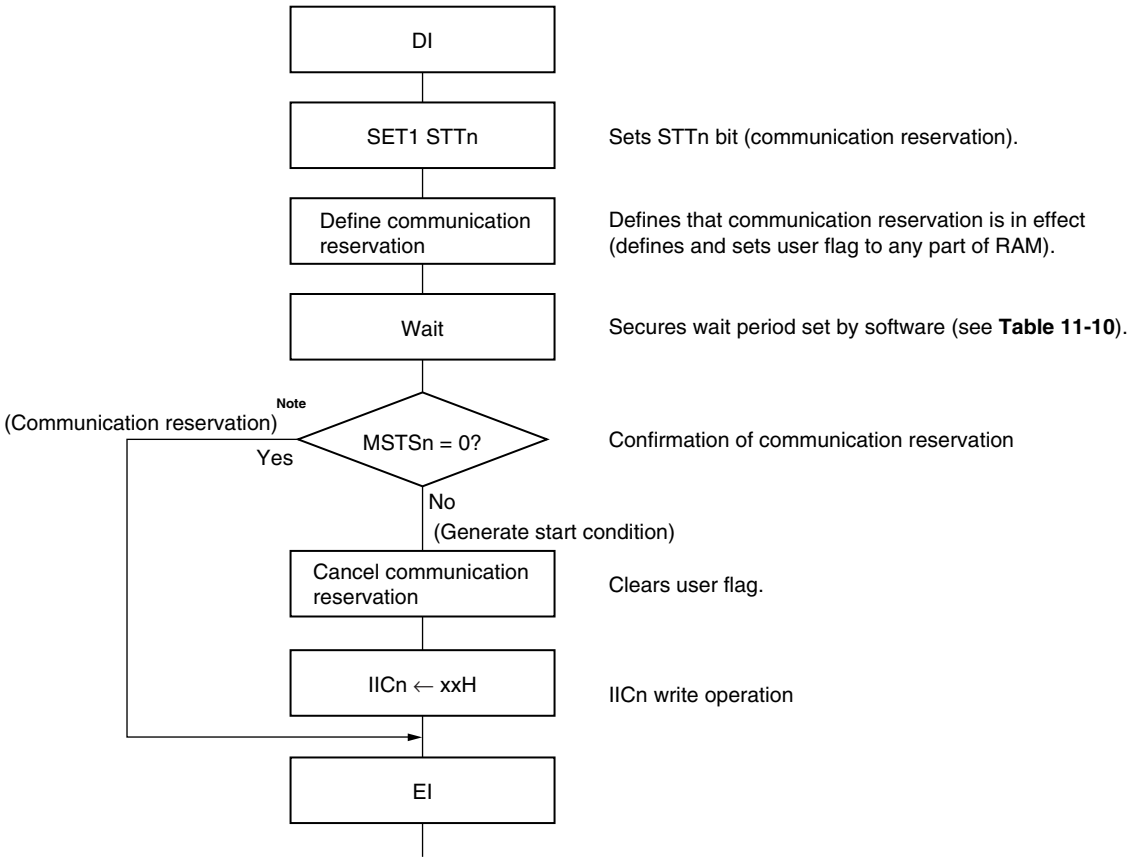
Figure 11-33. Timing for Accepting Communication Reservations



Remark n = 0, 1

The communication reservation flowchart is illustrated below.

Figure 11-34. Communication Reservation Flowchart (1)



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)**

When the STTn bit of the IICCn register is set when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where 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).

To confirm whether the start condition was generated or request was rejected, check the STCFn flag of the IICFn register. The time shown in Table 11-11 is required until the STCFn flag is set after setting STTn = 1. Therefore, secure the time by software.

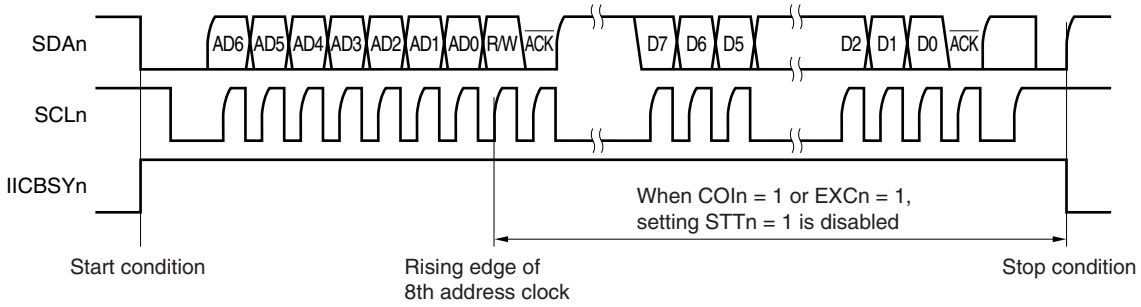
Table 11-11. Wait Time

IICCEn1	IICCEn0	CLn1	CLn0	Wait Time
×	×	0	0	3 clocks
×	×	0	1	3 clocks
×	×	1	0	6 clocks
0	0	1	1	3 × N
0	1	1	1	6 clocks
1	0	1	1	9 clocks

Remarks 1. N: TM5 and TM6 outputs
×: don't care
2. n = 0, 1

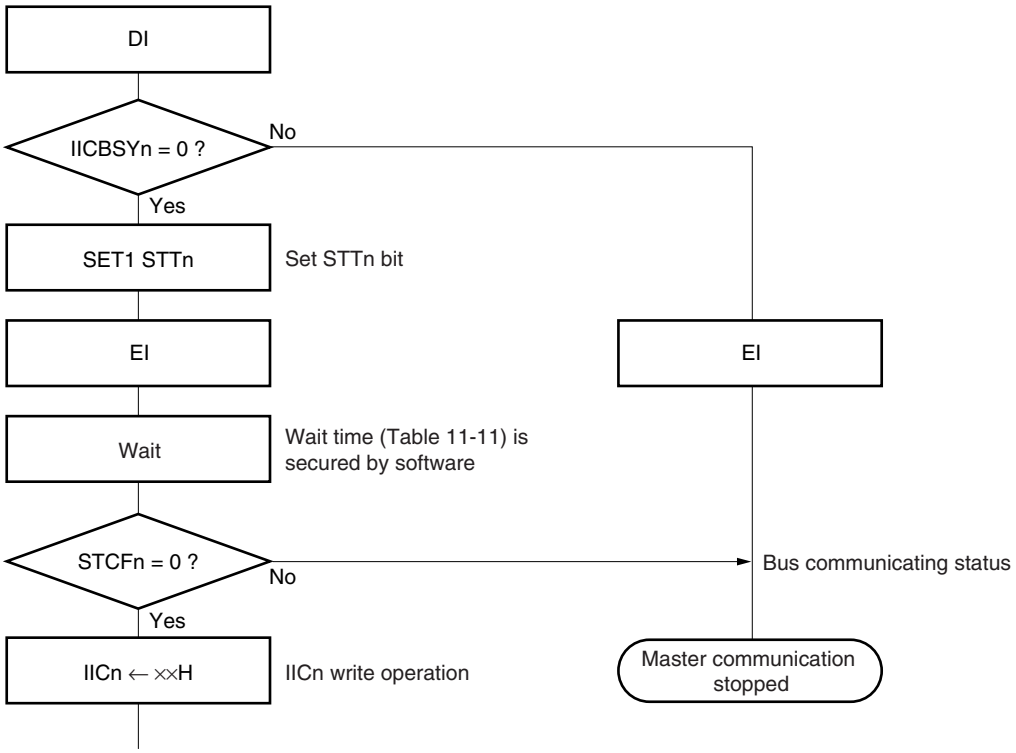
Caution If the slave status is entered by an address match or expansion code reception (timing shown in Figure 11-35), do not set STTn to 1. When set, the communication reservation status is entered.

Figure 11-35. STTn = 1 Setting Disabled Timing



Remark n = 0, 1

Figure 11-36. Communication Reservation Flowchart (2)



Remark n = 0, 1

★ 11.5.13 Cautions

(1) When STCENn of IIC flag register n (IICFn) = 0

Immediately after the I^2Cn operation is enabled, the bus communication status (IICBSYn of IICFn register = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set IIC clock select register n (IICCLn).
<2> Set IICEn of the IIC control register n (IICCn).
<3> Set SPTn of IICCn.

(2) When STCENn of IIC flag register n (IICFn) = 1

Immediately after I^2Cn operation is enabled, the bus released status (IICBSYn of IICFn register = 0) is recognized regardless of the actual bus status. To issue the first start condition (STTn of IICCn register = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

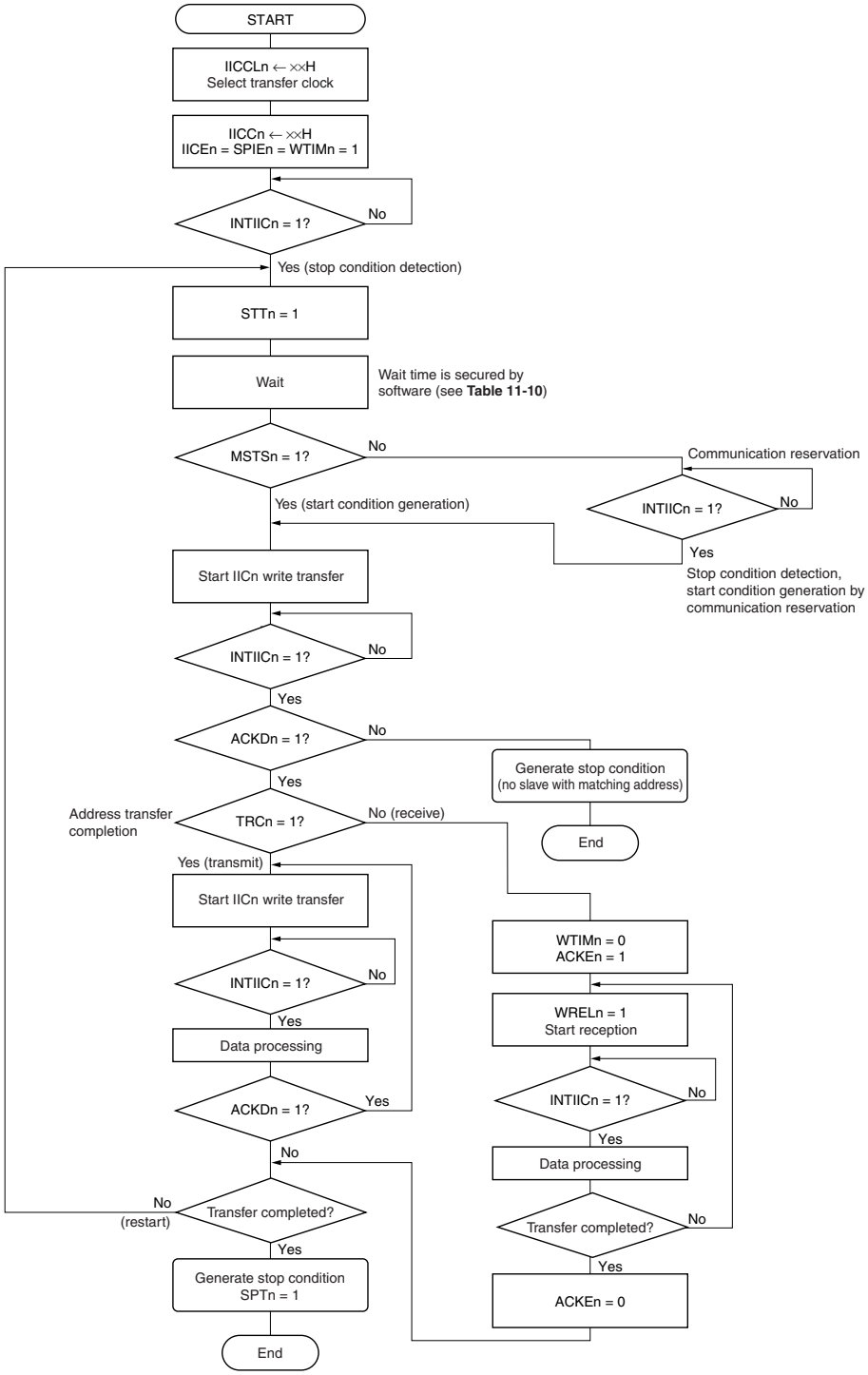
Remark n = 0, 1

11.5.14 Communication operations

★ (1) Master operations (1)

The following shows the flowchart for master communication when the communication reservation function is enabled (IICRSVn = 0) and the master operation is started after a stop condition is detected (STCENn = 0).

Figure 11-37. Master Operation Flowchart (1)

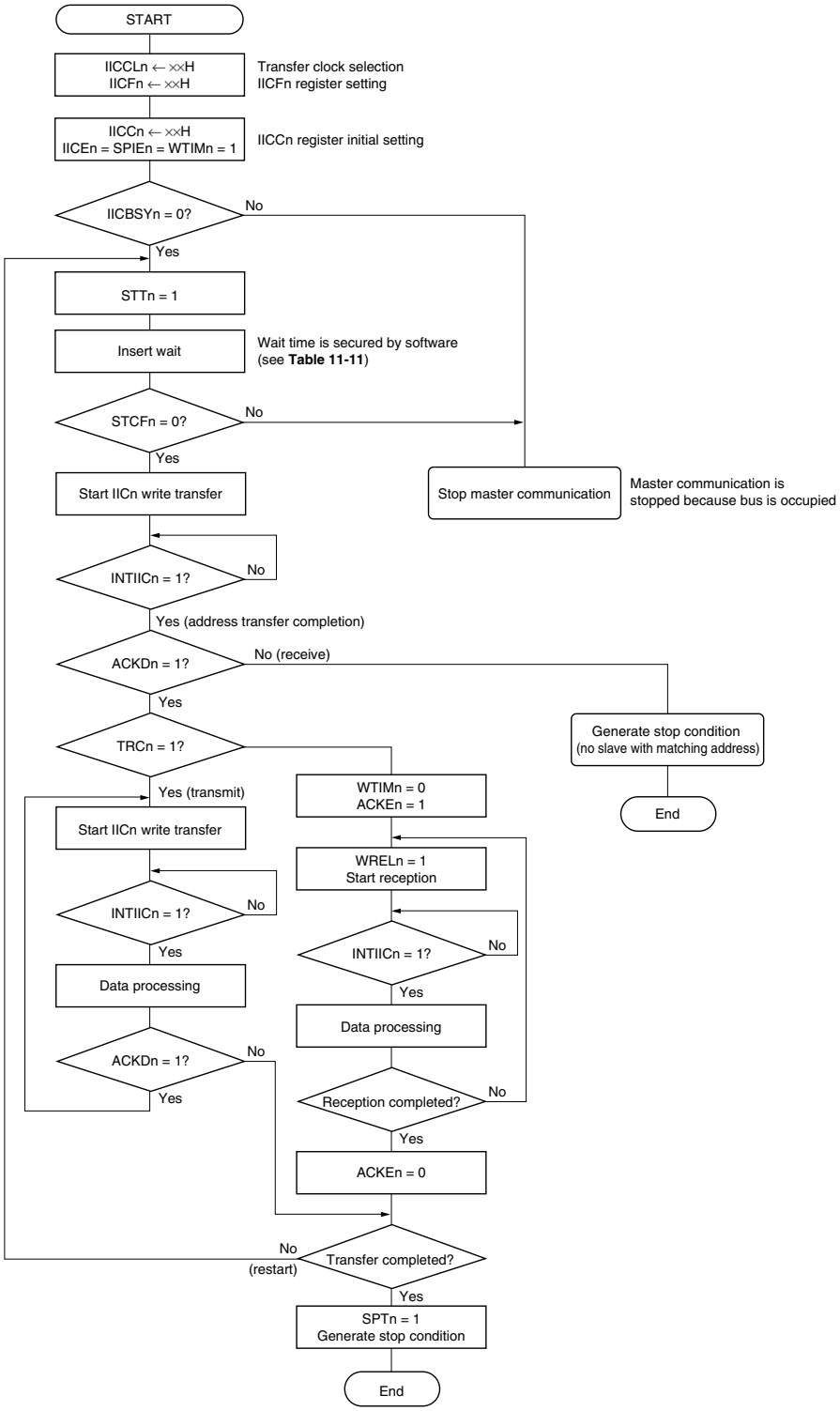


Remark n = 0, 1

★ **(2) Master operations (2)**

The following shows the flowchart for master communication when the communication reservation function is disabled (IICRSVn = 1) and the master operation is started without detecting a stop condition (STCENn = 1).

Figure 11-38. Master Operation Flowchart (2)

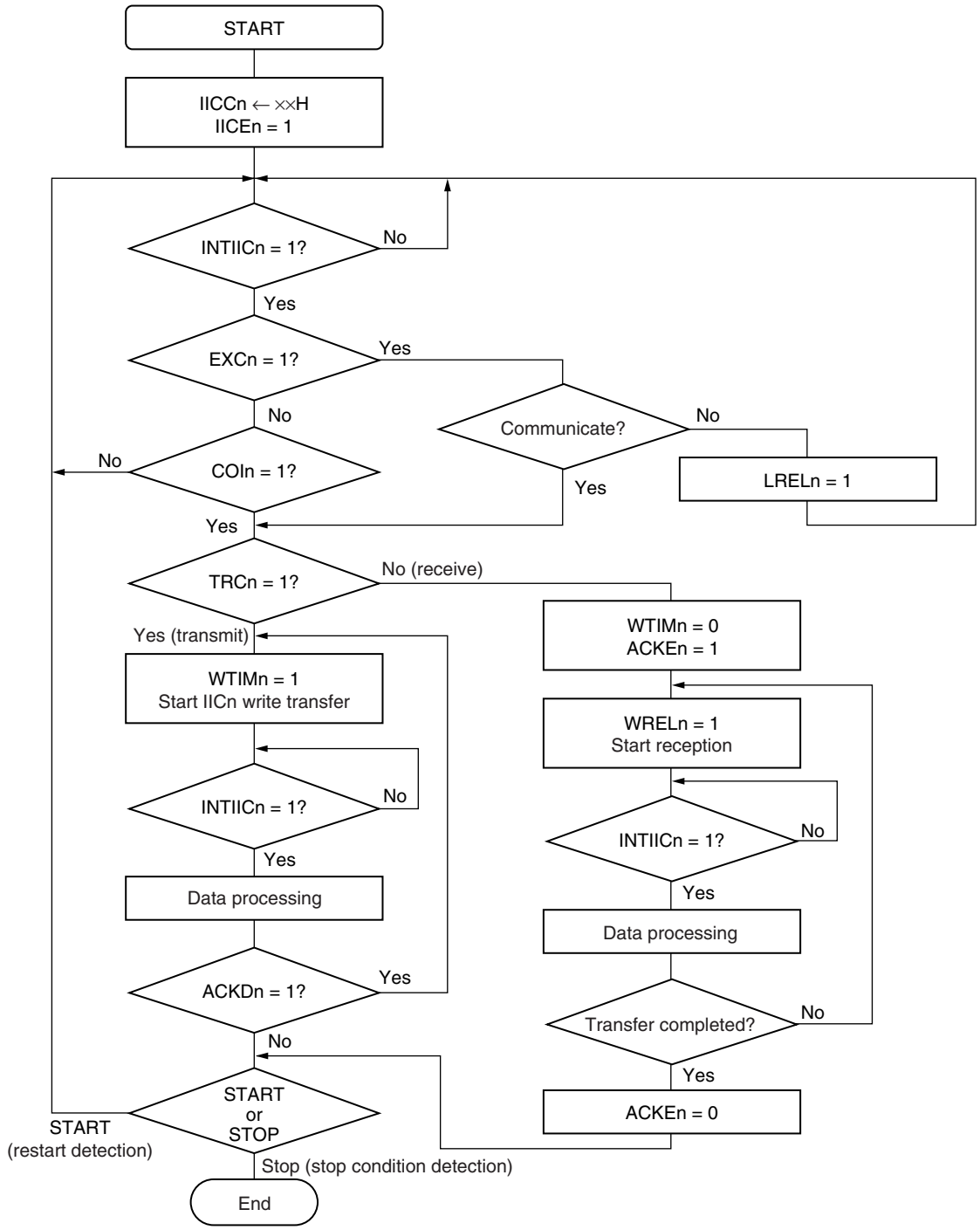


Remark n = 0, 1

(3) Slave operation

The following shows the flowchart for slave communication.

Figure 11-39. Slave Operation Flowchart



Remark n = 0, 1

11.5.15 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)), which 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.

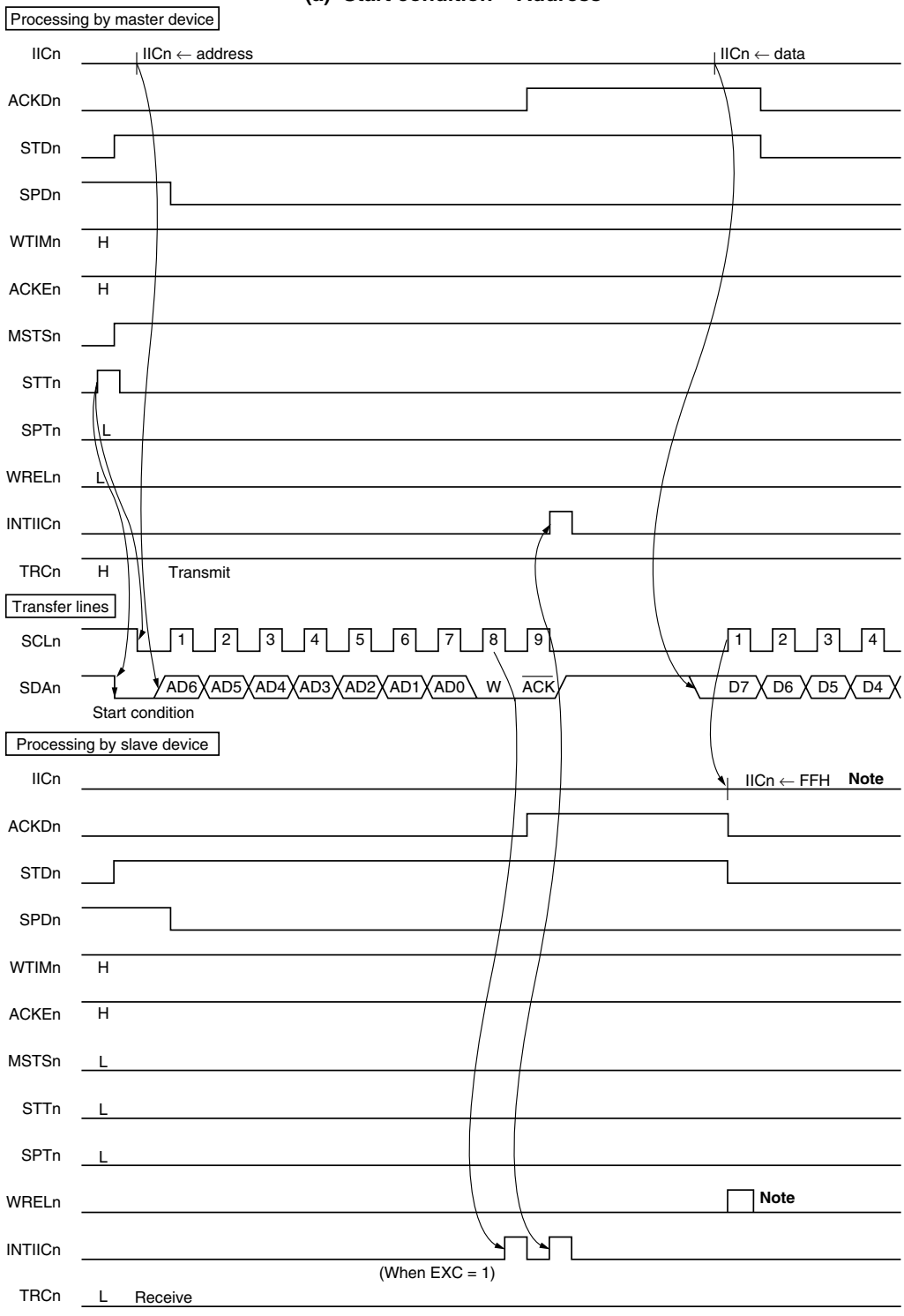
Data input via the SDAn pin is captured by IICn at the rising edge of SCLn.

The data communication timing is shown below.

Remark n = 0, 1

Figure 11-40. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ Address

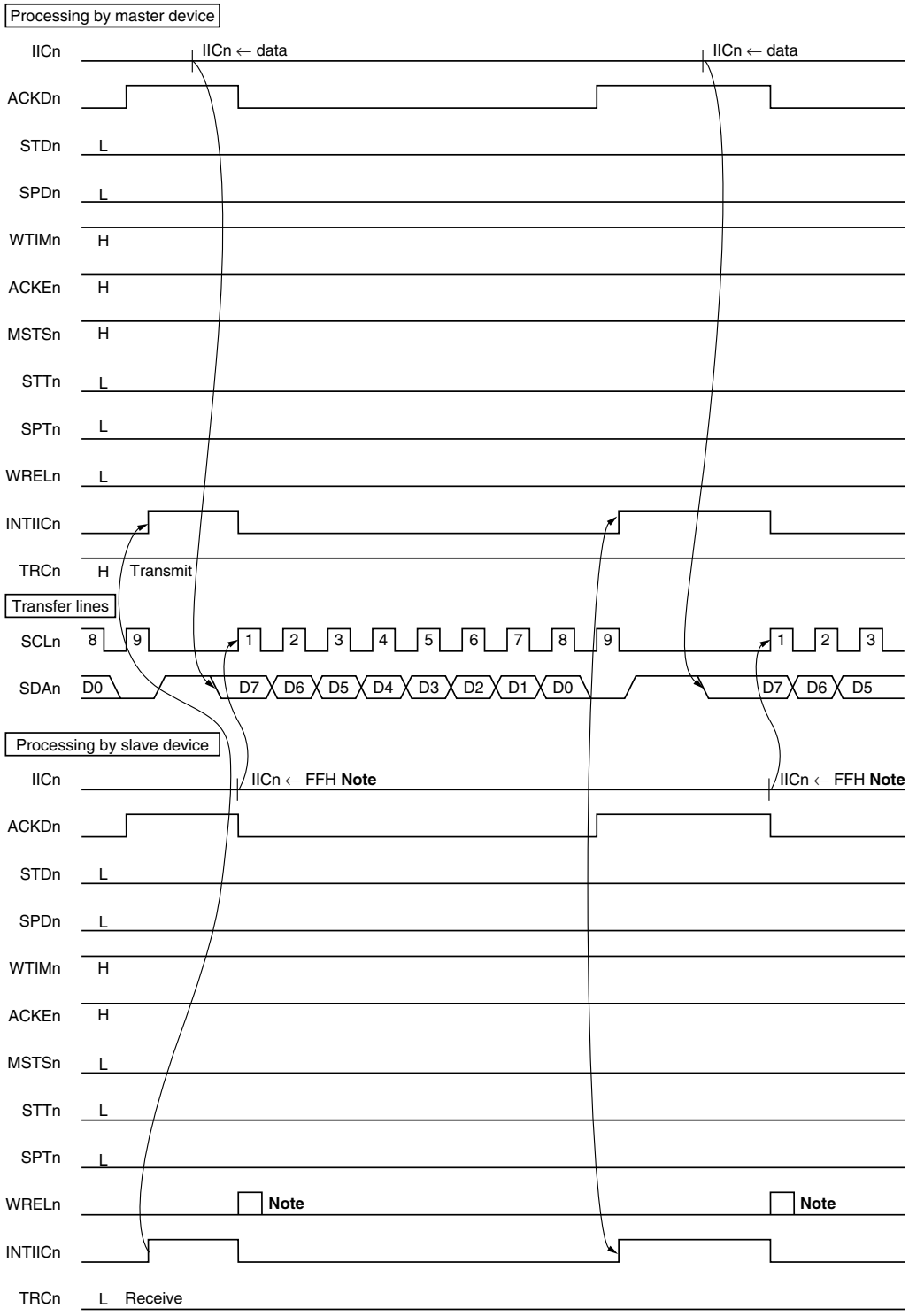


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 11-40. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

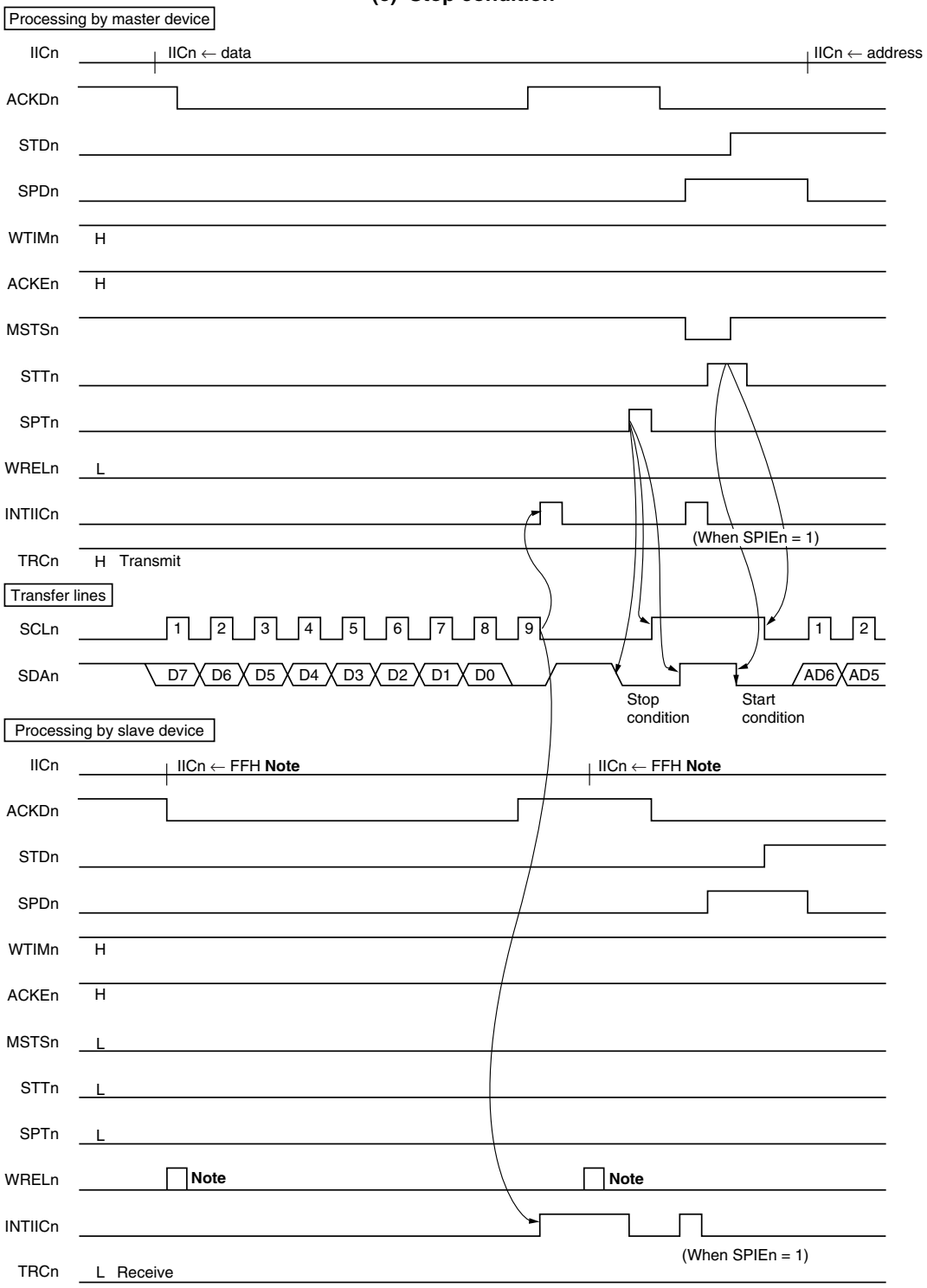


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 11-40. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition

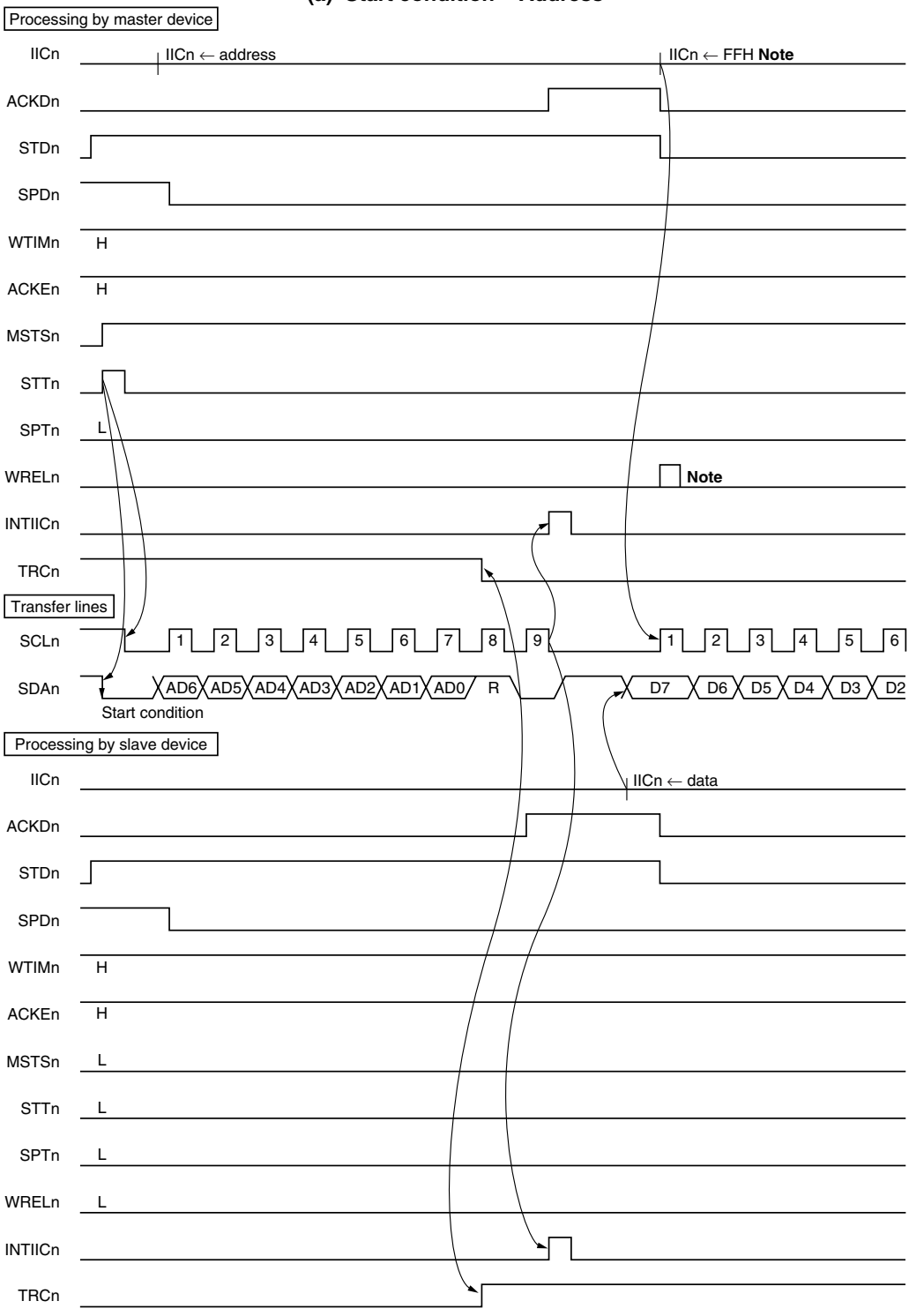


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 11-41. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ Address

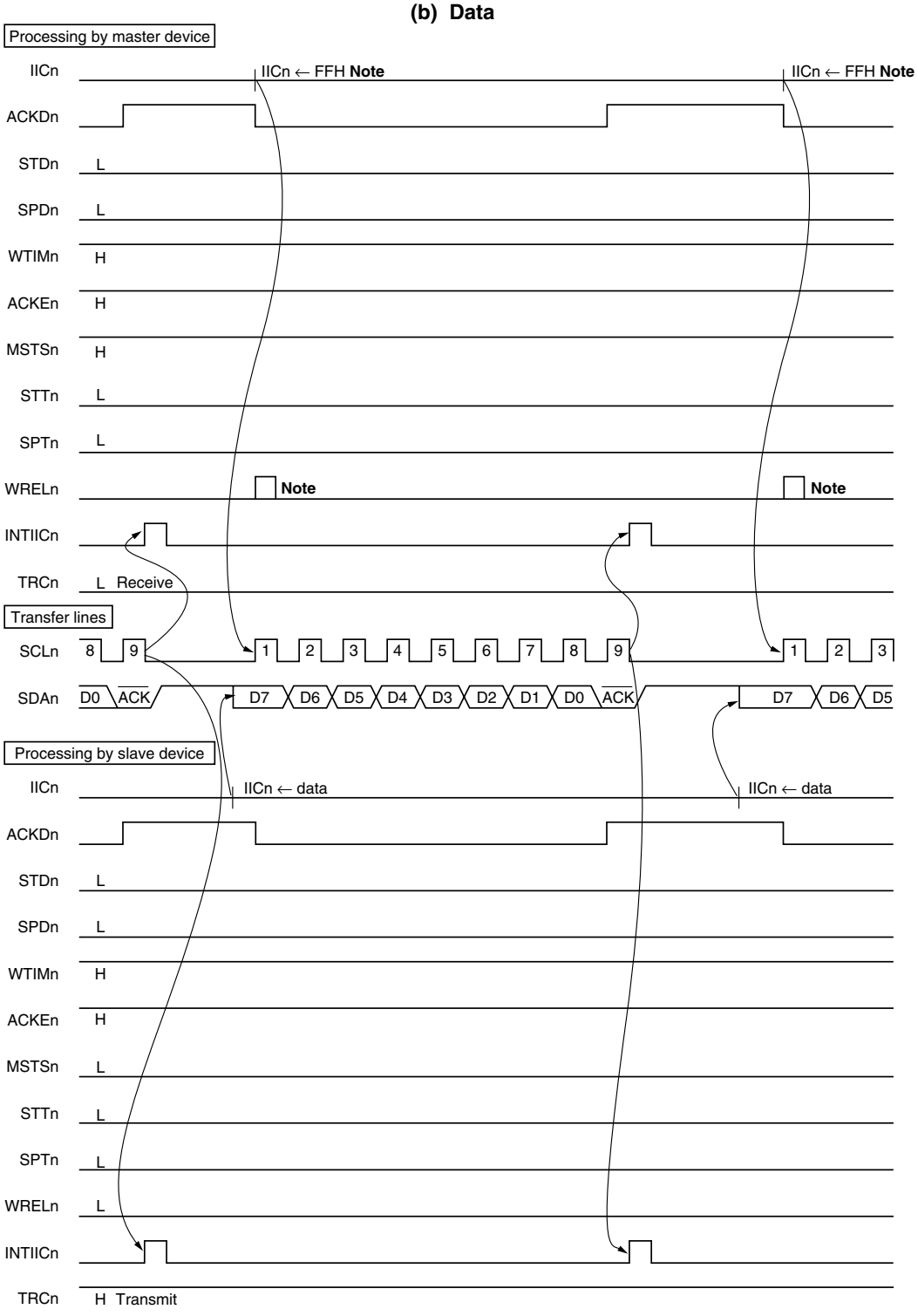


Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 11-41. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

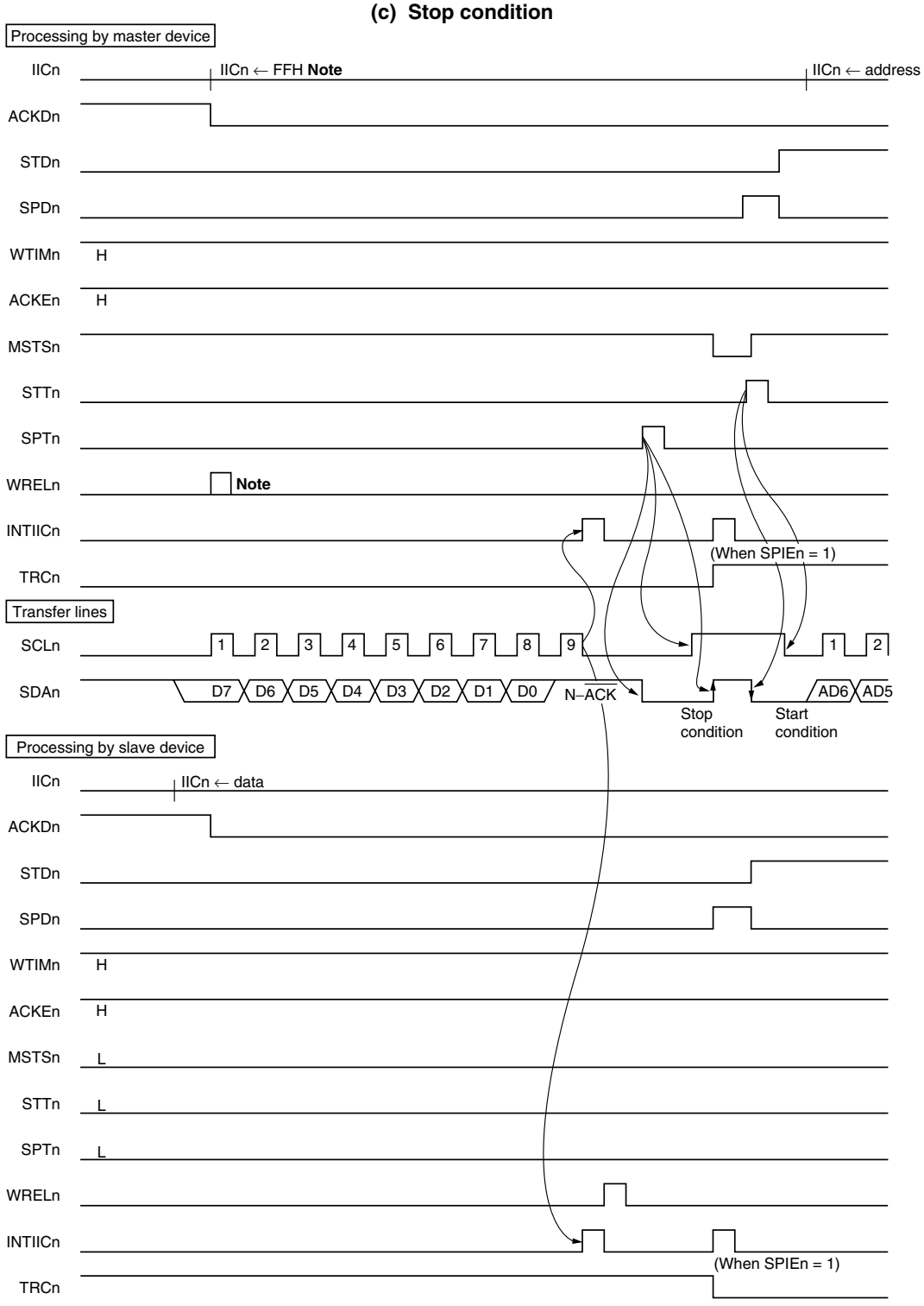


Note To cancel master wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 11-41. 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

11.6 Asynchronous Serial Interface (UART0 to UART3)

UARTn (n = 0 to 3) has the following two operation modes.

(1) Operation stop mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

(2) Asynchronous serial interface mode

This mode enables full-duplex operation in which one byte of data after the start bit is transmitted and received.

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).

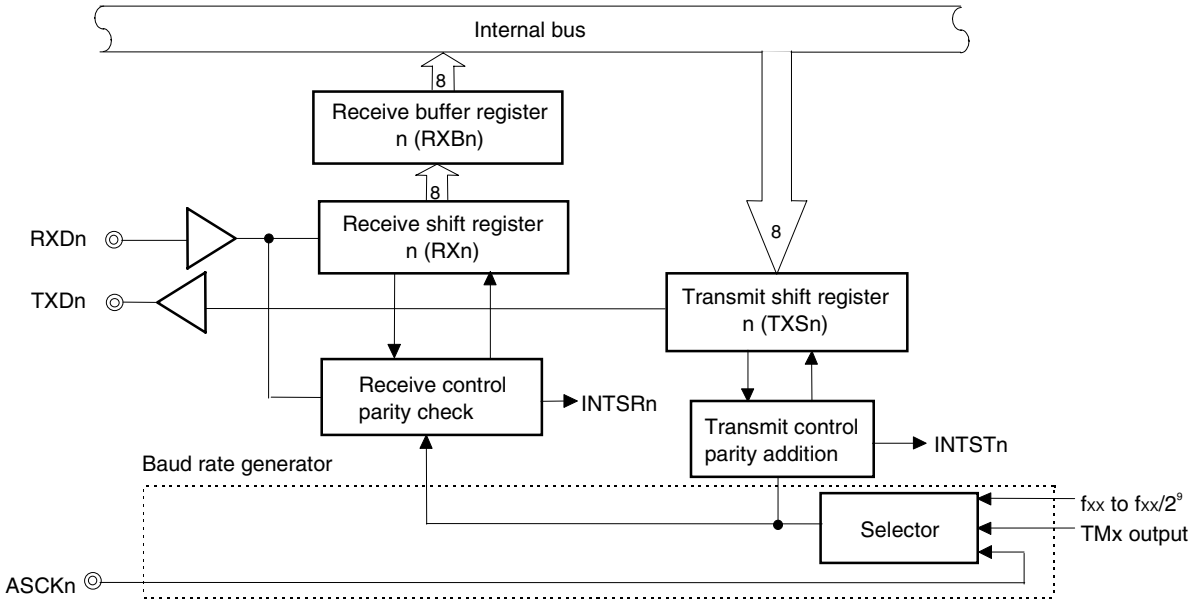
11.6.1 Configuration

The UARTn includes the following hardware.

Table 11-12. Configuration of UARTn

Item	Configuration
Registers	Transmit shift registers 0 to 3 (TXS0 to TXS3) Receive buffer registers 0 to 3 (RXB0 to RXB3)
Control registers	Asynchronous serial interface mode registers 0 to 3 (ASIM0 to ASIM3) Asynchronous serial interface status registers 0 to 3 (ASIS0 to ASIS3) Baud rate generator control registers 0 to 3 (BRGC0 to BRGC3) Baud rate generator mode control registers 00 to 03 (BRGMC00 to BRGMC03) Baud rate generator mode control registers 10 to 13 (BRGMC10 to BRGMC13)

Figure 11-42. Block Diagram of UARTn



Remarks 1. n = 0 to 3

2. TMx output is as shown below.

When n = 0, 2: TM6 output

When n = 1, 3: TM5 output

(1) Transmit shift registers 0 to 3 (TXS0 to TXS3)

TXSn is the register for setting transmit data. Data written to TXSn is transmitted as serial data.

When the data length is set as 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 is written by an 8-bit memory manipulation instruction. It cannot be read.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Caution Do not write to TXSn during a transmit operation.

(2) Receive shift registers 0 to 3 (RX0 to RX3)

The RXn register converts serial data input via the RXDn pin to parallel data. When one byte of data is received at RXn, the received data is transferred to receive buffer registers n (RXBn).

RXn cannot be manipulated directly by a program.

(3) Receive buffer registers 0 to 3 (RXB0 to RXB3)

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 as 7 bits, received data is sent to bit 0 to bit 6 of RXBn. In RXBn, the MSB must be set to 0.

RXBn is read by an 8-bit memory manipulation instruction. It cannot be written.

$\overline{\text{RESET}}$ 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.

11.6.2 UARTn control registers

UARTn is controlled by the following registers (n = 0 to 3).

- 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 to 3 (ASIM0 to ASIM3)

ASIMn is an 8-bit register that controls UARTn's serial transfer operations.

ASIMn can be set by an 8-bit or 1-bit memory manipulation instruction.

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

After reset: 00H R/W Address: ASIM0: FFFFF300H ASIM1: FFFFF310H ASIM2: FFFFF230H ASIM3: FFFFF2B0H

	<7>	<6>	5	4	3	2	1	0
ASIMn (n = 0 to 3)	TXEn	RXEn	PS1n	PS0n	UCLn	SLn	ISRMn	0

TXEn	RXEn	Operating mode	RXDn/Pxx pin function	TXDn/Pxx pin function
0	0	Operation stopped	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 operating mode until after the current serial transmit/receive operation has stopped.

★ **2. Receive error interrupts are not provided in the V850/SC1, V850/SC2, and V850/SC3. To detect receive errors, always set ISRMn to 0.**

★ **3. Always set bit 0 to 0.**

(2) Asynchronous serial interface status registers 0 to 3 (ASIS0 to ASIS3)

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 sets these registers to 00H.

After reset: 00H R Address: ASIS0: FFFFF302H ASIS1: FFFFF312H
ASIS2: FFFFF232H ASIS3: FFFFF2B2H

	7	6	5	4	3	<2>	<1>	<0>
ASISn (n = 0 to 3)	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 as 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 to 3 (BRGC0 to BRGC3)

These registers set the serial clock for UARTn.

BRGCn can be set by an 8-bit memory manipulation instruction.

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

After reset: 00H R/W Address: BRGC0: FFFFF304H BRGC1: FFFFF314H BRGC2: FFFFF234H BRGC3: FFFFF2B4H

BRGCn (n = 0 to 3)

7	6	5	4	3	2	1	0
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 BRGCn is written 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 (n = 0 to 3).

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

After reset: 00H R/W Address: BRGMC01: FFFFF320H BRGMC11: FFFFF322H
BRGMC21: FFFFF23CH BRGMC31: FFFFF2BCH

	7	6	5	4	3	2	1	0
BRGMCn1 (n = 0 to 3)	0	0	0	0	0	0	0	TPSn3

After reset: 00H R/W Address: BRGMC00: FFFFF30EH BRGMC10: FFFFF31EH
BRGMC20: FFFFF23AH BRGMC30: FFFFF2BAH

	7	6	5	4	3	2	1	0
BRGMCn0 (n = 0 to 3)	0	0	0	0	0	TPSn2	TPSn1	TPSn0

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, 2: TM6 output at n = 1, 3: TM5 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 BRGMCn0 or n1 is written 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 or n1 during communication processing.

★ **2. Always set bit 7 to 3 of BRGMCn0 to 0.**

Remarks 1. Source clock of 8-bit counter: f_{SCK}

2. If the selected clock is specified as a timer output, the P17/T05/TI5 and P30/T06/TI6 pins do not need to be in timer output mode.

11.6.3 Operations

UARTn has the following two operation modes.

- Operation stopped mode
- Asynchronous serial interface mode

(1) Operation stopped mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

In operation stopped mode, pins can be used as ordinary port pins.

(a) Register settings

Operation stopped mode settings are made via bits TXEn and RXEn of asynchronous serial interface mode register n (ASIMn).

Figure 11-43. ASIMn Setting (Operation Stopped Mode)

After reset: 00H R/W Address: ASIM0: FFFFF300H ASIM1: FFFFF310H
ASIM2: FFFFF230H ASIM3: FFFFF2B0H

	<7>	<6>	5	4	3	2	1	0
ASIMn (n = 0 to 3)	TXEn	RXEn	PS1n	PS0n	UCLn	SLn	ISRMn	0

TXEn	RXEn	Operating mode	RXDn/Pxx pin function	TXDn/Pxx pin function
0	0	Operation stopped	Port function	Port function

Cautions 1. Do not switch the operating mode until after the current serial transmit/receive operation has stopped.

★ **2. Always set bit 0 to 0.**

(2) Asynchronous serial interface mode

This mode enables full-duplex operation in which one byte of data after the start bit is transmitted and received.

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 to 3).

Figure 11-44. ASIMn Setting (Asynchronous Serial Interface Mode)

After reset: 00H R/W Address: ASIM0: FFFFF300H ASIM1: FFFFF310H ASIM2: FFFFF230H ASIM3: FFFFF2B0H

	<7>	<6>	5	4	3	2	1	0
ASIMn (n = 0 to 3)	TXEn	RXEn	PS1n	PS0n	UCLn	SLn	ISRMn	0

TXEn	RXEn	Operating 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

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 operating mode until after the current serial transmit/receive operation has stopped.

★ **2. Receive error interrupts are not provided in the V850/SC1, V850/SC2, and V850/SC3. To detect receive errors, always set ISRMn to 0.**

★ **3. Always set bit 0 to 0.**

Figure 11-45. ASISn Setting (Asynchronous Serial Interface Mode)

After reset: 00H R Address: ASIS0: FFFFF302H ASIS1: FFFFF312H ASIS2: FFFFF232H ASIS3: FFFFF2B2H

	7	6	5	4	3	<2>	<1>	<0>
ASISn (n = 0 to 3)	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 as two bits by setting bit 2 (SLn) in the 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 11-46. BRGCn Setting (Asynchronous Serial Interface Mode)

After reset: 00H R/W Address: BRGC0: FFFFF304H BRGC1: FFFFF314H BRGC2: FFFFF234H BRGC3: FFFFF2B4H

BRGCn (n = 0 to 3)

7	6	5	4	3	2	1	0
MDLn7	MDLn6	MDLn5	MDLn4	MDLn3	MDLn2	MDLn1	MDLn0

MD Ln7	MD Ln6	MD Ln5	MD Ln4	MD Ln3	MD Ln2	MD Ln1	MD Ln0	Input clock selection	k
0	0	0	0	0	×	×	×	Setting prohibited	–
0	0	0	0	1	0	0	0	f_{SCK}/8	8
0	0	0	0	1	0	0	1	f_{SCK}/9	9
0	0	0	0	1	0	1	0	f_{SCK}/10	10
0	0	0	0	1	0	1	1	f_{SCK}/11	11
0	0	0	0	1	1	0	0	f_{SCK}/12	12
0	0	0	0	1	1	0	1	f_{SCK}/13	13
0	0	0	0	1	1	1	0	f_{SCK}/14	14
0	0	0	0	1	1	1	1	f_{SCK}/15	15
0	0	0	1	0	0	0	0	f_{SCK}/16	16
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	f_{SCK}/255	255

Cautions 1. Reset input sets BRGCn 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 BRGCn is written 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 11-47. BRGMCn0 and BRGMCn1 Settings (Asynchronous Serial Interface Mode)

After reset: 00H R/W Address: BRGMC01: FFFFF320H BRGMC11: FFFFF322H
BRGMC21: FFFFF23CH BRGMC31: FFFFF2BCH

	7	6	5	4	3	2	1	0
BRGMCn1 (n = 0 to 3)	0	0	0	0	0	0	0	TPSn3

After reset: 00H R/W Address: BRGMC00: FFFFF30EH BRGMC10: FFFFF31EH
BRGMC20: FFFFF23AH BRGMC30: FFFFF2BAH

	7	6	5	4	3	2	1	0
BRGMCn0 (n = 0 to 3)	0	0	0	0	0	TPSn2	TPSn1	TPSn0

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, 2: TM6 output at n = 1, 3: TM5 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 BRGMCn0 or n1 is written 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 or BRGMCn1 during communication processing.

★ **2. Always set bits 7 to 3 of BRGMCn0 to 0.**

Remarks 1. f_{XX}: Main clock oscillation frequency

2. If the selected clock is specified as a timer output, the P17/T05/TI5 and P30/T06/TI6 pins do not need to be in 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 tolerance**

  The baud rate tolerance depends on the number of bits in a frame and the counter division ratio [1/(16 + k)].

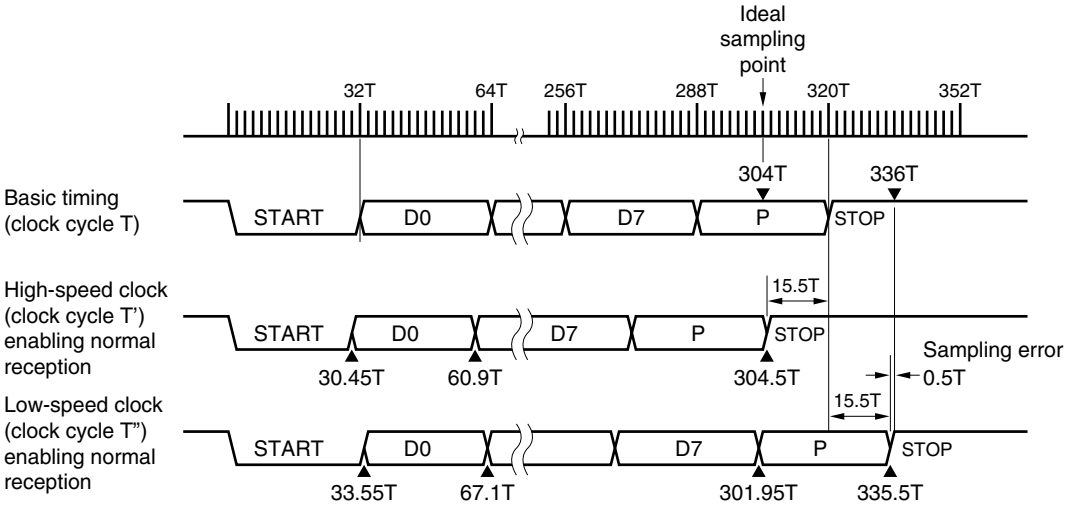
  Table 11-13 shows the relationship between the main clock and the baud rate, and Figure 11-48 shows an example of the allowable baud rate error range.

Table 11-13. Relationship Between Main Clock and Baud Rate

Baud Rate (bps)	f_{XX} = 20 MHz			f_{XX} = 18.87 MHz			f_{XX} = 16 MHz		
	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)
32	–	–	–	–	–	–	–	–	–
64	–	–	–	–	–	–	244	9	0.06
128	152	9	–0.39	144	9	–0.02	244	8	0.06
300	130	8	0.16	123	8	–0.12	208	7	0.16
600	130	7	0.16	123	7	–0.12	208	6	0.16
1200	130	6	0.16	123	6	–0.12	208	5	0.16
2400	130	5	0.16	123	5	–0.12	208	4	0.16
4800	130	4	0.16	123	4	–0.12	208	3	0.16
9600	130	3	0.16	123	3	–0.12	208	2	0.16
19200	130	2	0.16	123	2	–0.12	208	1	0.16
38400	130	1	0.16	123	1	–0.12	208	0	0.16
76800	130	0	0.16	123	0	–0.12	104	0	0.16
150000	67	0	–0.50	63	0	–0.16	53	0	0.63
300000	33	0	1.01	31	0	1.45	27	0	–1.24
524000	19	0	0.44	18	0	0.03	15	0	1.78
1250000	8	0	0.00	–	–	–	–	–	–

Remark f_{XX}: Main clock oscillation frequency

Figure 11-48. Allowable Baud Rate Error Range (When k = 16), Including Sampling Errors



Remark T: 8-bit counter's source clock cycle

$$\text{Allowable baud rate error range (when k = 16)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

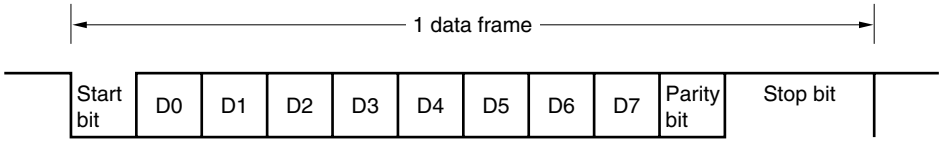
(3) Communication operations

(a) Data format

As shown in Figure 11-49, 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 to 3).

Figure 11-49. 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 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 to 3).

If a receive error occurs, information about the receive error can be ascertained by reading asynchronous serial interface status register n (ASISn) (n = 0 to 3).

(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 a parity bit is controlled so that the number of “1” bits is even. 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 among the receive data is counted, including a parity bit, and a parity error is generated 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 the number of “1” bits is odd. 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 among the receive data is counted, including a parity bit, and a parity error is generated 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 be generated 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 be generated.

(c) Transmission

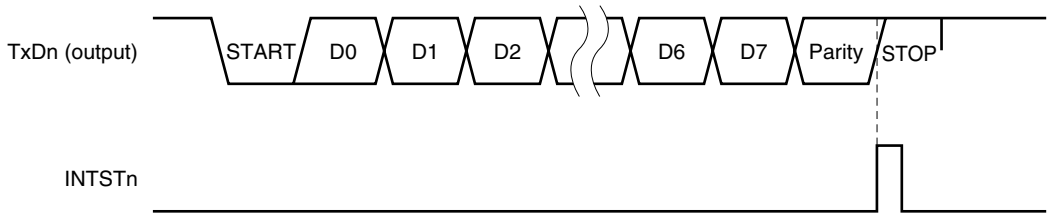
A transmit operation is started when transmit data is written to transmit shift register n (TXSn). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting a transmit operation shifts out the data in TXSn, thereby emptying TXSn, after which a transmission completion interrupt (INTSTn) is issued.

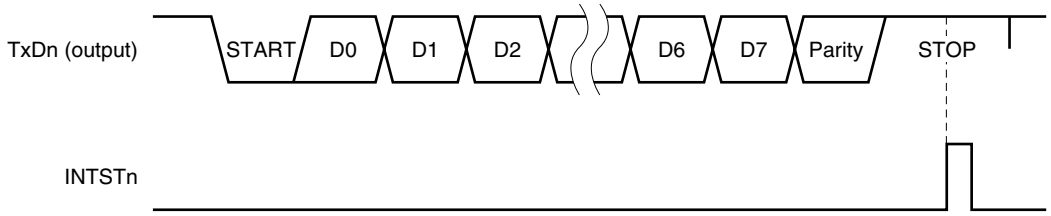
The timing of the transmission completion interrupt is shown below.

Figure 11-50. Timing of Asynchronous Serial Interface Transmission 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, enter a $\overline{\text{RESET}}$ to restore normal operation).

Whether or not a transmit operation is in progress can be determined via software using the transmission completion interrupt (INTSTn) or the interrupt request flag (STIFn) set by INTSTn.

Remark n = 0 to 3

(d) Reception

A receive operation is enabled when bit 6 (RXEn) of asynchronous serial interface mode register n (ASIMn) is set to 1, and 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 reception completion interrupt (INTSRn) occurs.

Even if an error has occurred, the receive data in which the error occurred is still transferred to RXBn.

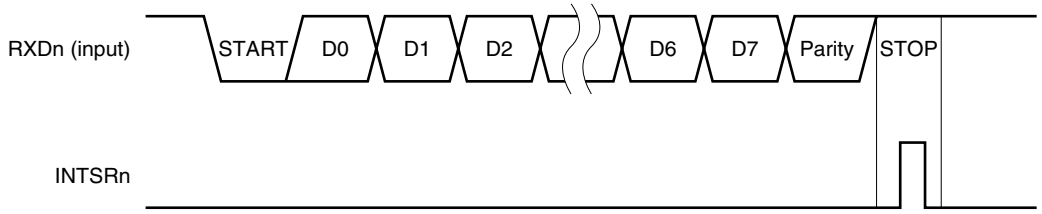
When an error occurs, INSTRn is generated if bit 1 (ISRMn) of ASIMn is cleared (0). On the other hand, INTSRn is not generated if the ISRMn bit is set (1).

★ The receive error type can be ascertained by reading the contents of ASISn in the reception completion interrupt servicing (INTSRn).

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 occur.

The timing of the asynchronous serial interface reception completion interrupt is shown below.

Figure 11-51. Timing of Asynchronous Serial Interface Reception 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 to 3

(e) Receive error

There are three types of error during a receive operation: parity errors, framing errors, and overrun errors. When, as the result of data reception, an error flag is set in asynchronous serial interface status register n (ASISn). ★

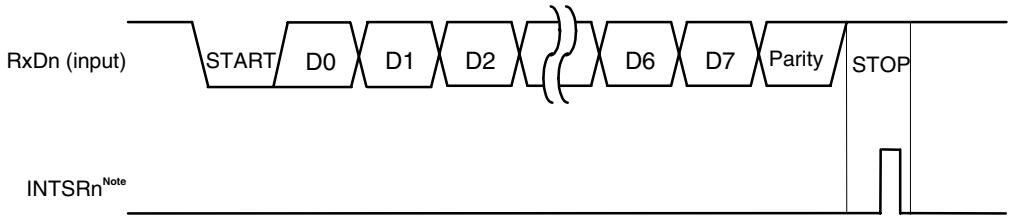
By reading the contents of ASISn during receive completion interrupt servicing (INTSRn), it is possible to detect which error has occurred at reception.

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 11-14. 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 11-52. Receive Error Timing**



Note Even if a receive error occurs when the ISRMn bit of ASIMn is set (1), INTSRn is not generated. The receive error type can be ascertained by reading the contents of ASISn in the reception completion interrupt servicing (INTSRn).

Cautions 1. The contents of asynchronous serial interface status register n (ASISn) are reset (0) by reading receive buffer register n (RXBn) or receiving subsequent data. To check the contents of an error, always read ASISn before reading RXBn.

2. Be sure to read receive buffer register n (RXBn) even when a receive error has occurred. 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 to 3

11.6.4 Standby function

(1) Operation in HALT mode

Serial transfer is 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 is 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 is performed normally.

CHAPTER 12 A/D CONVERTER

12.1 Function

The A/D converter converts analog input signals into digital values, has a resolution of 10 bits, and can handle 12 channels of analog input signals (ANI0 to ANI11).

The V850/SC1, V850/SC2, and V850/SC3 support low-speed conversion and a low-power-consumption mode.

(1) Hardware start

Conversion is started by trigger input (ADTRG) (rising edge, falling edge, or both rising and falling edges can be specified).

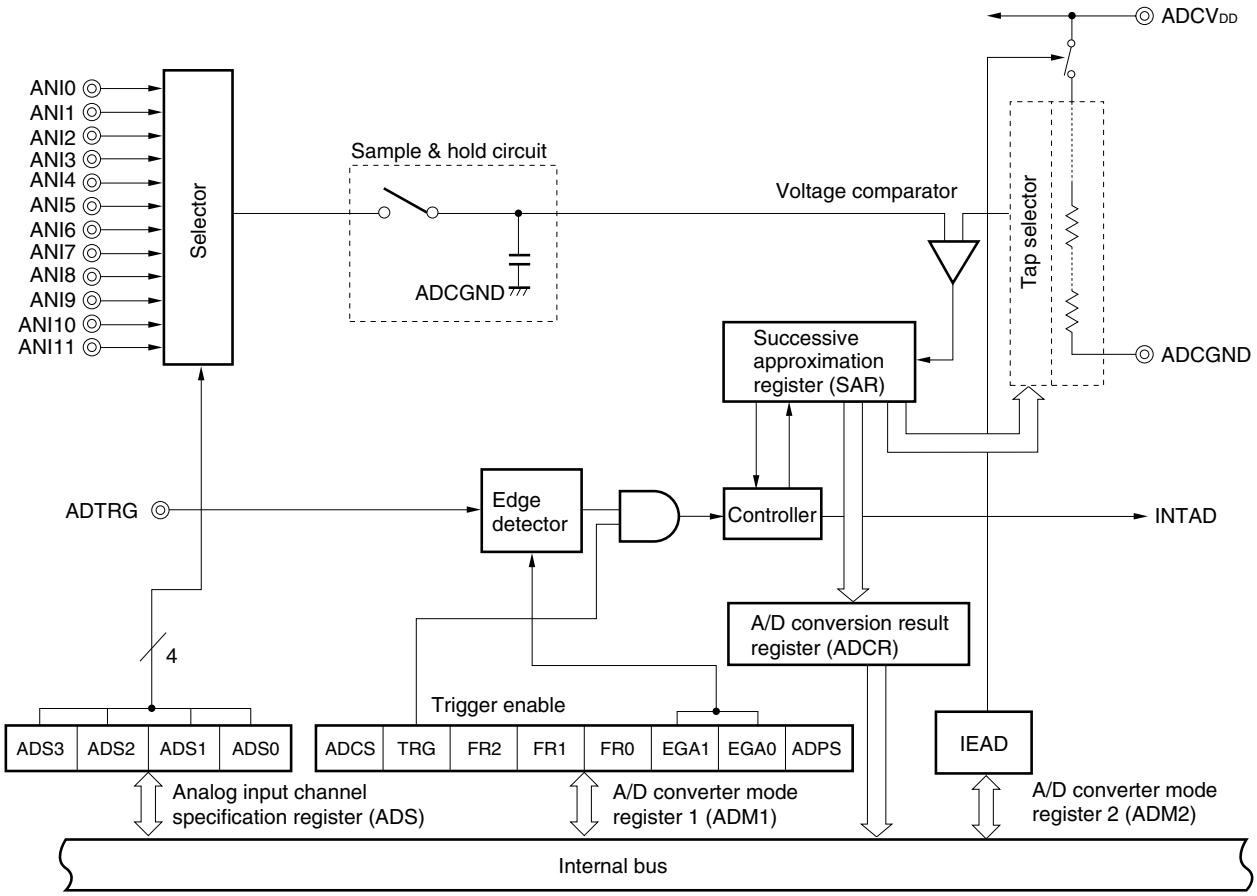
(2) Software start

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 a hardware start, conversion stops once it has been completed, and an interrupt request (INTAD) is generated. If conversion has been started by a software start, conversion is performed repeatedly. Each time conversion has been completed, INTAD is generated.

The block diagram is shown below.

Figure 12-1. Block Diagram of A/D Converter



12.2 Configuration

The A/D converter includes the following hardware.

Table 12-1. Configuration of A/D Converter

Item	Configuration
Analog inputs	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 held down to the least significant bit (LSB) (i.e., when 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 complete, 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 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 Writing 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 conversion result during A/D conversion (ADCS = 1). Incorrect conversion results may 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 $ADCV_{DD}$ and ADCGND and generates a voltage for comparison with the analog input signal.

(6) ANI0 to ANI11 pins

These are analog input pins for the 12 channels of the A/D converter, and are used to input analog signals to be converted into digital signals. Pins other than ones selected as analog input with the analog input channel specification register (ADS) can be used as input ports.

Caution Make sure that the voltages input to ANI0 through ANI11 do not exceed the rated values. If a voltage higher than $ADCV_{DD}$ or lower than ADCGND (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) ADCGND pin

This is the ground pin of the A/D converter. Always make the potential at this pin the same as that at the GND0 pin even when the A/D converter is not in use.

(8) $ADCV_{DD}$ pin

This is the analog power supply pin of the A/D converter. Always make the potential at this pin the same as that at the V_{DD0} pin even when the A/D converter is not in use.

12.3 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.

ADM1 is set by an 8-bit or 1-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets 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]	f_{XX}		
					20 MHz	18.87 MHz	16 MHz
0	0	0	0	168/f_{XX}	8.4 μs	8.9 μs	Setting prohibited
0	0	0	1	120/f_{XX}	6.0 μs	6.4 μs	7.5 μs
0	0	1	0	84/f_{XX}	Setting prohibited	Setting prohibited	5.25 μs
0	0	1	1	60/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	0	48/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	1	36/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	0	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	1	1	1	12/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
1	0	0	0	168/f_{XX} + 64/f_{XX}	8.4 + 3.2 μs	8.9 + 3.4 μs	Setting prohibited
1	0	0	1	120/f_{XX} + 60/f_{XX}	6.0 + 3.0 μs	6.4 + 3.2 μs	7.5 + 3.75 μs
1	0	1	0	84/f_{XX} + 42/f_{XX}	Setting prohibited	Setting prohibited	5.25 + 2.63 μs
1	0	1	1	60/f_{XX} + 30/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0	48/f_{XX} + 24/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1	36/f_{XX} + 18/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited
1	1	1	0	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	1	1	12/f_{XX} + 6/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited

EGA1	EGA0	Valid edge specification for external trigger signal
0	0	No edge detection
0	1	Detected at falling edge
1	0	Detected at rising edge
1	1	Detected at both rising and falling edges

ADPS	Comparator control while 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.

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 the ADPS bit to 1 when the ADCS bit is set to 0.

(2) Analog input channel specification register (ADS)

This register specifies the port for inputting the analog voltage to be converted into a digital signal.

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

$\overline{\text{RESET}}$ input sets ADS to 00H.

After reset: 00H R/W Address: FFFFF3C2H

	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
1	0	1	0	ANI10
1	0	1	1	ANI11
Other than above				Setting prohibited

★ **Caution Always set bits 7 to 4 to 0.**

(3) A/D converter mode register 2 (ADM2)

This register specifies connection/disconnection of $ADCV_{DD}$ and the series resistor string.

ADM2 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets 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 $ADCV_{DD}$ and series resistor string
1	Connect between $ADCV_{DD}$ and series resistor string

12.4 Operation

12.4.1 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) $ADCV_{DD}$.

<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) $ADCV_{DD}$, the MSB of the SAR remains set. If the analog input voltage is less than (1/2) $ADCV_{DD}$, 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) $ADCV_{DD}$
- Bit 9 = 0: (1/4) $ADCV_{DD}$

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.

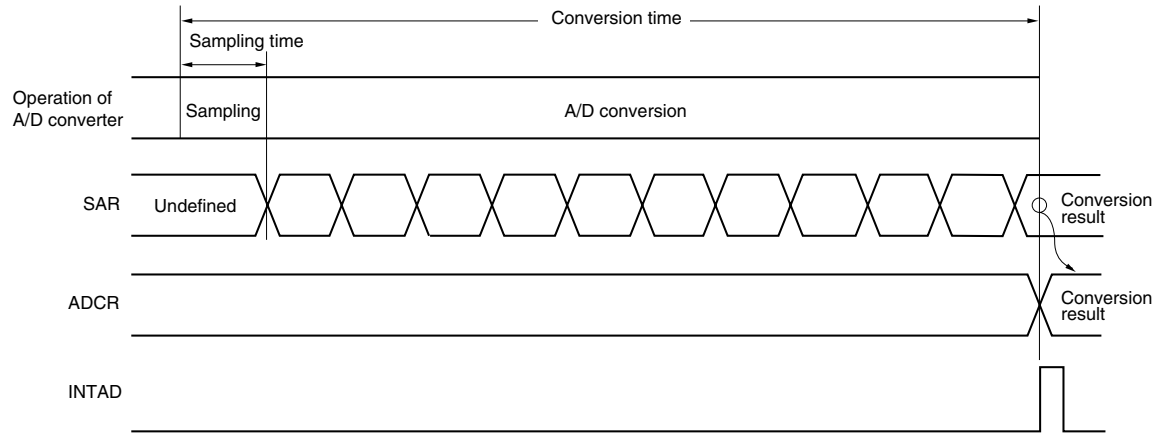
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage ≤ Voltage tap: Bit 8 = 0

<7> The above steps are repeated until the bit 0 of the SAR has been manipulated.

<8> When comparison of all the 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 to 1 (from 0) may not satisfy the ratings.

Figure 12-2. 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 (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 (1) at this time, conversion is started from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register (ADCR) to 0000H.

12.4.2 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:

$$ADCR = INT(\frac{V_{IN}}{ADCV_{DD}} \times 1024 + 0.5)$$

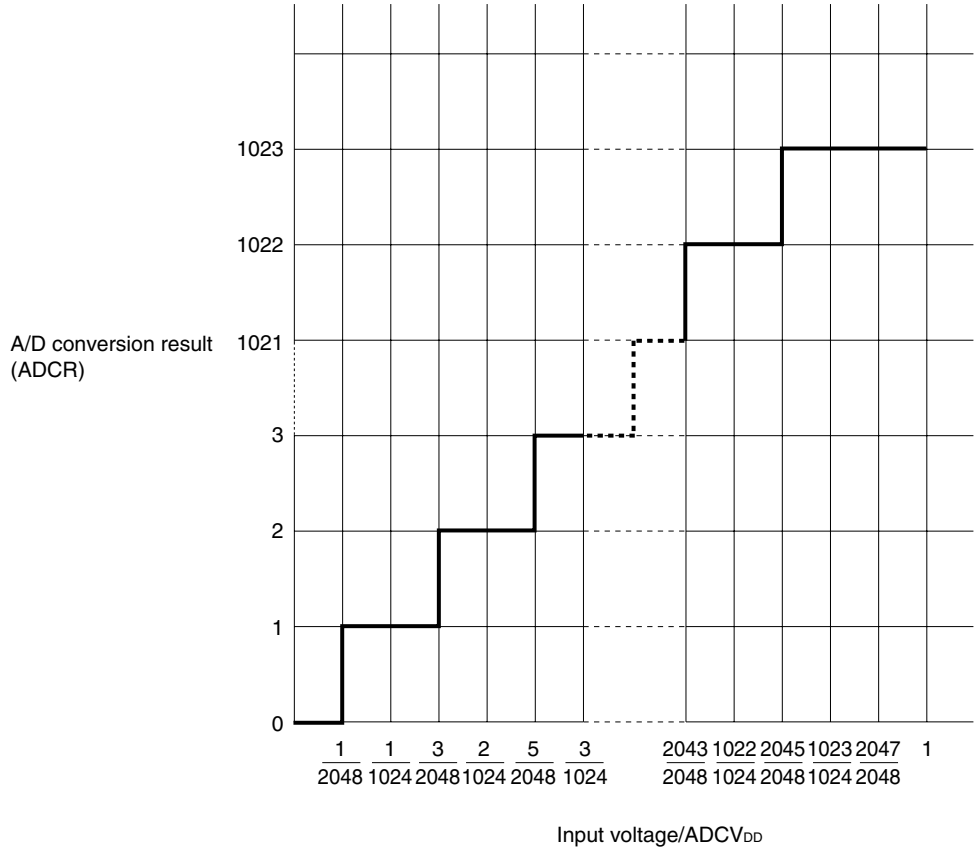
Or,

$$(ADCR - 0.5) \times \frac{ADCV_{DD}}{1024} \leq V_{IN} < (ADCR + 0.5) \times \frac{ADCV_{DD}}{1024}$$

INT ( ): Function that returns integer of value in ( )
V_{IN}: Analog input voltage
$ADCV_{DD}$: A/D converter reference 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 12-3. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 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.

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

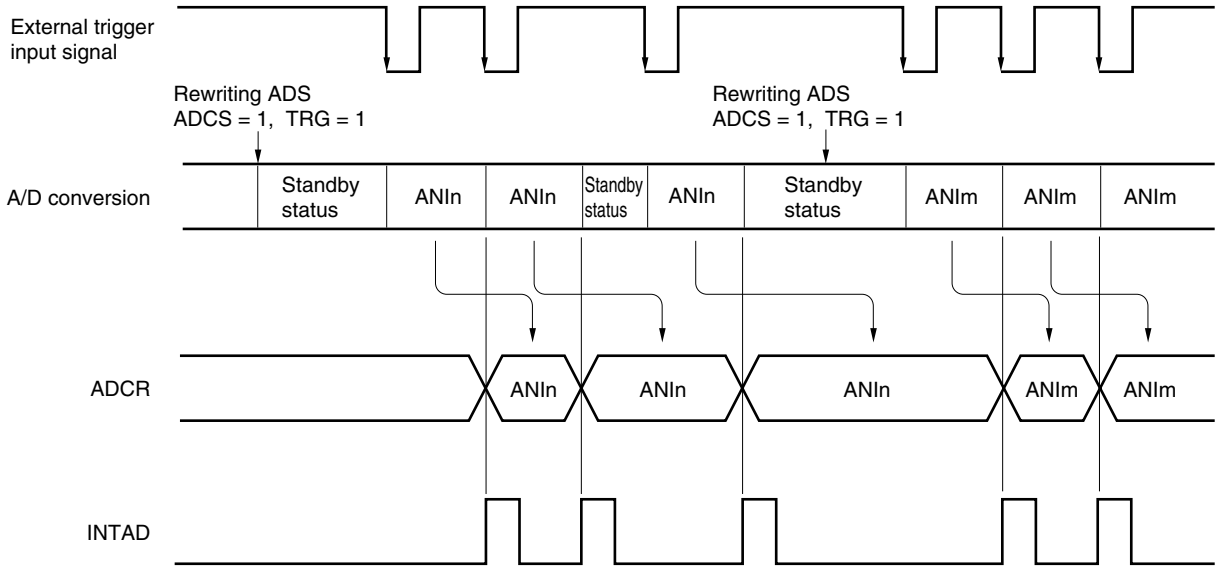
A/D conversion is on standby if bit 6 (TRG) and bit 7 (ADCS) of A/D converter mode register 1 (ADM1) are set to 1. When an external trigger signal is input, the A/D converter starts converting the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) into a digital signal.

When A/D conversion is complete, 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 12-4. 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

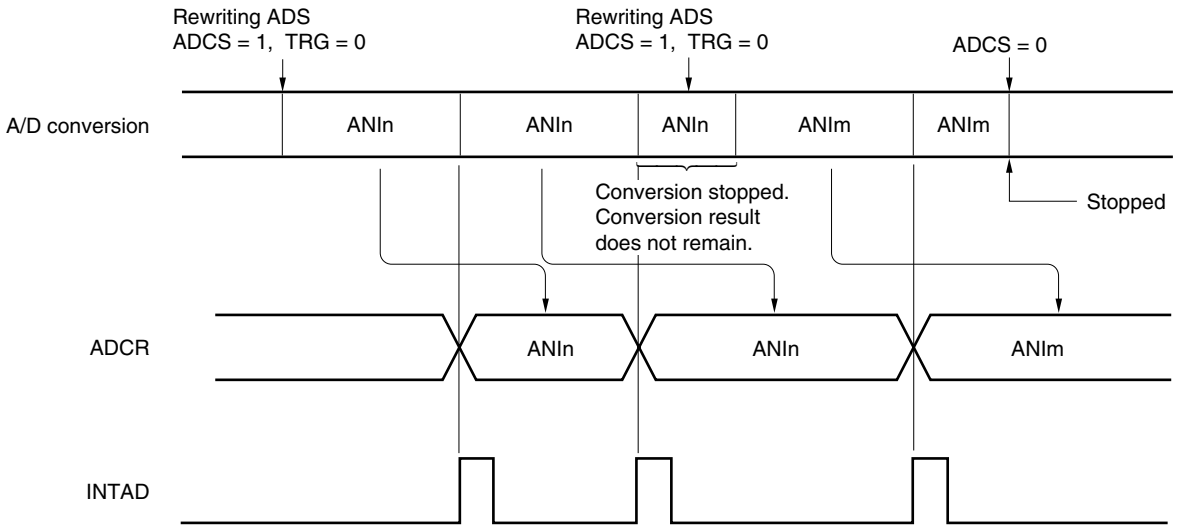
If bit 6 (TRG) and bit 7 (ADCS) of A/D converter mode register 1 (ADM1) are set to 1, the A/D converter starts converting the voltage applied to an analog input pin specified by the analog input channel specification register (ADS) into a digital signal.

When A/D conversion is complete, 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 12-5. A/D Conversion by Software Start



Remarks 1. n = 0, 1, ..., 11

2. m = 0, 1, ..., 11

12.5 Low-Power-Consumption Mode

The V850/SC1, V850/SC2, and V850/SC3 feature a function that can cut or connect the current between $ADCV_{DD}$ and the series resistor string. Switching can be performed by setting A/D converter mode register 2 (ADM2).

When not using the A/D converter, cut off the tap selector (a function to reduce current) from the voltage supply block ($ADCV_{DD}$) while A/D conversion is stopped (ADCS = 0) to cut the current consumption.

- Set the ADPS bit of A/D converter mode register 1 (ADM1) to 1.
- Set the ADPS bit of A/D converter mode register 2 (ADM1) to 0.

When the ADPS bit is reset to 0 (comparator on), stabilization time (5 μs max.) is required before starting A/D conversion. Therefore, secure a wait of at least 5 μs by software.

12.6 Cautions

(1) Current consumption in standby mode

The A/D converter stops operation in the IDLE and STOP modes (operable in the HALT mode). At this time, the current consumption of the A/D converter can be reduced by stopping the conversion (by re-setting bit 7 (ADCS) of A/D converter mode register 1 (ADM1) to 0).

(2) Input range of ANI0 to ANI11

Keep the input voltage of the ANI0 through ANI11 pins to within the rated range. If a voltage greater than $ADCV_{DD}$ or lower than ADCGND (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 accepted during the writing of ADCR.

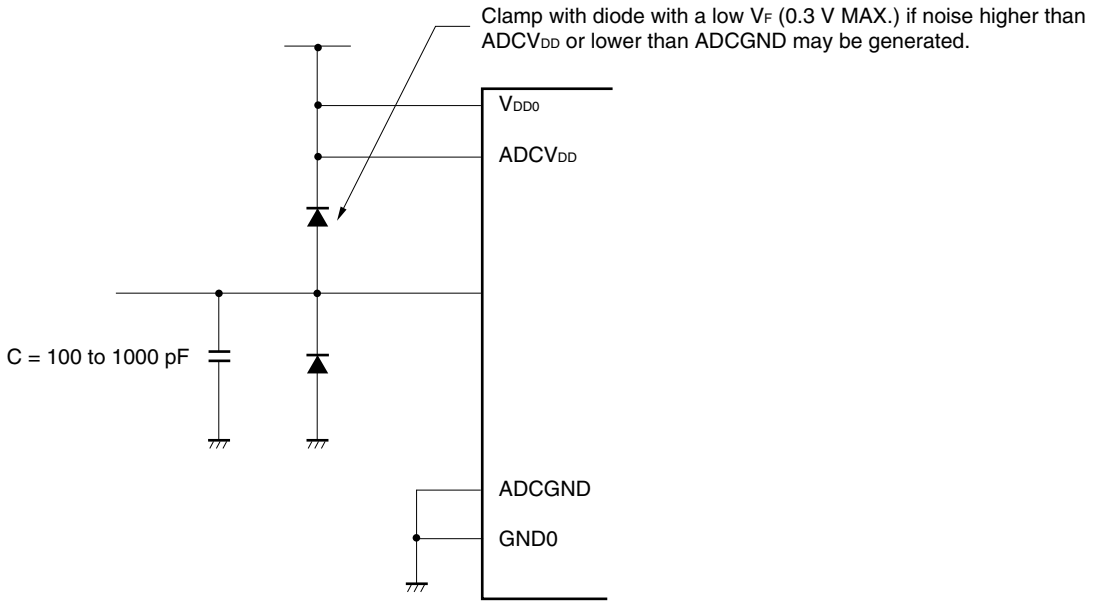
★ **<3> Conflict between writing of ADCR and writing A/D converter mode register 1 (ADM1) or analog input channel specification register (ADS)**

When ADM1 or ADS is written immediately after ADCR is written following the end of A/D conversion, an undefined value is stored in the ADCR register, so the conversion result is not guaranteed.

(4) Countermeasures against noise

To keep the resolution of 10 bits, it is necessary to prevent noise from being superimposed on the 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 below is recommended.

Figure 12-6. 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.

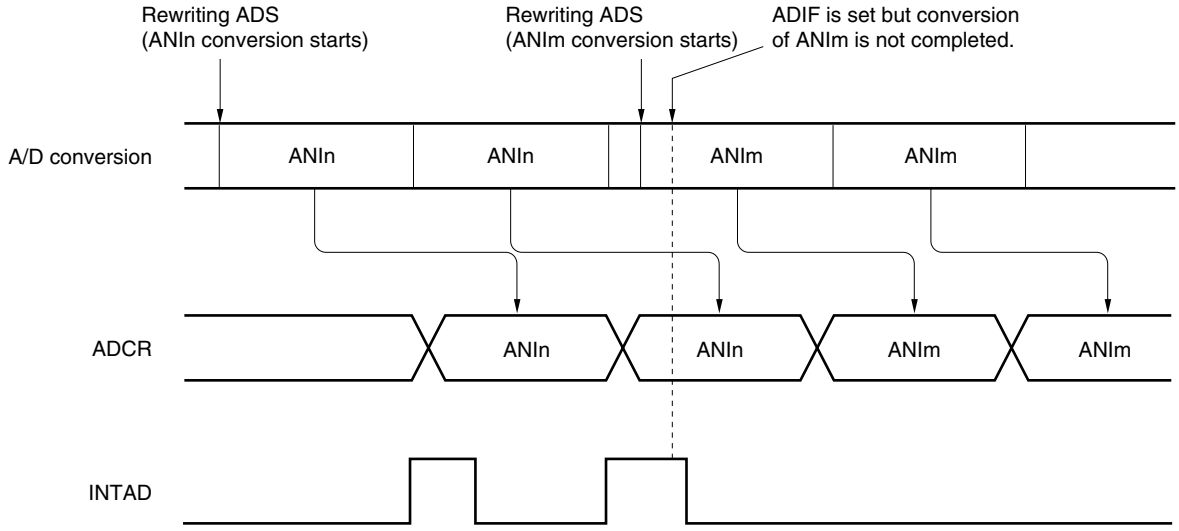
(6) Interrupt request flag (ADIF)

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 12-7. A/D Conversion End Interrupt Generation Timing



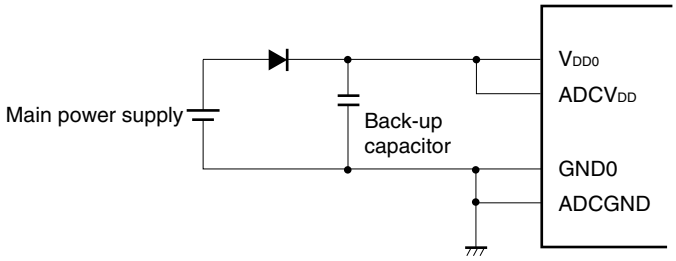
Remarks 1. n = 0, 1, ..., 11

2. m = 0, 1, ..., 11

(7) $ADCV_{DD}$ pin

The $ADCV_{DD}$ 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 back-up power supply is used, therefore, be sure to apply the same voltage as the V_{DD0} pin to the $ADCV_{DD}$ pin as shown below.

Figure 12-8. Handling of $ADCV_{DD}$ Pin



(8) Reading out A/D converter result register (ADCR)

★ Writing 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 conversion result during A/D conversion (ADCS = 1). Incorrect conversion results may be read if the timing is other than the above.

CHAPTER 13 DMA FUNCTIONS

13.1 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 interfaces, 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.

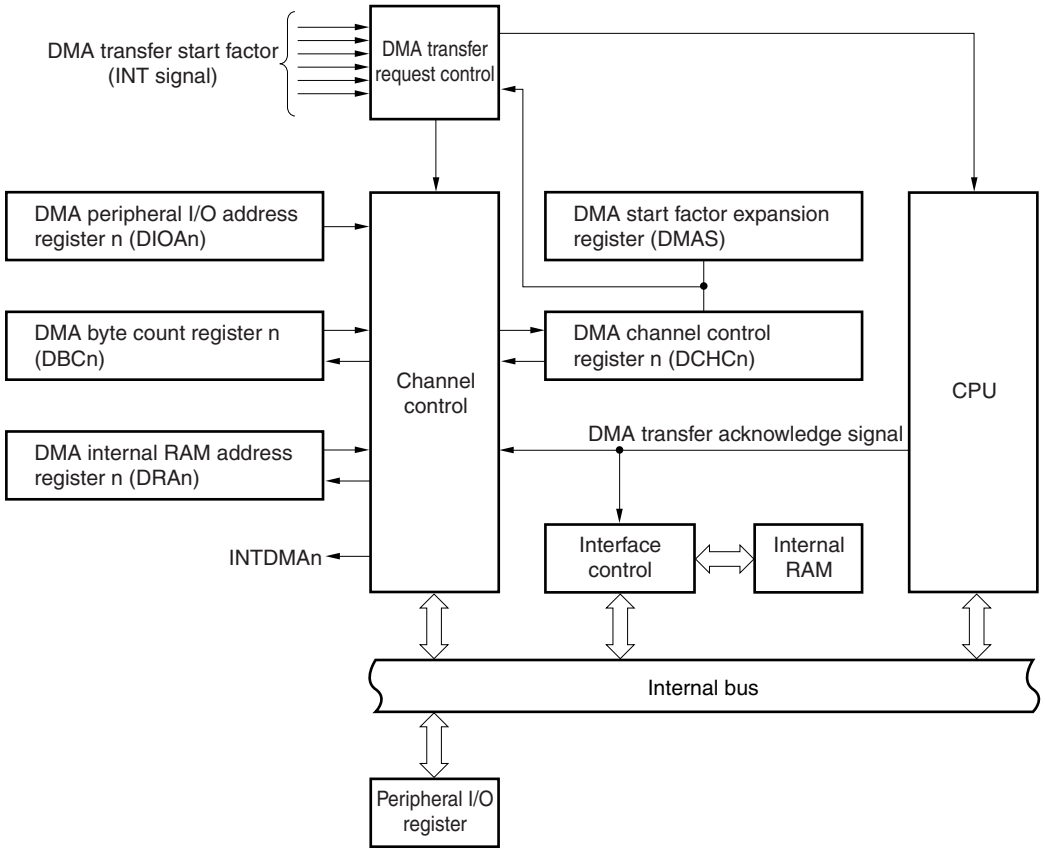
DMA0 > DMA1 > DMA2 > DMA3 > DMA4 > DMA5

13.2 Transfer Completion Interrupt Request

After a DMA transfer has occurred a specified number of times and the TCn bit in corresponding DMA channel control register is 0 to 5 (DCHC0 to DCHC5) has been set to 1, a DMA transfer completion interrupt request (INTDMA0 to INTDMA5) for the interrupt controller occurs on each channel.

★ 13.3 Configuration

Figure 13-1. DMA Block Diagram



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 DMA transfer start factor (INT signal) specified by DMA channel control register n (DCHCn) and the DMA start factor expansion register (DMAS) is input.

When the DMA transfer request signal is acknowledged, the CPU generates a DMA transfer acknowledge signal for the channel control block and interface control block after the current CPU processing has finished.

(2) Channel control block

The channel control block distinguishes the DMA transfer channel (DMA0 to DMA5) to be transferred and controls the internal ROM, 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 start factors (INT signals) are generated simultaneously.

13.4 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 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	1	0
DIOAn	0	0	0	0	0	0	IOAn9 to IOAn1		0

(n = 0 to 5)

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, POCS, VM45C, FCAN register (see CHAPTER 19)

(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 register for each product are shown below.

Table 13-1. Internal RAM Area Usable in DMA

Product		Internal RAM Capacity	RAM Size Usable in DMA	RAM Area Usable in DMA
V850/SC1	μPD703068Y, 70F3089Y	24 KB	16 KB	xxFF9000H to xxFFBFFFH, xxFFE000H to xxFFEFFFH
V850/SC2	μPD703069Y, 70F3089Y			
V850/SC3	μPD703088Y, 703089Y, 70F3089Y			

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	0
DRAn	0	0	RAn13 to RAn00	

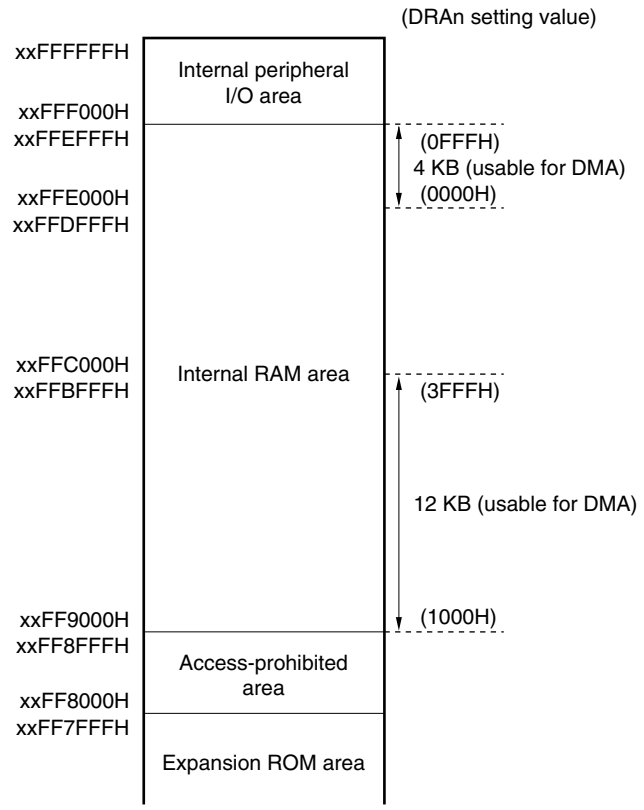
(n = 0 to 5)

The following shows the correspondence between the DRAn setting value and the internal RAM area.

(a) V850/SC1 (μPD703068Y, 70F3089Y)
V850/SC2 (μPD703069Y, 70F3089Y)
V850/SC3 (μPD703088Y, 703089Y, 70F3089Y)

Set the DRAn register to a value in the range of 0000H to 0FFFH or 1000H to 3FFFH (n = 0 to 5).

Figure 13-2. Correspondence Between DRAn Setting Value and Internal RAM



Caution Do not set odd addresses for 16-bit transfer (DCHCn register DSn = 1).

Remark The values in parentheses indicate the DRAn register setting values.

(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 DMA transfer.

The transfer count is decremented by 1 per transfer if the transfer is a byte (8-bit) transfer, and by 2 per transfer if the transfer is a 16-bit transfer. Transfer ends 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 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 **(6) Start factor settings** (n = 0 to 5).

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

Refer to **(6) Start factor settings** for the setting of the TTYPn1 and TTYPn0 bits.

These registers can be read/written in 8- or 1-bit units.

After reset: 00H R/W Address: DCHC0 FFFFF186H DCHC3 FFFFF1B6H
DCHC1 FFFFF196H DCHC4 FFFFF1C6H
DCHC2 FFFFF1A6H DCHC5 FFFFF1D6H

DCHCn (n = 0 to 5)

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

TDIRn	Transfer direction control between peripheral I/O and internal RAM[Note 2]
0	From internal RAM to peripheral I/O
1	From peripheral I/O to internal RAM

DSn	Control of transfer data size for DMA transfer[Note 2]
0	8-bit transfer
1	16-bit transfer

ENn	Control of DMA transfer enable/disable status[Note 3]
0	Disabled
1	Enabled (reset to 0 after DMA transfer is completed)

Notes 1. TCn (n = 0 to 5) is set (1) when a specified number of transfers are complete, and is cleared (0) when a write instruction is executed.

2. 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).

3. After the specified number of transfers is complete, this bit is cleared to 0.

(6) Start factor settings

The DMA start factor is set using bits 2 to 0 (DMAS2 to DMAS0) of the DMA start factor expansion register (DMAS) in combination with bits 4 and 3 (TTYPn1, TTYPn0) of DMA channel control registers 0 to 5 (DCHC0 to DCHC5)(n = 0 to 5).

Table 13-2 shows the DMA start factor settings.

Cautions 1. If the interrupt that is the DMA start factor is not masked, interrupt servicing is performed each time DMA starts.

To prevent interrupt servicing from being performed, mask the interrupt.

★ **2. If an interrupt source is generated asynchronously to the internal system clock, do not set the interrupt source as a multiple DMA start trigger (for example, when the serial interface is operated on the external clock input).**

If set, the priority order of DMA may be reversed.

Table 13-2. Start Factor Settings

Channel n	DMAS2	DMAS1	DMAS0	TTYPn1	TTYPn0	DMA Transfer Start Factor Setting
0	x	x	x	0	0	INTCSI0/INTIIC0
				0	1	INTCSI5
				1	0	INTAD
				1	1	INTTM00
1	x	x	0	0	0	INTCSI0/INTIIC0
			1	0	0	INTCSI1/INTSR0
			x	0	1	INTST0
				1	0	INTP0
				1	1	INTSR3
2	x	0	x	0	0	INTCSI2/INTIIC1
		1		0	0	INTCSI3/INTSR1
		x		0	1	INTP6
				1	0	INTIE1 (setting prohibited for other than V850/SC2)
				1	1	INTAD
3	0	x	x	0	0	INTCSI6
	1			0	0	INTCSI3/INTSR1
	x			0	1	INTST2
				1	0	INTIE1 (setting prohibited for other than V850/SC2)
				1	1	INTTM70
4	x	x	x	0	0	INTST1
				0	1	INTCSI4/INTSR0
				1	0	INTCSI6
				1	1	INTSR2
5	x	x	x	0	0	INTST3
				0	1	INTCSI4/INTSR0
				1	0	INTCSI2/INTIIC1
				1	1	INTTM6/INTP9

Remarks 1. DMAS2 to DMAS0: Bits 2 to 0 of the DMA start factor expansion register (DMAS)

2. TTYPn1, TTYPn0: Bits 4 and 3 of DMA channel control register n (DCHCn)

3. x: don’t care

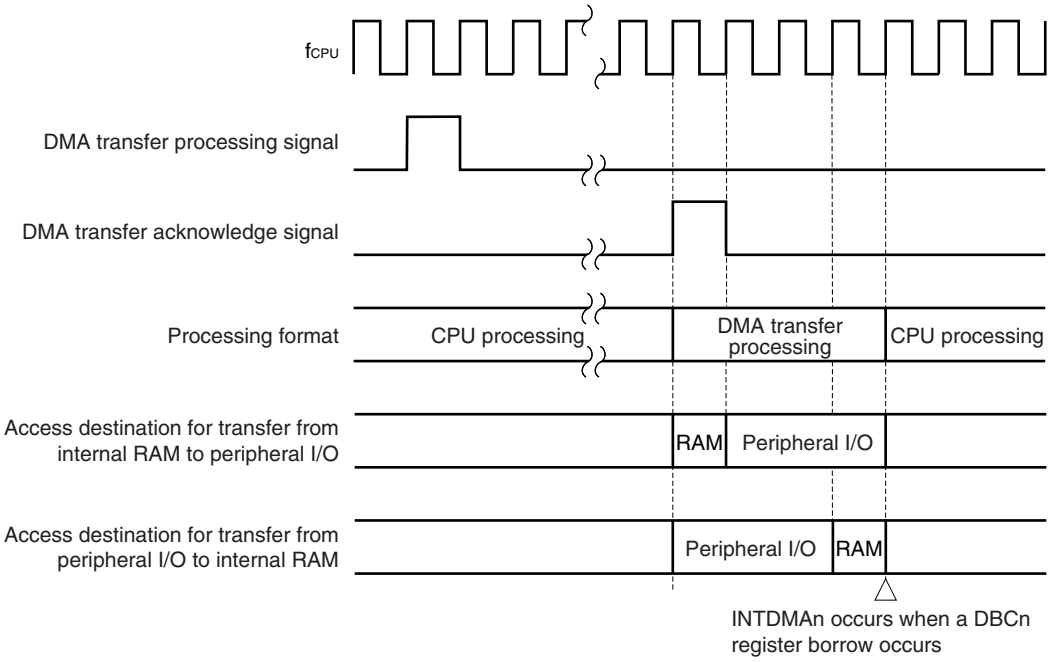
★ 13.5 Operation

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 13-3. DMA Transfer Operation Timing

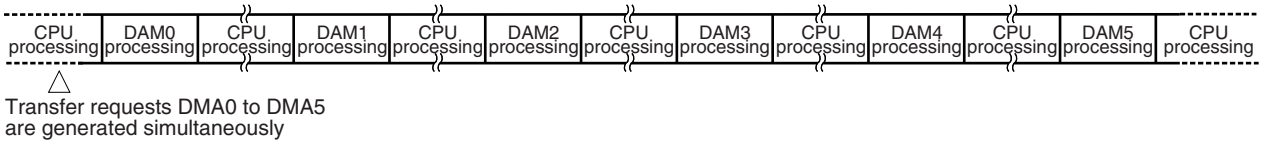


Remark n = 0 to 5

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 13-4. 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.

★ 13.6 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 7-2**)
n: Peripheral unit number (see **Table 7-2**)

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.

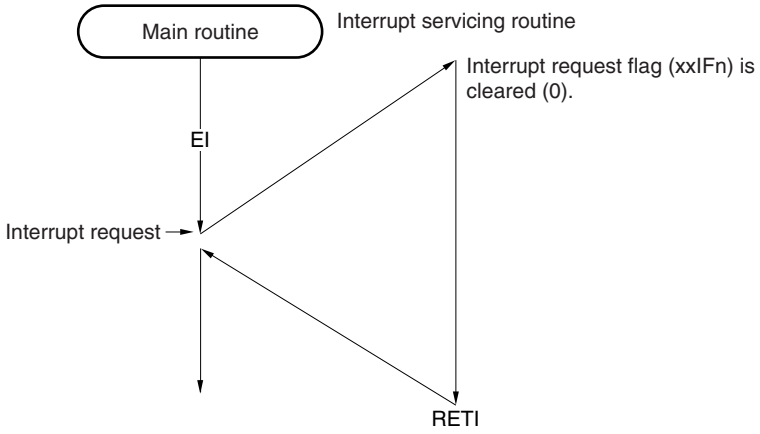
(ii) The interrupt request flag (xxIFn) must be cleared (0) at the start of the interrupt routine.

Caution When the DMA function is not used, execution of (i) or (ii) is not necessary.

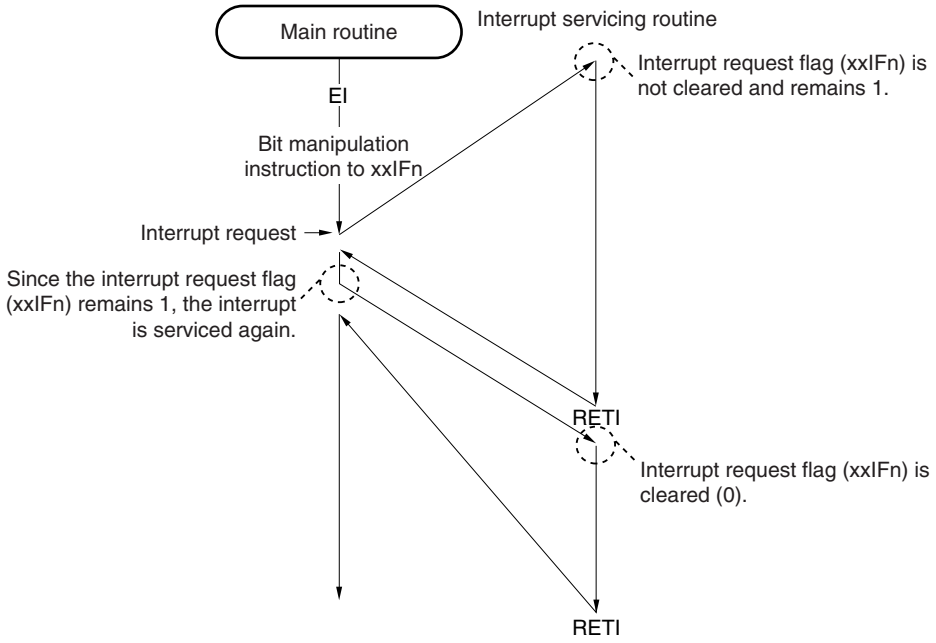
Remark xx: Identifying name of peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

Figure 13-5. When Interrupt Servicing Occurs Twice During DMA Operation (1/2)

(a) Normal interrupt servicing



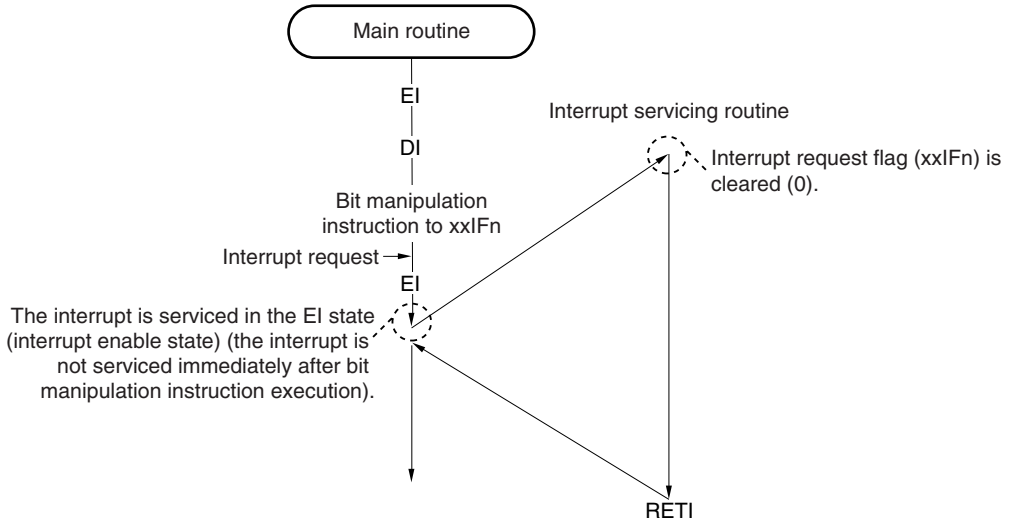
(b) Interrupt servicing when interrupt servicing occurs twice



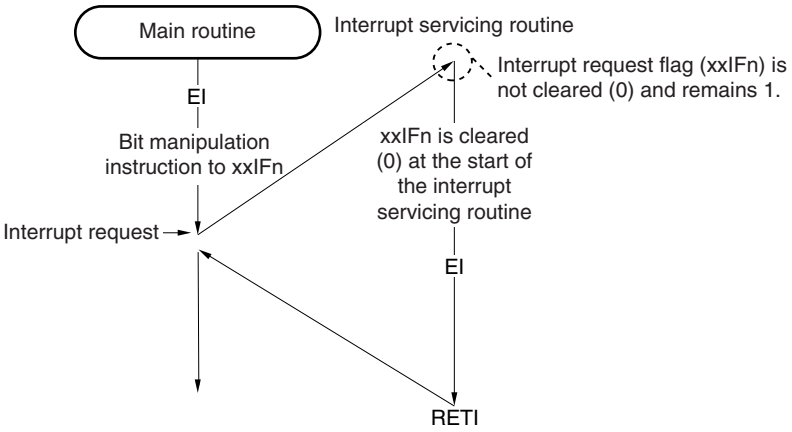
Remark xx: Identifying name of peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

Figure 13-5. When Interrupt Servicing Occurs Twice During DMA Operation (2/2)

(c) Countermeasure (use condition (i))



(d) Countermeasure (use condition (ii))



Remark xx: Identifying name of peripheral unit (see **Table 7-2**)
n: Peripheral unit number (see **Table 7-2**)

CHAPTER 14 RESET FUNCTION

14.1 General

There are three methods used to generate a reset signal.

(1) External reset by $\overline{\text{RESET}}$ signal input
(2) Internal reset by watchdog timer loop time detection (watchdog timer overflow)
(3) Internal reset by power-on-clear (POC)

(1) External reset by $\overline{\text{RESET}}$ signal input

When low-level input occurs at 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 canceled and the CPU resumes program execution after the oscillation stabilization time has elapsed (2^{18}/fxx). 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.

(2) Internal reset by watchdog timer loop time detection

When the watchdog timer overflows, a system reset is performed and the various on-chip hardware devices are initialized. In addition, the main clock stops oscillation during the reset period, although the subclock continues oscillation.

The reset by the watchdog timer is released immediately after reset and the CPU resumes program execution after the oscillation stabilization time has elapsed (2^{18}/fxx).

(3) Internal reset by power-on-clear (POC)

When either of the following conditions is satisfied, a system reset is performed by power-on-clear.

- When the supply voltage is less than 3.5 V[Note] at power application
- When the supply voltage is less than 2.2 V[Note] in STOP mode
- ★ When the supply voltage becomes less than 3.5 V[Note] (other than when STOP mode is selected)

When any one of the conditions above is satisfied, a system reset is performed and the various on-chip hardware devices are initialized. In addition, the main clock stops oscillation during the reset period, although the subclock continues oscillation.

The power-on-clear reset is released after the power supply voltage reaches a certain voltage and the system starts program execution after the oscillation stabilization time has elapsed (2^{18}/fxx).

★ Whether the 3.5 V power-on-clear reset detection voltage is enabled or disabled is set using the POCC register (when initial power is supplied, it is enabled).

Note The voltage values are maximum values; a system reset is actually performed at lower voltage than each.

14.2 Pin Operations

During the system reset period, almost all pins are set to high impedance (except for $\overline{\text{RESET}}$, X2, XT2, CPUREG, V_{DD0}, V_{DD1}, $ADCV_{DD}$, ADCGND, $PORTV_{DD0}$ to $PORTV_{DD2}$, PORTGND0, PORTGND1, GND0, GND1, GND2, and V_{PP}/IC).

Accordingly, if connected to an external memory device, be sure to attach a pull-up (or pull-down) resistor to 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 such effects at the signal outputs by on-chip peripheral I/O functions and output ports.

Figure 14-1. System Reset Timing by $\overline{\text{RESET}}$ Signal Input

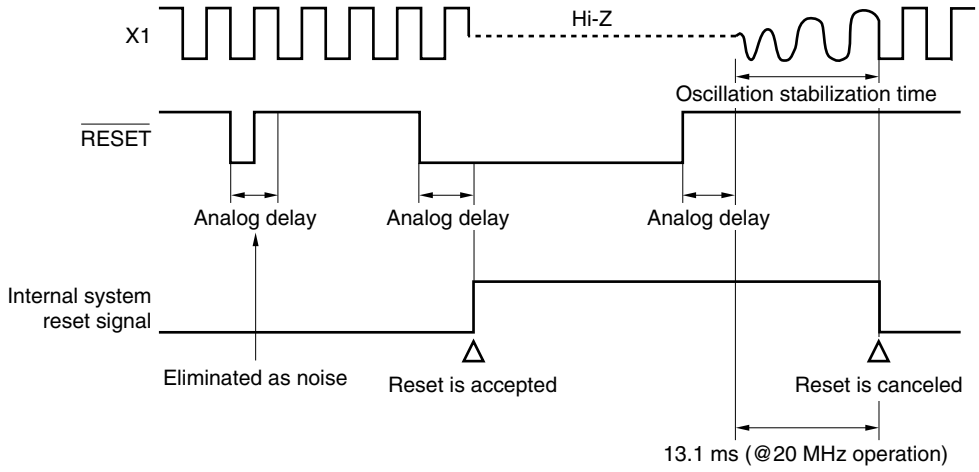


Figure 14-2. System Reset Timing by Watchdog Timer Overflow

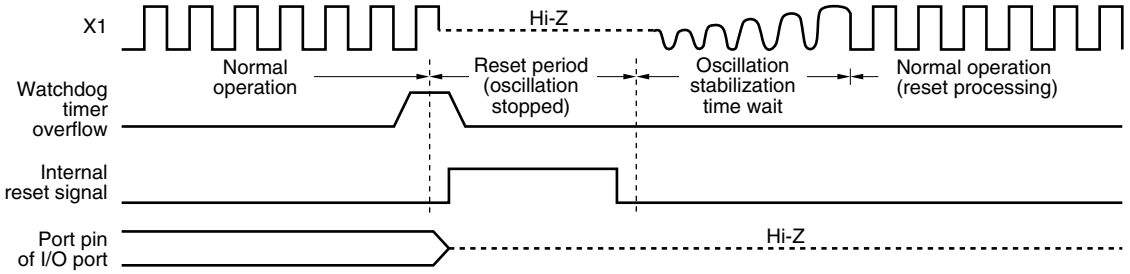
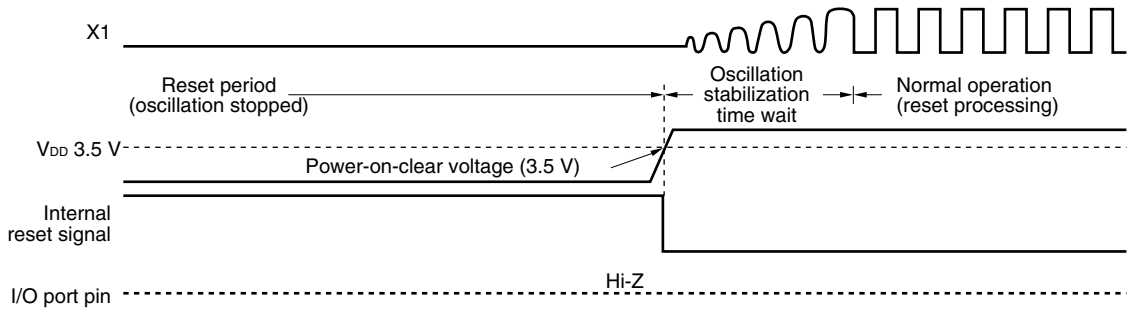
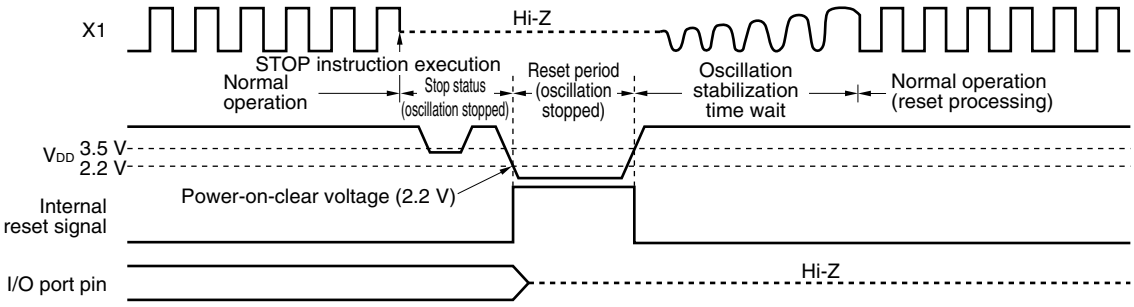


Figure 14-3. System Reset Timing by Power-on-Clear

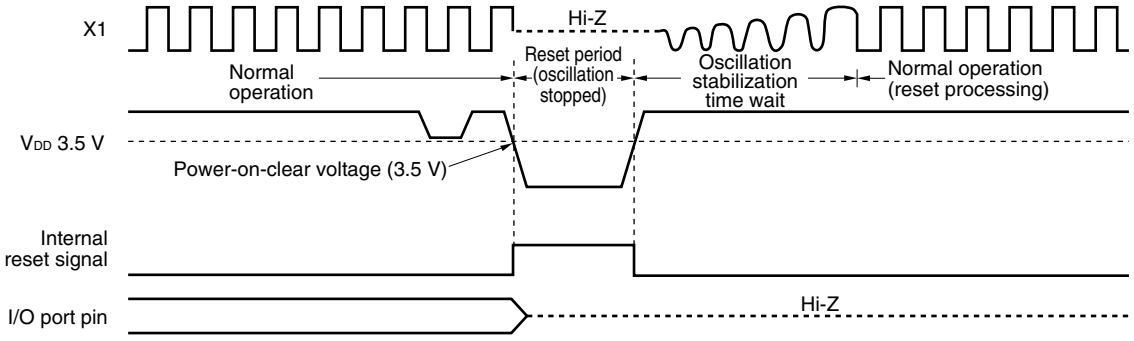
(a) At power application



(b) In STOP mode



(c) In normal operating mode (including HALT mode)



14.3 Power-on-Clear Operation

The V850/SC1, V850/SC2, and V850/SC3 include a power-on-clear circuit (POC), through which low-voltage detection and V_{DD0} pin voltage detection (4.2 ±0.3 V) can be performed by means of the POC status register (POCS).

(1) POC status register (POCS)

When a power-on-clear is generated, bit 0 of the POCS register is set to 1.

In addition, if the voltage level at the V_{DD0} pin is less than 4.2 ±0.3 V, bit 1 (VM45) of the POCS register is set to 1, enabling detection of a voltage level of less than 4.2 ±0.3 V at the V_{DD0} pin.

In the case of a reset generated by the $\overline{\text{RESET}}$ pin, however, the POCM and VM45 bits retain their previous statuses. A low-voltage state can be detected by reading the POCS register following reset cancellation.

The POCS register is read-only, using an 8-bit memory manipulation instruction. This register is reset when read.

After reset: Retained[Note] R Address: FFFFF07AH

	7	6	5	4	3	2	1	0
POCS	0	0	0	0	0	0	VM45	POCM

POCM	Detection of power-on-clear generation status
0	Power-on-clear not generated
1	Power-on-clear reset generated

VM45	Detection of V_{DD0} pin voltage level
0	V_{DD0} pin voltage of less than 4.5 V not detected
1	V_{DD0} pin voltage of less than 4.5 V detected

Note This value is 03H only after a power-on-clear reset; it is not initialized by a reset from the $\overline{\text{RESET}}$ pin.

(2) VM45 control register (VM45C)

The detection status (detected/undetected) according to the VM45 bit of the POCS register can be output (monitored) at the VM45/P176 pin via control by the VM45C register.

After reset: 00H R/W Address: FFFFF07CH

	7	6	5	4	3	2	1	0
POCS	0	0	0	0	0	0	VM45C1	VM45C0

VM45C1	VM45 (V_{DD0} 4.5 V monitor) output enabled/disabled
0	VM45 output at VM45/P176 pin disabled (port function)
1	VM45 output at VM45/P176 pin enabled[Note]

VM45C0	VM45 (V_{DD0} 4.5 V monitor) output selection
0	High-level output when VM45 detected
1	Low-level output when VM45 detected

Note When using P176 as an alternate function pin, it is necessary to set the PM176 bit of the port 17 mode register (PM17) to 0 (output mode), or the P176 bit of port 17 (P17) to 0 (0 output).

★ **(3) POC control register (POCC)**

This register sets whether the 3.5 V power-on-clear reset detection voltage is enabled disabled.

However, detection of less than 2.5 V in STOP mode cannot be disabled.

Reset by power-on-clear when the initial power supply is applied is enabled, and reset by power-on-clear caused by a subsequent voltage drop is prohibited. Reset applied through the $\overline{\text{RESET}}$ pin clears POCC to 00H.

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

After reset: 00H R/W Address: FFFFF076H

	7	6	5	4	3	2	1	0
POCC	0	0	0	0	0	0	0	POCC0

POCC0	3.5 V power-on-clear reset detection voltage enabled/disabled
0	Enabled (3.5 V power-on-clear reset detection voltage is valid)
1	Disabled (3.5 V power-on-clear reset detection voltage is invalid)

CHAPTER 15 REGULATOR

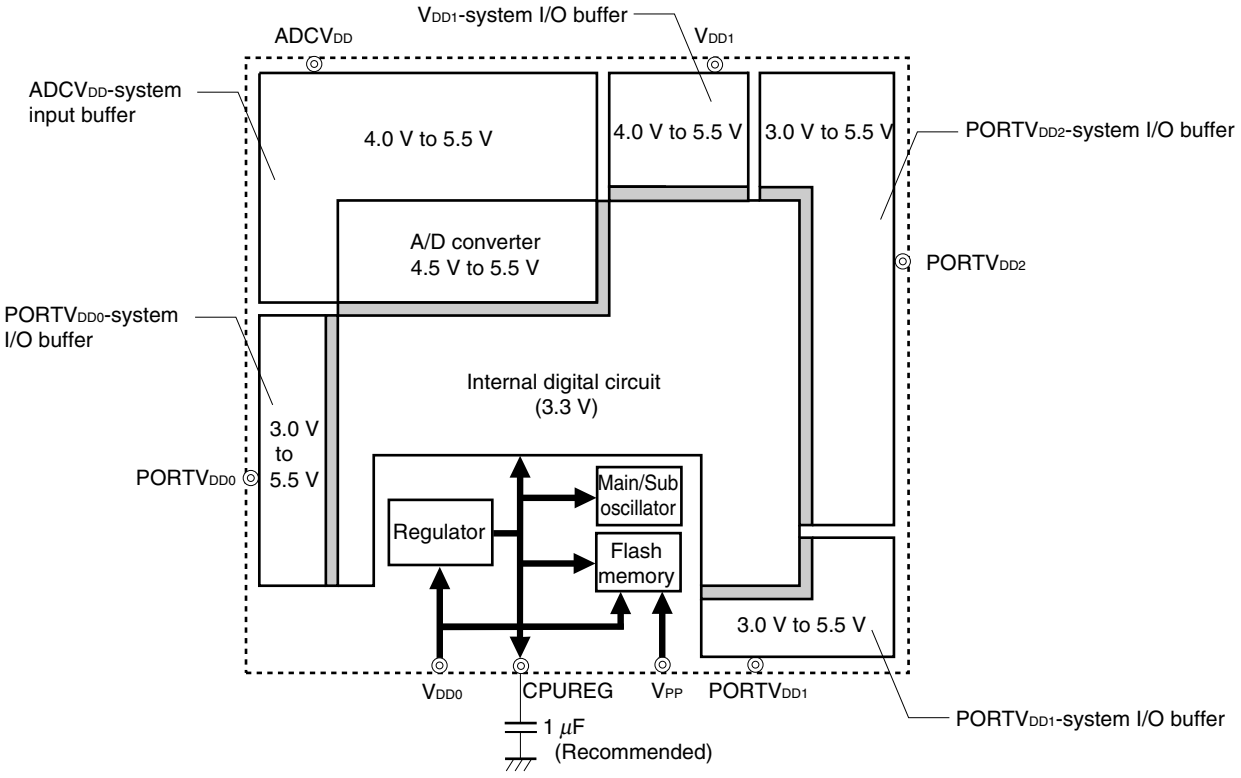
15.1 Outline

The V850/SC1, V850/SC2, and V850/SC3 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 oscillation block and on-chip logic circuits (excluding the A/D converter and output buffers). The regulator output voltage is set to 3.3 V.

Refer to **2.4 I/O Circuit Types, I/O Buffer Power Supply and Connection of Unused Pins** for the power supply corresponding to each pin.

Figure 15-1. Regulator (μPD70F3089Y)



Remark : Bidirectional level shifter

15.2 Operation

The regulator of the V850/SC1, V850/SC2, and V850/SC3 operates in every mode (STOP, IDLE, HALT).

For stabilization of regulator outputs, connect a capacitor of about 1 μF (recommended value) to the CPUREG pin.

CHAPTER 16 ROM CORRECTION FUNCTION

16.1 General

The ROM correction function provided in the V850/SC1, V850/SC2, and V850/SC3 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 (correction address) is replaced with the JMP r0 instruction and 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 (n = 0 to 3).

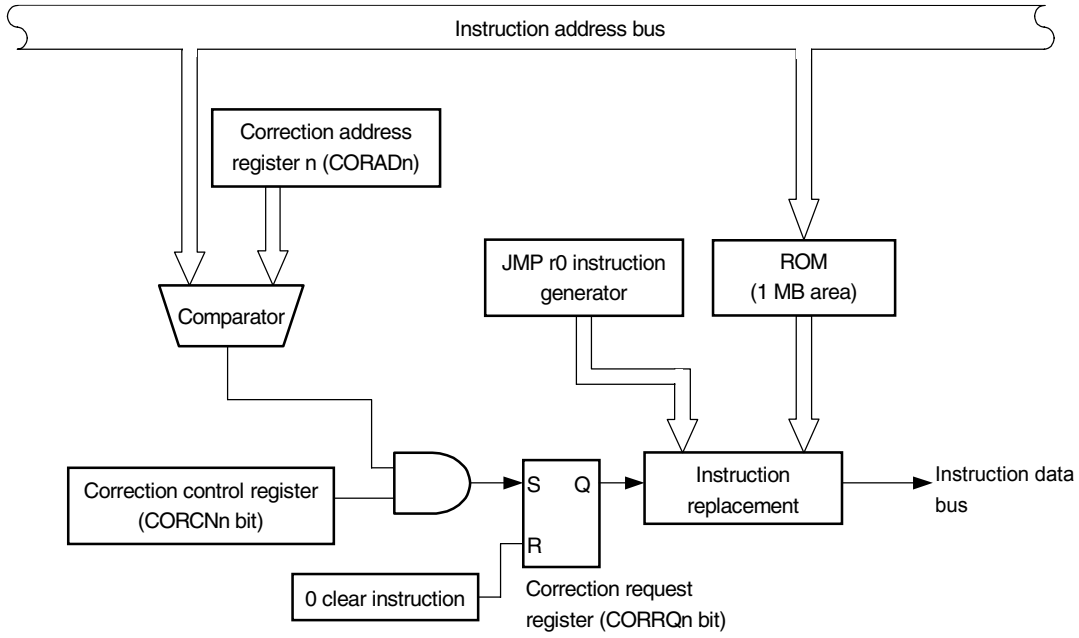
Instruction bugs found in the mask ROM can be avoided, and program flow can be changed by using the ROM correction function.

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 the 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 CORCN, CORRQ, or CORAD0 to CORAD3 registers is prohibited.

Figure 16-1. Block Diagram of ROM Correction



Remark n = 0 to 3

16.2 ROM Correction Peripheral I/O Registers

16.2.1 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 set to the correction address register (CORADn) 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 an 8-bit or 1-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 (n = 0 to 3)
0	Match detection disabled
1	Match detection enabled

Remark n = 0 to 3

16.2.2 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[Note]	Channel n ROM correction request flag
0	No ROM correction request occurred.
1	ROM correction request occurred.

Note The CORRQn bit is cleared by a “write 0” instruction.

Remark n = 0 to 3

16.2.3 Correction address registers 0 to 3 (CORAD0 to CORAD3)

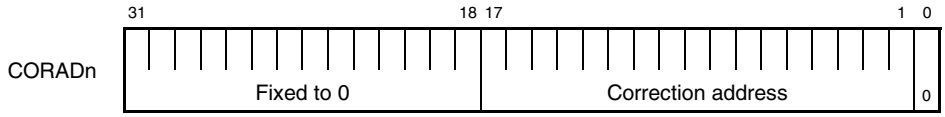
CORADn sets the start address of the instruction to be corrected (correction address) in the ROM.

Up to four points of the program can be corrected at once since the V850/SC1, V850/SC2, and V850/SC3 have four correction address registers (CORADn) (n = 0 to 3).

Set 00000000H to 0007FFFEH since the V850/SC1, V850/SC2, and V850/SC3 incorporate a 512 KB ROM.

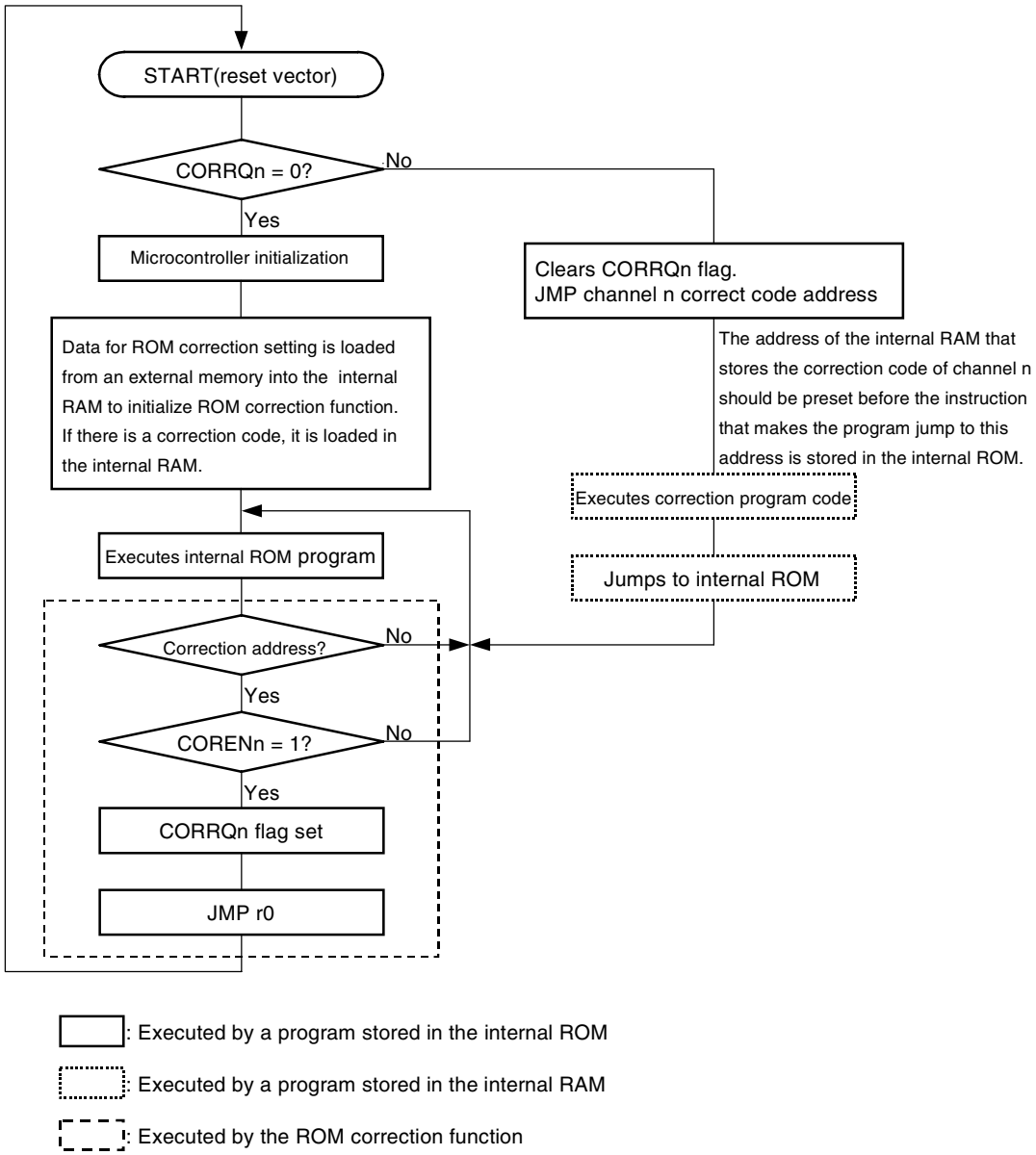
Bits 0 and 18 to 31 should be fixed to 0.

After reset: 00000000H R/W Address: CORAD0: FFFFF370H CORAD2: FFFFF378H
CORAD1: FFFFF374H CORAD3: FFFFF37CH



(n = 0 to 3)

Figure 16-2. ROM Correction Operation and Program Flow



Caution Check the ROM correction generation from a 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 17 FLASH MEMORY (μPD70F3089Y)

The μPD70F3089Y is the flash memory version of the V850/SC1, V850/SC2, and V850/SC3 and incorporates a 512 KB flash memory.

Caution There are differences in the amount of noise tolerance and noise radiation between flash memory versions and mask ROM versions. When considering changing from a flash memory version to a mask ROM version during the process from experimental manufacturing to mass production, make sure to sufficiently evaluate commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.

In the instruction fetch to this flash memory, 4 bytes can be accessed by a single clock as well as the mask ROM version.

Writing to flash memory can be performed with 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 the applications using flash memory.

- Software can be altered after the V850/SC1, V850/SC2, and V850/SC3 are 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.

17.1 Features

- 4-byte/1-clock access (in instruction fetch access)
- All area batch 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 by self rewriting in area (128 KB) units is possible

17.1.1 Erasing unit

This product has the following two erasure units.

(a) All area batch erase

The area of xx000000H to xx07FFFFH can be erased at the same time. The erasure time is 8.0 s.

(b) Area erase

Erasure can be performed in area units (there are four 128 KB unit areas). The erasure time is 2.0 s for each area.

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

17.1.2 Write/read time

The write/read time is shown below.

Write time: 50 μs/byte
Read time: 50 ns (cycle time)

17.2 Writing with Flash Programmer

Writing can be performed either on-board or off-board with the dedicated flash programmer.

(1) On-board programming

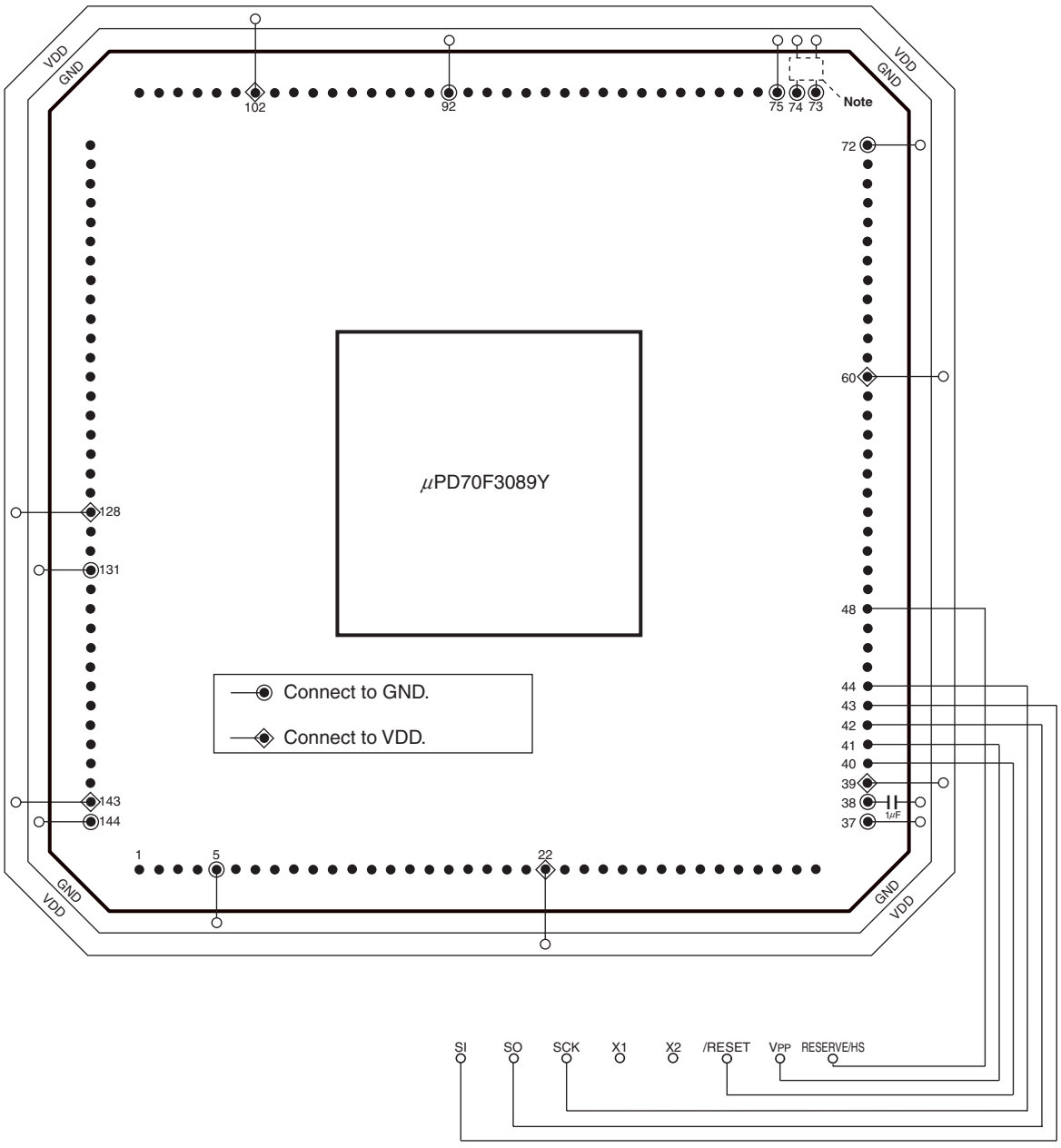
The contents of the flash memory is rewritten after the V850/SC1, V850/SC2, and V850/SC3 are mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

(2) Off-board programming

Writing to a flash memory is performed by the dedicated program adapter (FA Series), etc., before mounting the V850/SC1, V850/SC2, and V850/SC3 on the target system.

Remark The FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

★ **Figure 17-1. Example of Wiring of Adapter for Flash Programming (FA-144GJ-UEN)**

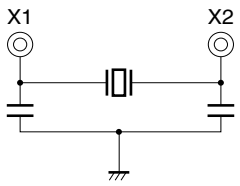


Note The μPD70F3089Y cannot be supplied with the clock from the CLK pin of the flash programmer (PG-FP3).

Supply the clock by creating an oscillator on the flash writing adapter (broken-line portion).

An example of the oscillator is shown below.

Example



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 Supply and Connection of Unused Pins**). When connecting via a resistor, use of a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is for a 144-pin plastic LQFP package.

3. This diagram shows the wiring when using CSI supporting handshake.

★ **Table 17-1. Table for Wiring of Adapter for μPD70F3089Y Flash Programming (FA-144GJ-UEN)**

Flash Programmer (PG-FP3)			When Using CSI0 + HS		When Using CSI0		When Using UART0	
			Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	P11/SO0	43	P11/SO0	43	P14/SO4/TXD0	47
SO/TxD	Output	Transmit signal	P10/SI0/SDA0	42	P10/SI0/SDA0	42	P13/SI4/RXD0	46
SCK	Output	Transfer clock	P12/$\overline{\text{SCK0}}$/SCL0	44	P12/$\overline{\text{SCK0}}$/SCL0	44	Unnecessary	Unnecessary
CLK[Note]	–	Unused	Unnecessary	Unnecessary	Unnecessary	Unnecessary	Unnecessary	Unnecessary
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	40	$\overline{\text{RESET}}$	40	$\overline{\text{RESET}}$	40
VPP	Output	Writing voltage	MODE/V_{PP}	41	MODE/V_{PP}	41	MODE/V_{PP}	41
HS	Input	Handshake signal of CSI0 + HS communication	P15/$\overline{\text{SCK4}}$/ASCK0	48	Unnecessary	Unnecessary	Unnecessary	Unnecessary
VDD	–	VDD voltage generation/power supply monitoring	V_{DD0}	39	V_{DD0}	39	V_{DD0}	39
			V_{DD1}	128	V_{DD1}	128	V_{DD1}	128
			$PORTV_{DD0}$	22	$PORTV_{DD0}$	22	$PORTV_{DD0}$	22
			$PORTV_{DD1}$	60	$PORTV_{DD1}$	60	$PORTV_{DD1}$	60
			$PORTV_{DD2}$	102	$PORTV_{DD2}$	102	$PORTV_{DD2}$	102
GND	–	Ground	GND0	37	GND0	37	GND0	37
			GND1	131	GND1	131	GND1	131
			GND2	72	GND2	72	GND2	72
			PORTGND0	5	PORTGND0	5	PORTGND0	5
			PORTGND1	92	PORTGND1	92	PORTGND1	92
			P00/NMI	75	P00/NMI	75	P00/NMI	75

Note The μPD70F3089Y cannot be supplied with the clock from the CLK pin of the flash programmer (PG-FP3).

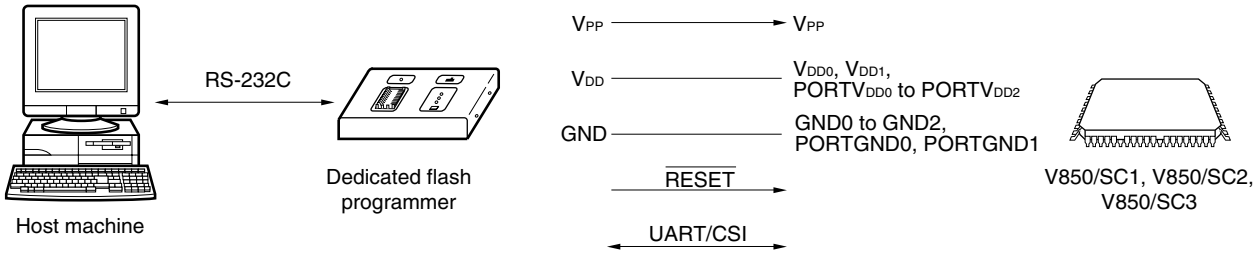
Supply the clock by creating an oscillator on the flash writing adapter (FA-144GJ-UEN).

For an example of the oscillator, refer to **Figure 17-1 Example of Wiring of Adapter for Flash Programming (FA-144GJ-UEN)**.

17.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850/SC1, V850/SC2, and V850/SC3.

Figure 17-2. 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/SC1, V850/SC2, and V850/SC3 to perform writing, erasing, etc. A dedicated program adapter (FA Series) is required for off-board writing.

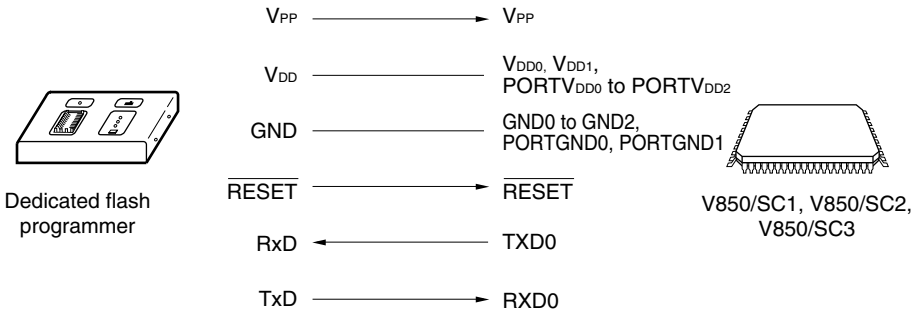
17.4 Communication Mode

Communication between the dedicated flash programmer and the V850/SC1, V850/SC2, and V850/SC3 is serial communication performed using UART0 or CSI0 of the V850/SC1, V850/SC2, and V850/SC3.

(1) UART0

Transfer rate: 4800 to 76800 bps

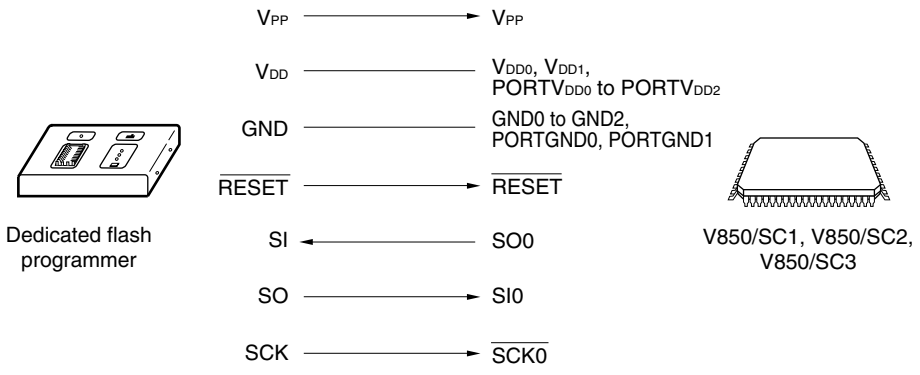
Figure 17-3. Communication with Dedicated Flash Programmer (UART0)



(2) CSI0

Serial clock: Up to 1 MHz (MSB first)

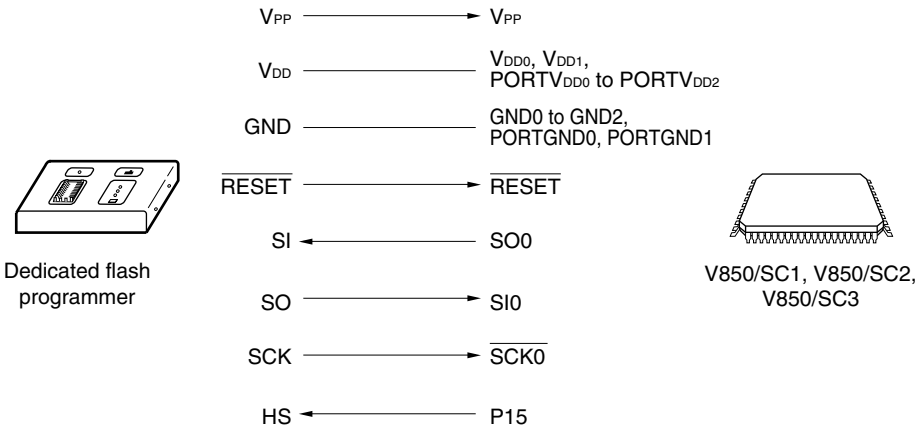
Figure 17-4. Communication with Dedicated Flash Programmer (CSI0)



(3) CSI0 + HS

Serial clock: Up to 1 MHz (MSB first)

Figure 17-5. Communication with Dedicated Flash Programmer (CSI0 + HS)



The dedicated flash programmer outputs the transfer clock, and the V850/SC1, V850/SC2, and V850/SC3 operate as a slave.

When the PG-FP3 is used as the dedicated flash programmer, it generates the following signals to the V850/SC1, V850/SC2, and V850/SC3. For the details, refer to the **PG-FP3 User's Manual**.

Table 17-2. Signal Generation of Dedicated Flash Programmer (PG-FP3)

PG-FP3			V850/SCx	Connection		
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_{DD0}, V_{DD1}, $PORTV_{DD0}$ to $PORTV_{DD2}$	◎	◎	◎
GND	–	Ground	GND0 to GND2, PORTGND0, PORTGND1	◎	◎	◎
CLK[Note]	–	Unused	X1	×	×	×
$\overline{RESET}$	Output	Reset signal	$\overline{RESET}$	◎	◎	◎
SI/RxD	Input	Receive signal	SO0/TXD0	◎	◎	◎
SO/TxD	Output	Transmit signal	SI0/RXD0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{SCK0}$	◎	×	◎
HS	Input	Handshake signal of CSI0 + HS	P15	×	×	◎

Note The μPD70F3089Y 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 (FA-144GJ-UEN).

For an example of the oscillator, refer to **Figure 17-1 Example of Wiring of Adapter for Flash Programming (FA-144GJ-UEN)**.

Remarks 1. ◎: Always connected

×: Does not need to be connected

2. V850/SCx: V850/SC1, V850/SC2, V850/SC3

17.5 Pin Connection

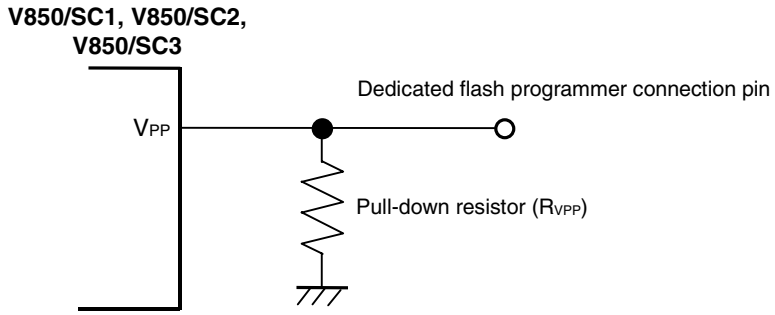
When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, install a function on-board to switch from the normal operating mode to the flash memory programming mode.

When switched to the flash memory programming mode, all the pins not used for the flash memory programming become the same status as that immediately after reset. Therefore, all the ports become output high-impedance status, so that pin handling is required when the external device does not acknowledge the output high-impedance status.

17.5.1 V_{PP} pin

In the normal operating mode, 0 V is input to V_{PP} pin. In the flash memory programming mode, a 7.8 V writing voltage is supplied to V_{PP} pin. The following shows an example of the connection of the V_{PP} pin.

Figure 17-6. V_{PP} Pin Connection Example



17.5.2 Serial interface pin

The following shows the pins used by each serial interface.

Table 17-3. Pins Used by Serial Interfaces

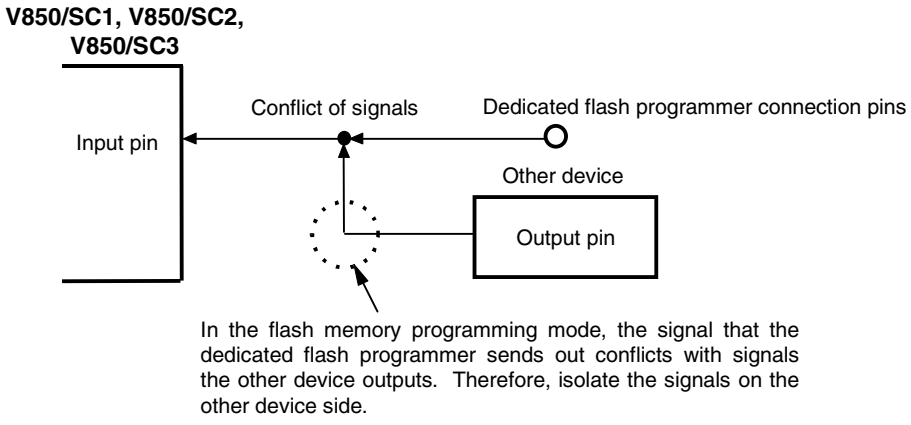
Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$
CSI0 + HS	SO0, SI0, $\overline{SCK0}$, P15
UART0	TXD0, RXD0

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 the conflict of signals and the malfunction of 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.

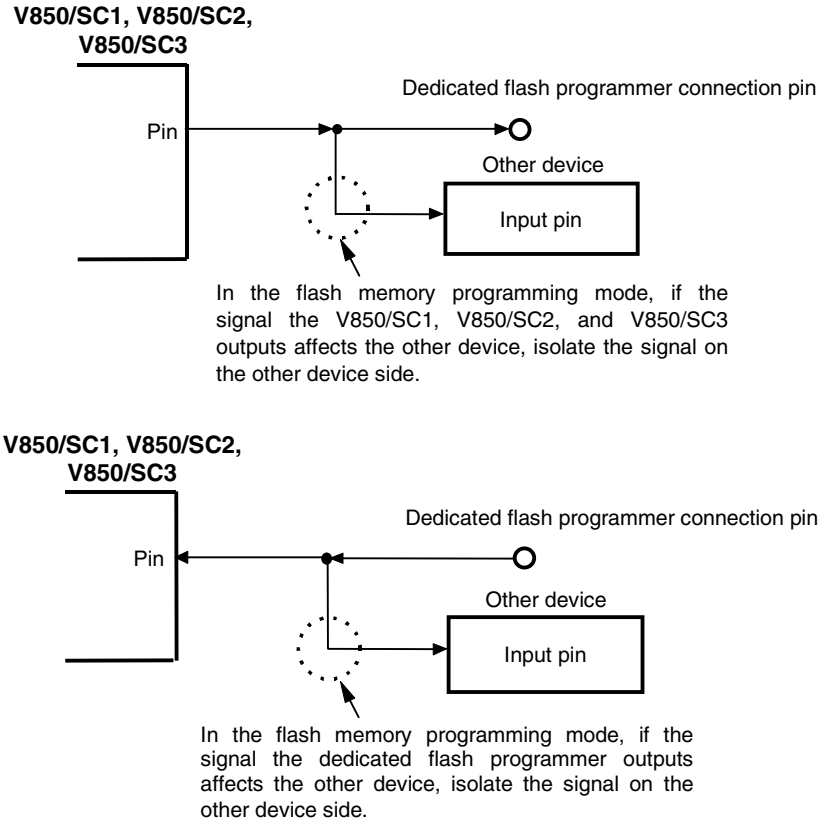
Figure 17-7. Conflict of Signals (Serial Interface Input Pin)



(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 set so that the input signal to the other device is ignored.

Figure 17-8. Malfunction of Other Device

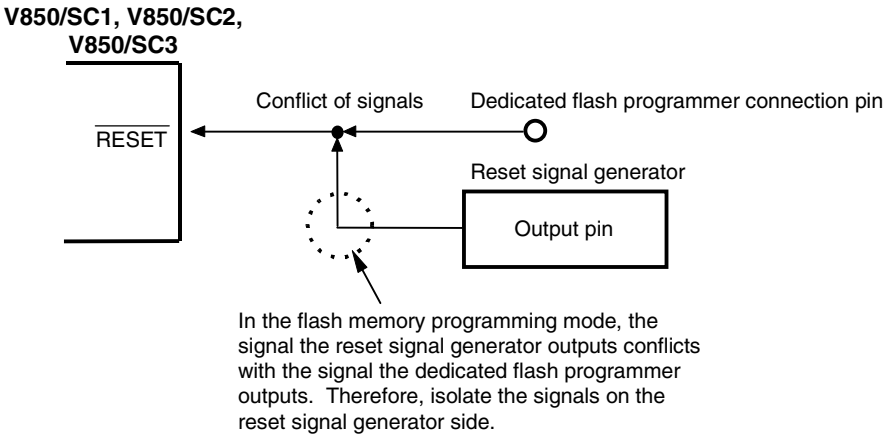


17.5.3 $\overline{\text{RESET}}$ pin

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, programming operations will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

Figure 17-9. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



17.5.4 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 become output high-impedance status. If problems such as disabling output high-impedance status should occur to the external devices connected to the port, connect them to V_{DD0}, V_{DD1}, $PORTV_{DD0}$ to $PORTV_{DD2}$, $ADCV_{DD}$, GND0 to GND2, PORTGND0, PORTGND1, or ADCGND via resistors.

17.5.5 Other signal pins

Connect X1, X2, XT1, and XT2 in the same status as that in the normal operating mode.

17.5.6 Power supply

Supply the power supply as follows:

$V_{DD0} = PORTV_{DD1}$

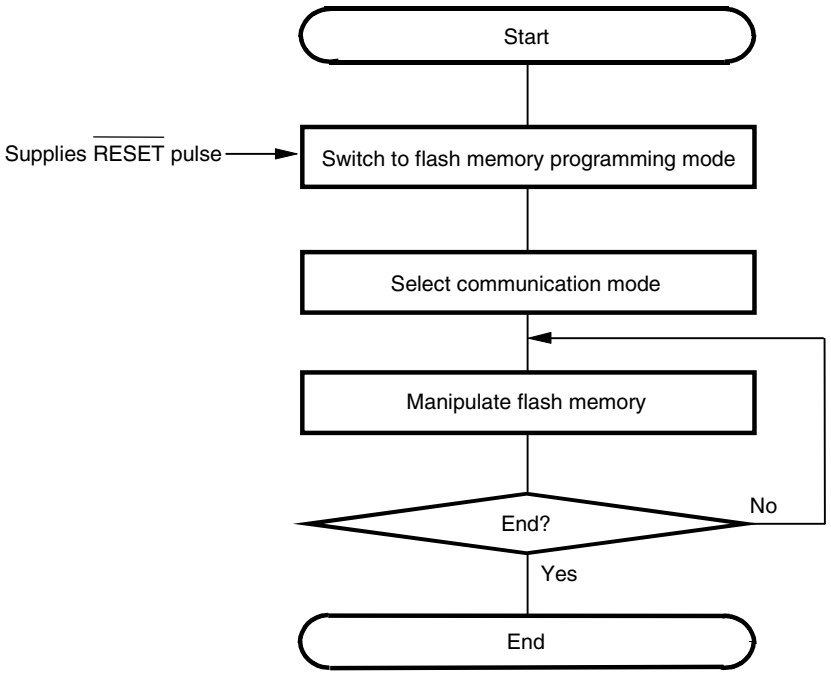
Supply the power (V_{DD1}, $PORTV_{DD0}$, $PORTV_{DD2}$, $ADCV_{DD}$, ADCGND, GND0 to GND2, PORTGND0, and PORTGND1) the same as when in normal operating mode.

17.6 Programming Method

17.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 17-10. Procedure for Manipulating Flash Memory

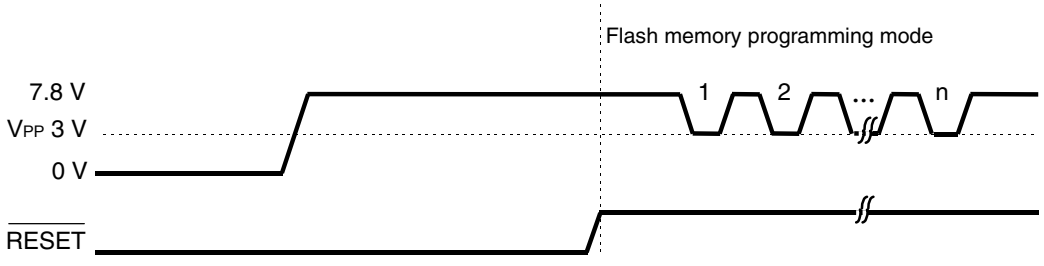


17.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850/SC1, V850/SC2, and V850/SC3 in the flash memory programming mode. When switching modes, set the V_{PP} pin before canceling reset.

When performing on-board writing, switch modes using a jumper, etc.

Figure 17-11. Flash Memory Programming Mode



V_{PP}	Operation mode
0 V	Normal operation mode
7.8 V	Flash memory programming mode

17.6.3 Selection of communication mode

In the V850/SC1, V850/SC2, and V850/SC3, the communication mode is selected by inputting pulses (16 pulses max.) to 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 17-4. List of Communication Modes

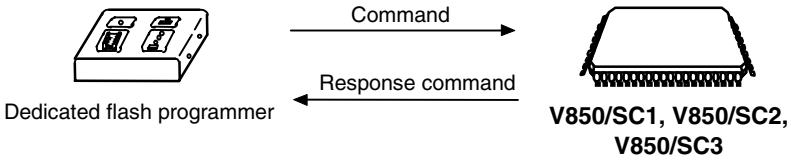
V_{PP} Pulses	Communication Mode	Remarks
0	CSI0	V850/SC1, V850/SC2, and V850/SC3 perform slave operation, MSB first
3	CSI0 + HS	V850/SC1, V850/SC2, and V850/SC3 perform slave operation, MSB first
8	UART0	Communication rate: 9600 bps (at reset), LSB first
Other	RFU	Setting prohibited

Caution When UART0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the V_{PP} pulse.

17.6.4 Communication command

The V850/SC1, V850/SC2, and V850/SC3 communicate with the dedicated flash programmer by means of commands. The command sent from the dedicated flash programmer to the V850/SC1, V850/SC2, and V850/SC3 is called the “command”. The response signal sent from the V850/SC1, V850/SC2, and V850/SC3 to the dedicated flash programmer is called the “response command”.

Figure 17-12. Communication Command



The following shows the commands for flash memory control of the V850/SC1, V850/SC2, and V850/SC3. All of these commands are issued from the dedicated flash programmer, and the V850/SC1, V850/SC2, and V850/SC3 perform the various processings corresponding to the commands.

Table 17-5. Flash Memory Control Commands

Category	Command Name	Function
Verify	Batch verify command	Compares the contents of the entire memory and the input data.
Erase	Batch erase command	Erases the contents of the entire memory.
	Write back command	Writes back the contents which is overerased.
Blank check	Batch blank check command	Checks the erase state of the entire memory.
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 verify check.
	Continuous write command	Writes data from the address following the high-speed write command executed immediately before, and executes verify check.
System setting and control	Status read out command	Acquires the status of operations.
	Oscillating frequency setting command	Sets the oscillating frequency.
	Erasure time setting command	Sets the erasing time of batch erase.
	Write time setting command	Sets the writing time of data write.
	Write back time setting command	Sets the write back time.
	Baud rate setting command	Sets the baud rate when using UART.
	Silicon signature command	Reads outs the silicon signature information.
	Reset command	Escapes from each state.

The V850/SC1, V850/SC2, and V850/SC3 send back response commands to the commands issued from the dedicated flash programmer. The following shows the response commands the V850/SC1, V850/SC2, and V850/SC3 send out.

Table 17-6. Response Commands

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

CHAPTER 18 IEBus CONTROLLER (V850/SC2)

The IEBus (Inter Equipment Bus) is a small-scale digital data transfer system that transfers data between units. To implement the IEBus with the V850/SC2, an external IEBus driver and receiver are necessary because they are not provided.

The internal IEBus controller of the V850/SC2 is of negative logic.

18.1 IEBus Controller Function

18.1.1 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) Broadcast communication function

Communication between one unit and plural units can be performed as follows:

- Group-unit broadcast communication: Broadcast communication to group units
- All-unit broadcast communication: Broadcast communication to all units.

(3) Effective transfer rate

The effective transfer rate is in mode 1 (the V850/SC2 does not support modes 0 and 2 of 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 a 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> Broadcast 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.

18.1.2 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 grants 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 condition of the bus is predetermined as follows:

Caution The bus mastership is released if communication is aborted.

(1) Priority by communication type

Broadcast 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.

18.1.3 Communication mode

Although the IEBus has three communication modes each having a different transfer rate, the V850/SC2 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 18-1.

Table 18-1. 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 mating unit (slave unit) are not the same, communication is not correctly executed.

18.1.4 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)

18.1.5 Broadcast communication

Normally, transmission or reception is performed between the master unit and its mating slave unit on a one-to-one basis. During broadcast 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 broadcast communication or normal communication is to be executed is selected by the broadcast bit (for this bit, refer to **18.1.6 (2) Broadcast bit**).

Broadcast communication can be classified into the two types; group-unit broadcast communication and all-unit broadcast communication. Group-unit broadcast communication and all-unit broadcast communication are identified by the value of the slave address (for the slave address, refer to **18.1.6 (4) Slave address field**).

(1) Group-unit broadcast communication

Broadcast communication is performed to the units in a group identified by the group number indicated by the higher 4 bits of the communication address.

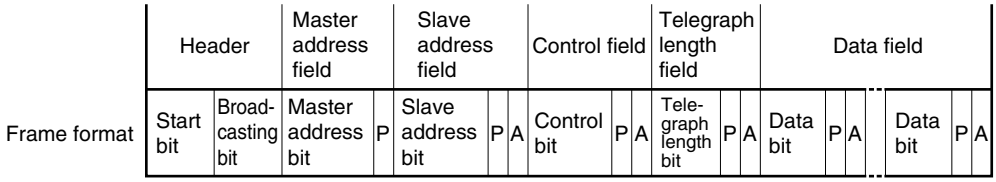
(2) All-unit broadcast communication

Broadcast communication is performed to all the units, regardless of the value of the group number.

18.1.6 Transfer format of IEBus

Figure 18-1 shows the transfer signal format of the IEBus.

Figure 18-1. IEBus Transfer Signal Format



Remarks 1. P: Parity bit, A: $\overline{\text{ACK}}$/NACK bit

2. The master station ignores the acknowledge bit during broadcast 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 $\overline{\text{IETX0}}$ pin for a specific time, and then starts outputting the broadcast bit.

If another unit has already output its start bit when one unit attempts 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 broadcast 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) Broadcast bit

This bit indicates whether the master selects one slave (individual communication) or plural slaves (broadcast communication) as the other party of communication.

When the broadcast bit is 0, it indicates broadcast communication; when it is 1, individual communication is indicated. Broadcast 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 **18.1.6 (4) Slave address field**).

Because two or more slave units exist in the case of broadcast 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, broadcast communication takes precedence over individual communication, and wins in arbitration.

If one station occupies the bus as the master, the value set to the broadcast 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 18-2.

If two or more units start transmitting the broadcast 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.

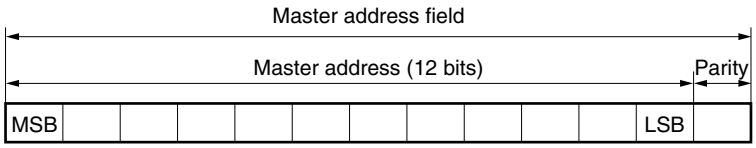
Because the IEBus is configured of wired AND, the unit having the minimum master address of the units participating in arbitration (arbitration masters) wins in arbitration.

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 the 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 18-2. Master Address Field



(4) Slave address field

The master outputs the address of the unit with which it is to communicate.

Figure 18-3 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 broadcast 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.

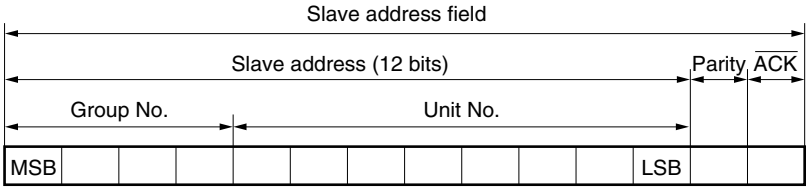
During broadcast communication, the slave address is used to identify group-unit broadcast or all-unit broadcast, as follows:

If slave address is FFFH:	All-unit broadcast communication
If slave address is other than FFFH:	Group-unit broadcast communication

Remark The group No. during group-unit broadcast 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 18-3. 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 18-4.

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 length 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 broadcast communication, however, the master unit does not confirm the acknowledge signal, and starts outputting the telegraph length field.

Table 18-2 shows the contents of the control bits.

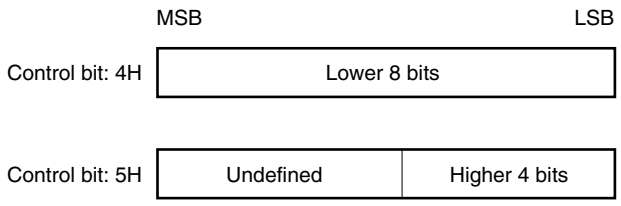
Table 18-2. Contents of Control Bits

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 **18.1.7 (4) Locking and unlocking**).

3. The lock address is transferred in 1-byte (8-bit) units and is configured as follows:



If the control bit received from the master unit is not as shown in Table 18-3, the unit locked by the master unit reject accepting the control bit, and does not output the acknowledge bit.

Table 18-3. 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)

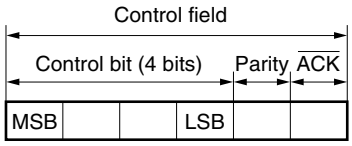
Moreover, units for which lock is not set by the master unit reject acknowledgement and do not output an acknowledge bit when the control data shown in Table 18-4 is acknowledged.

Table 18-4. 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 18-4. Control Field



★ **Table 18-5. Control Field Acknowledge Signal Output Conditions**

(a) When received control data is AH, BH, EH, FH

Communication type (ALL TRANS) Individual communication = 1 Broadcast communication = 1	Communication target (SLVRQ) Slave specification = 1 No specification = 0	Lock status (LOCK) Locked = 1 Unlocked = 0	Master unit judgment (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) When received control data is 0H, 3H, 4H, 5H, 6H, 7H

Communication type (ALL TRANS) Individual communication = 1 Broadcast communication = 1	Communication target (SLVRQ) Slave specification = 1 No specification = 0	Lock status (LOCK) Locked = 1 Unlocked = 0	Master unit judgment (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 as shown in Table 18-5, it becomes × unconditionally ($\overline{\text{ACK}}$ is not returned).

Remarks 1. ○: $\overline{\text{ACK}}$ returned
×: $\overline{\text{ACK}}$ not returned

2. ENSLVTX: Bit 4 of IEBus unit control register (BCR)
ENSLVRX: Bit 3 of IEBus unit control register (BCR)
SLVRQ: Bit 6 of IEBus unit status register (USR)
LOCK: Bit 2 of 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 18-5.

Table 18-6 shows the relationship between the telegraph length bit and the number of transmit data.

Figure 18-5. Telegraph Length Field

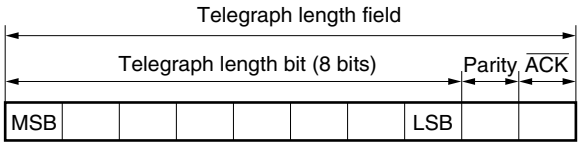


Table 18-6. Contents of Telegraph Length Bit

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 (when control bit 3 is 1) or receives (when control bit 3 is 0) data.

(a) When master transmits data

The telegraph length bit and parity bit 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 broadcast 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.

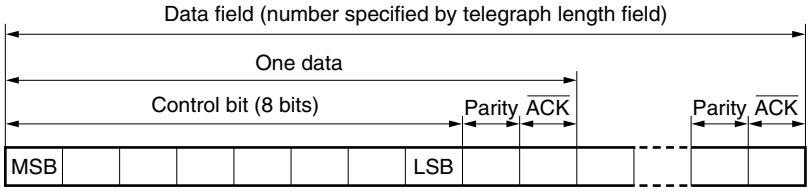
(7) Data field

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 18-6. Data Field



Following the data bit, the parity bit and acknowledge bit are respectively output by the master unit and slave unit.

Use broadcast 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 broadcast 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 broadcast 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 a master reception in broadcast communication, because the slave unit cannot be defined and a 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 to see 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 broadcast communication, however, the content of the acknowledge bit is 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) (refer to **18.3.2 (1) IEBus Control Register (BCR)**)
- If the control bit indicates reading of data (3H or 7H) when the slave transmission enable flag (ENSLVTX) is not set (1) (refer to **18.3.2 (1) IEBus Control Register (BCR)**)
- 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

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 broadcast 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 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 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 proceeding 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 broadcast communication, the transmission side does not execute transmission again, a communication error occurs on the reception side and reception stops.

18.1.7 Transfer data

(1) Slave status

The master unit can ascertain why the slave unit did not return the acknowledge bit ($\overline{\text{ACK}}$) by reading the slave status.

The slave status is determined depending on 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 18-7. 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.

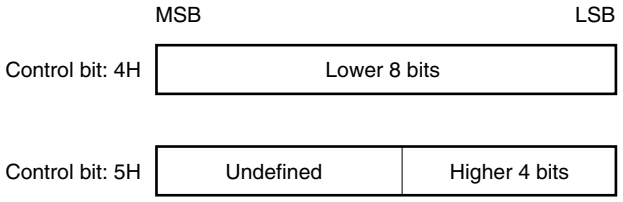
3. When the V850/SC2 serves as a slave unit, this bit corresponds to the status indicated by bit 4 (ENSLVTX) of the IEBus control register (BCR).

4. When the V850/SC2 serves as a slave unit, bits 7 and 6 are both fixed to mode 1 (bits 7, 6 = 0, 1).

(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 18-8. Configuration of Lock Address



(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 broadcast communication).

A unit is locked or unlocked as follows:

(a) Locking

If the communication frame is completed without succeeding in transmission or reception of the 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 broadcast communication.

Locking and unlocking conditions are shown below.

(c) Lock setting conditions

Control Data	Broadcast 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) Lock release conditions (while locked)

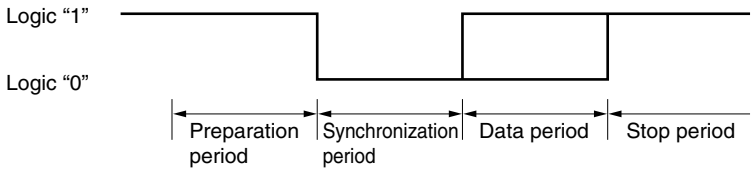
Control Data	Broadcast Communication from Lock Request Unit		Individual Communication from Lock Request Unit	
	Communication End	Frame End	Communication End	Frame End
3H, 6H[Note]			Unlocked	Remains locked
AH, BH	Unlocked	Unlocked	Unlocked	Remains locked
0H, 4H, 5H, EH, FH	Remains locked	Remains locked	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.

18.1.8 Bit format

The format of the bits constituting the communication frame of the IEBus is shown below.

Figure 18-9. 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

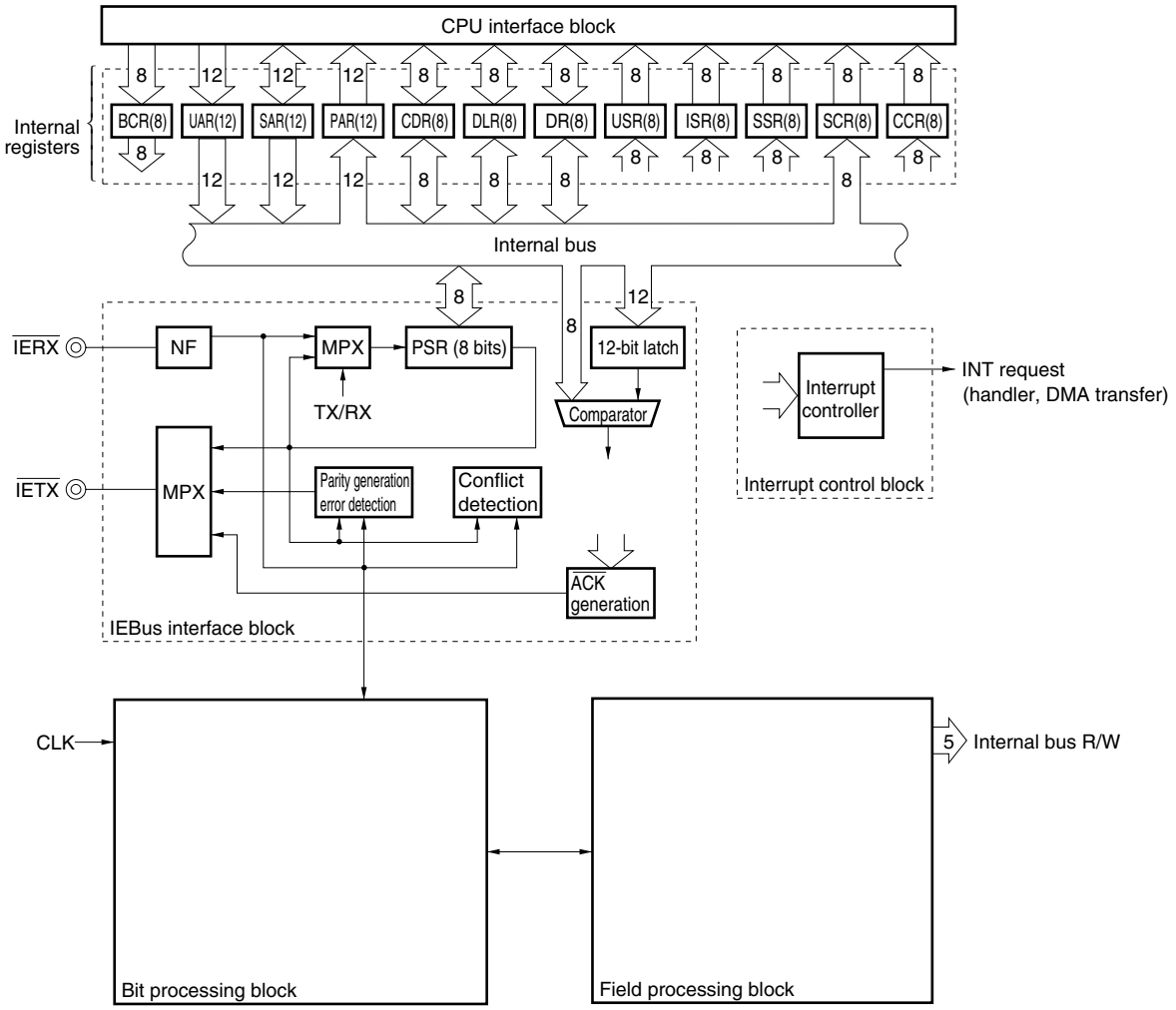
The synchronization period and data period are almost equal to each other in length.

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. During communication, the master and slave units detect whether each period (preparation period, synchronization period, data period, and stop period) is output for the specified time. If not, the master and slave units regard it as a timing error, and immediately stop communication and return to the wait status.

18.2 IEBus Controller Configuration

The block diagram of the IEBus controller is shown below.

Figure 18-10. 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/SC2) and the IEBus.

(b) Interrupt control block

This control block transfers interrupt request signals from the IEBus to the CPU.

(c) Internal registers

These registers set data to the control registers and fields that control the IEBus (for the internal registers, refer to **18.3 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, conflict detector, parity detector, parity generator, and $\overline{\text{ACK}}$/NACK generator.

18.3 Internal Registers of IEBus Controller

18.3.1 Internal register list

Table 18-7. 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 select register	IECLK	R/W	–	√	–	00H

18.3.2 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	Broadcast 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 acknowledgement 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 Make both of the following settings before setting the ENIEBUS flag.

- **Set the interrupt enable (EI) status and enable interrupt servicing of INTIE2 (IEBMK = 2).**
- **Set the IEBus unit address register (UAR)**

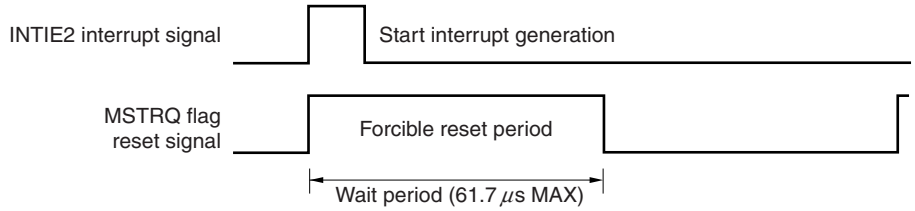
(b) Master request flag (MSTRQ)...Bit 6

<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 broadcast 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 while 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) while 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 broadcast communication. In individual communication, however, when the received slave address matches the local address, the start interrupt 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 the IEBus unit. This register must always be 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 always be 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 broadcast 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 broadcast 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 (control data: 0H, 6H) or a lock address request (4H, 5H), whether 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 acknowledgement 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 18-11. Interrupt Generation Timing (for (1), (3), and (4))

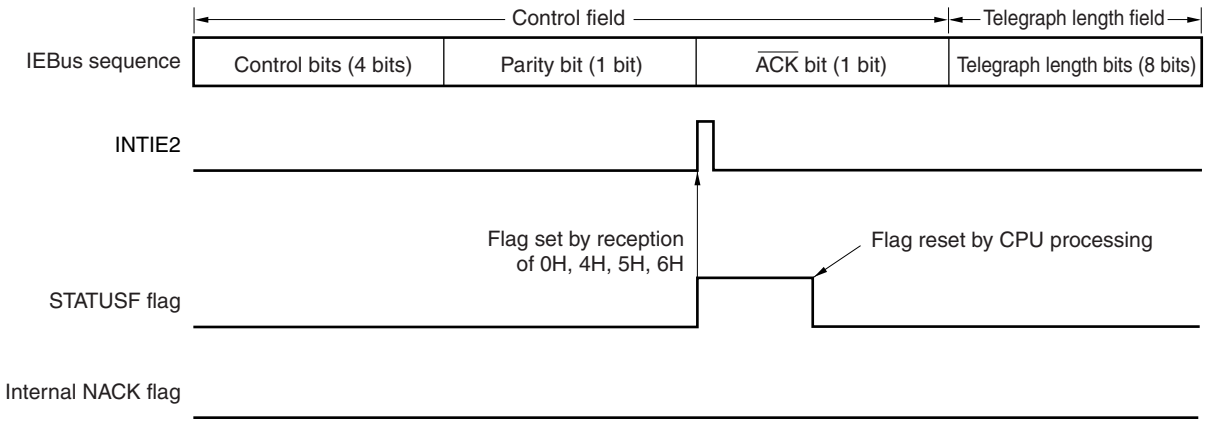
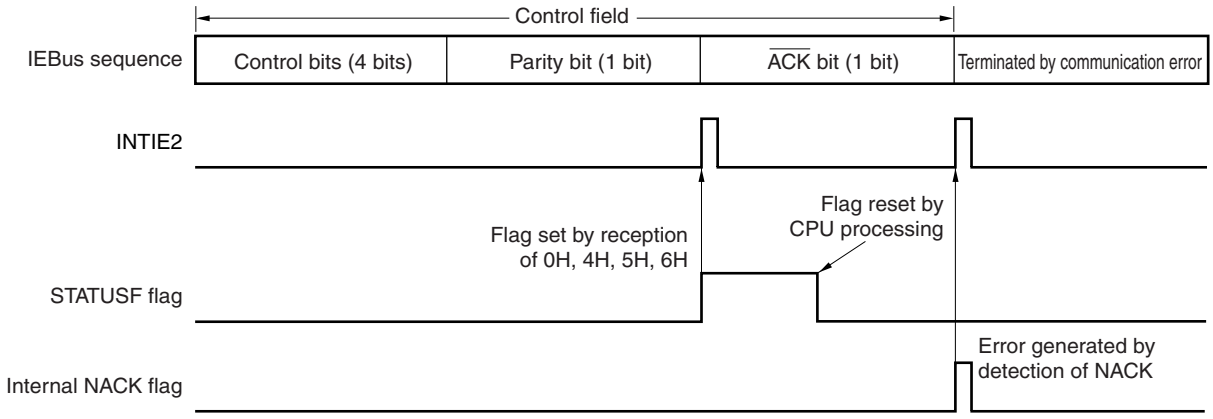
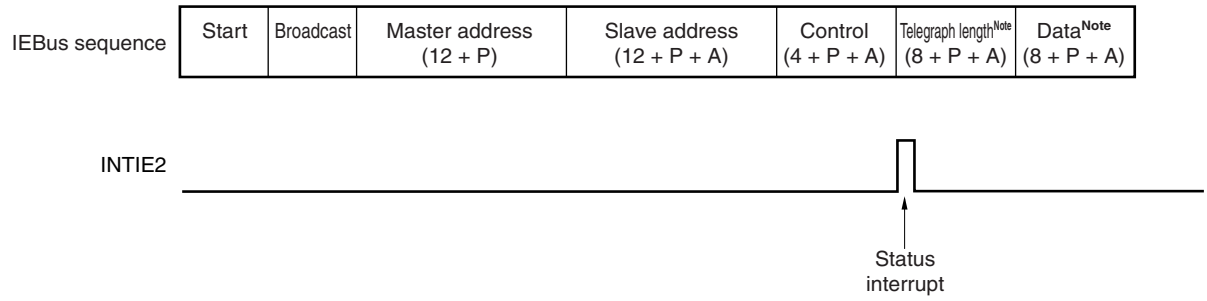


Figure 18-12. Interrupt Generation Timing (for (2) and (5))



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.

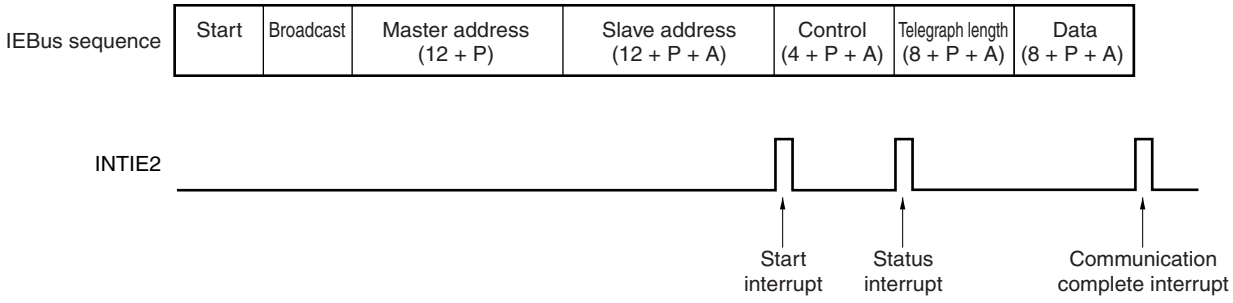
Figure 18-13. Timing of INTIE2 Interrupt Generation in Locked State (for (4) and (5))



Note For (5), there is no $\overline{\text{ACK}}$ return and therefore no transition to telegraph length and data.

Remark P: Parity bit, A: $\overline{\text{ACK}}$/NACK bit

Figure 18-14. 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 is divided into a writing side and a reading side, making it impossible for the written data to be readout as is. Only data received via IEBus communication can be readout.

After reset: 01H R/W Address: FFFFF3EAH

	7	6	5	4	3	2	1	0
DLR								

Bit								Setting value	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 (8 bits) to bits 7 to 0.

Remark The IEBus telegraph length register is divided into a writing side and a reading side, making it impossible for the written data to be readout as is. Only data received via IEBus communication can be readout.

(a) When transmission unit

The data (1 byte) written to the IEBus data register (DR) is stored in 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. In individual communication, however, if NACK is received after 1-byte data transmission, the transfer from DR to the shift register is not performed and the same data is transmitted again. In this case, 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 to stop 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 broadcast 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	Broadcast communication flag
0	Individual communication status
1	Broadcast 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

This flag indicates whether there was a slave request from the master.

<Set/reset conditions>

Set: When the IEBus unit has been sent a slave request (during individual communication reception, when the received slave address and local UAR match; during broadcast reception, when the higher 4 bits of the received slave address match, or the received slave address is FFFH), the flag is set by hardware at the start of the acknowledge period for the slave address field.

Reset: When the IEBUS unit is not sent a slave request, the flag is reset by hardware. The timing is the same as for Set. However, if immediately following normal reception of a communication (SLVRQ bit is set), the IEBus unit is sent a slave request and a parity error occurs in the slave address field for that communication, the flag is not reset.

(b) Contention result flag (ARBIT)...Bit 5

This flag shows the contention result.

<Set/reset conditions>

Set: This flag is set if, following a master request, the data output by the IEBus unit during the arbitration period and the bus line data do not match.

Reset: This flag is reset at the start bit timing.

Cautions 1. The reset timing of the contention result flag (ARBIT) differs depending on whether the IEBus unit outputs or does not output a start bit.

- **Start bit output: Reset at output start timing**
- **Start bit not output: Reset at start bit detection timing (approx. 160 μs from output)**

2. If, after a master request has been made, the start bit of another unit is output first and the IEBus unit does not output a start bit, the flag is reset at the start bit detection timing.

(c) Broadcast communication flag (ALLTRNS)...Bit 4

A flag indicating whether the unit is performing broadcast 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.

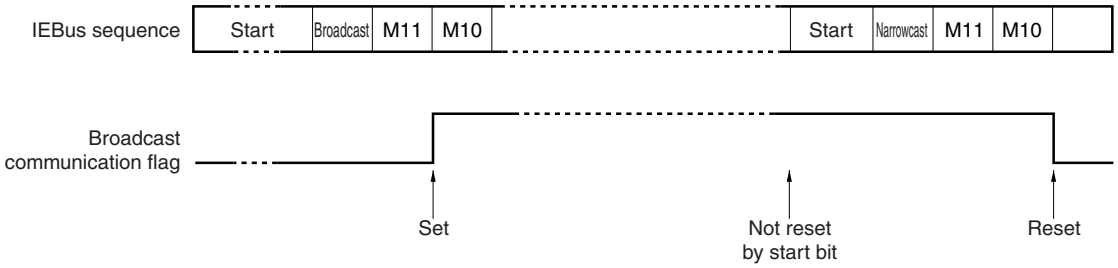
<Set/reset conditions>

Set: When “broadcast” 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 the IEBus is the communication target or not.

Figure 18-15. Example of Broadcast Communication Flag Operation



(d) $\overline{\text{ACK}}$ transmission flag (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 broadcast communication. Also, in locked status, individual communication from a unit other than the unit which has requested to be locked is not acknowledged. However, even communication from a unit which has not requested to be locked can be acknowledged as long as the communication is a slave status request.

(9) IEBus interrupt status register (ISR)

This register indicates the status when the IEBus issues an interrupt. The ISR is read to generate an interrupt, after which the specified interrupt servicing 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 18-8.

Table 18-8. 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 not generated
1	Start interrupt generated

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 18-16). 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 source flag of the generated interrupt before the next interrupt generation timing using an interrupt servicing program.

(a) Communication error flag (IEERR)...Bit 6

A flag that indicates the detection of an error during communication.

<Set/reset conditions>

Set: When a timing error, parity error (except in the data field), NACK reception (except for data field), underrun error, or overrun error (which occurs in broadcast communication) 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 the 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 the 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 the 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 broadcast 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: Occurs when NACK is received in the $\overline{\text{ACK}}$ period in the slave address, control, or telegraph length field, during individual communication regardless of master or slave unit.

  A NACK reception error only occurs in individual communication. $\overline{\text{ACK}}$ and NACK are not discriminated in broadcast 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: When the IEBus is a receiving unit, the IEBus data register (DR) is read by DMA or software after the generation of the interrupt (INTIE1) that causes data to be stored in 1-byte units in this register. 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 acknowledgement 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 broadcast 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 broadcast 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, transmit after releasing the overrun status.

(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 during execution of a communication error interrupt servicing program, etc.

(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, because the telegraph length is automatically set to “01H”, 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 “01” (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

This flag reflects the contents of the slave transmission enable flag.

(b) Lock status flag (STATLOCK)...Bit 2

This flag reflects the contents of the 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 $\overline{ACK}$ 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 bytes (end of communication) or 256 bytes[Note]

Note The bit length of 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 and is decremented during the $\overline{ACK}$ period of the data field regardless of $\overline{ACK}$/NACK. While SCR (IEBus success count register) is decremented upon normal communication ($\overline{ACK}$), CCR is decremented upon each 1-byte communication regardless of $\overline{ACK}$/NACK. When the count value has reached “00H”, the frame end flag (ENDFRAM) of the IEBus interrupt status register (ISR) 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 select register (IECLK)

This register selects the clock of the 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/18.874 MHz

After reset: 00H R/W Address: FFFFF3F8H

	7	6	5	4	3	2	1	0
IECLK	0	0	0	0	0	0	IECS1	IECS0

IECS1	IECS0	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	×	@ f_{XX} = 18.0 MHz or f_{XX} = 18.874 MHz

Remark ×: don’t care

18.4 Interrupt Operations of IEBus Controller

18.4.1 Interrupt control block

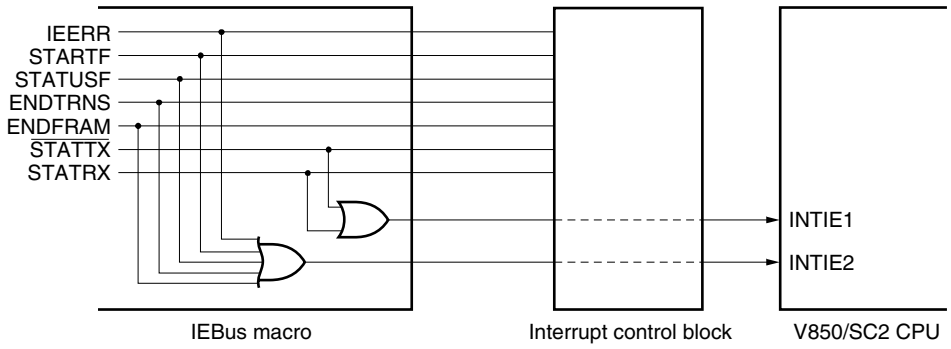
Interrupt request signal

<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 18-9 Interrupt Source List**.

The configuration of the interrupt control block is illustrated below.

Figure 18-16. 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 the V850/SC2.

18.4.2 Interrupt source list

The interrupt request signals of the internal IEBus controller in the V850/SC2 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 18-9. 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 (broadcast)		
	NACK reception	Reception (transmission)	Other than data (individual)		
	Underrun error	Transmission	Data		
	Overrun error	Reception	Data (broadcast)		
Start interrupt		Master	Slave/address	Slave request judgment Contention judgment (If loses, remaster processing) Communication preparation processing	Interrupt always occurs if loss in contention occurs for master request
		Slave	Slave/address	Slave request judgment Communication preparation processing	Generated only for slave request
Status transmission		Slave	Control	Refer to transmission processing example such as slave status.	Interrupt occurs regardless of slave transmission enable flag. Also occurs upon NACK return 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	Transmission data read[Note]	Set after transfer transmission data to internal shift register. Interrupt does not occur at the last data transfer.
Receive data read		Reception	Data	Receive data read[Note]	Set after normal data reception

Note When not using DMA transfer or software manipulation

18.4.3 Communication error cause processing list

The occurrence conditions for communication errors (timing errors, NACK reception errors, overrun errors, underrun errors, and parity errors), the internal IEBus controller error processing contents, and an example of processing by software are described below.

Table 18-10. Communication Error Cause Processing List (1/2)

		Timing Error			
Occurrence condition	Local node status	During reception		During transmission	
	Occurrence condition	When bit timing is off			
	Occurrence location	Other than data field	Data field	Other than data field	Data field
During broadcast communication	Hardware processing	• Reception stopped • INTIE2 generated • To start bit wait status **Remark** Communication between other nodes does not stop.		• Transmission stopped • INTIE2 generated • To start bit wait status	
	Software processing	• Error processing (resend request, etc.)		• Error processing (resend request, etc.)	
During individual communication	Hardware processing	• Reception stopped • INTIE2 generated • NACK returned • To start bit wait status		• Transmission stopped • INTIE2 generated • To start bit wait status	
	Software processing	• Error processing (resend request, etc.)		• Error processing (resend request, etc.)	

		NACK Reception Error				
Occurrence condition	Local node status	During reception		During transmission		
	Occurrence condition	Local node NACK transmission		NACK reception		
	Occurrence location	Other than data field	Data field	Other than data field	Data field	NACK reception for 32nd byte data
During broadcast communication	Hardware processing	–	–	–	–	–
	Software processing	–	–	–	–	–
During individual communication	Hardware processing	• Reception stopped • INTIE2 generated • To start bit wait status	• INTIE2 not generated • Receive data resent by other node	• Transmission stopped • INTIE2 generated • To start bit wait status	• INTIE2 not generated • Resend processing	• INTIE2 generated[Note] • To start bit wait status
	Software processing	• Error processing (resend request, etc.)	–	• Error processing (resend request, etc.)	–	• Error processing (resend request, etc.)

Note Both the IEERR and ENDFRAM bits of the ISR register are set (1).
To reset these bits, follow the conditions specified in Table 18-8.

Table 18-10. Communication Error Cause Processing List (2/2)

		Overrun Error		Underrun Error	
Occurrence condition	Local node status	During reception		During transmission	
	Occurrence condition	DR read is not executed by next data reception timing		DR write is not executed by next data transmission timing	
	Occurrence location	Other than data field	Data field	Other than data field	Data field
During broadcast communication	Hardware processing	–	• Reception stopped • INTIE2 generated • To start bit wait status **Remarks 1.** Communication between other nodes does not stop. **2.** Data reception is not possible until the overrun status is cancelled.	–	• Transmission stopped • INTIE2 generated • To start bit wait status
	Software processing	–	• Execute DR read to cancel overrun status • Error processing (resend request, etc.)	–	• Error processing (resend request, etc.)
During individual communication	Hardware processing	–	• INTIE2 not generated • NACK returned • Data reset from other node **Remark** Data reception is not possible until the overrun status is cancelled.	–	• Transmission stopped • INTIE2 generated • To start bit wait status
	Software processing	–	• Execute DR read to cancel overrun status • Error processing (resend request, etc.)	–	• Error processing (resend request, etc.)

		Parity Error			
Occurrence condition	Local node status	During reception		During transmission	
	Occurrence condition	Receive data and receive parity don't match		–	
	Occurrence location	Other than data field	Data field	Other than data field	Data field
During broadcast communication	Hardware processing	• Reception stopped • INTIE2 generated • To start bit wait status **Remark** Communication between other nodes does not stop.		–	–
	Software processing	• Error processing (resend request, etc.)		–	–
During individual communication	Hardware processing	• Reception stopped • INTIE2 generated • To start bit wait status	• Reception not stopped • INTIE2 not generated • NACK returned • Receive data sent from other nodes	–	–
	Software processing	• Error processing (resend request, etc.)	–	–	–

18.5 Interrupt Generation Timing and Main CPU Processing

18.5.1 Master transmission

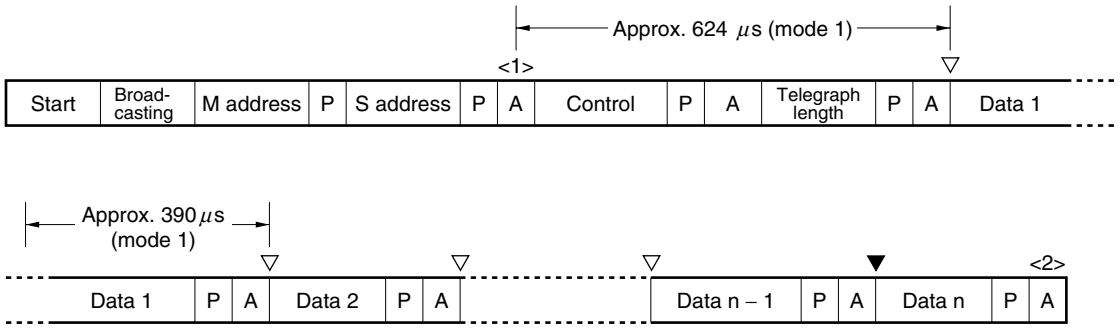
Initial preparation processing:

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 18-17. Master Transmission



<1> Interrupt (INTIE2) occurrence

Judgment of occurrence of error → Error processing

↓

Judgment of slave request → Slave reception processing
(See **18.5.1 (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 **18.5.1 (3) Re-communication processing**)

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **18.5.1 (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 servicing, the data transfer direction of the macro service must change from RAM (memory) ‘ SFR (peripheral) to SFR (peripheral) ‘ RAM (memory) by the time 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, but the same data is retransmitted by hardware.

If the transmit data is not written within the period of writing the next data, a communication error interrupt occurs due to the occurrence of an underrun, and communication ends midway.

(3) Re-communication processing

The vector interrupt servicing in <2> in Figure 18-17 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.

18.5.2 Master reception

Before performing master reception, it is necessary to notify the slave unit of slave transmission. 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.

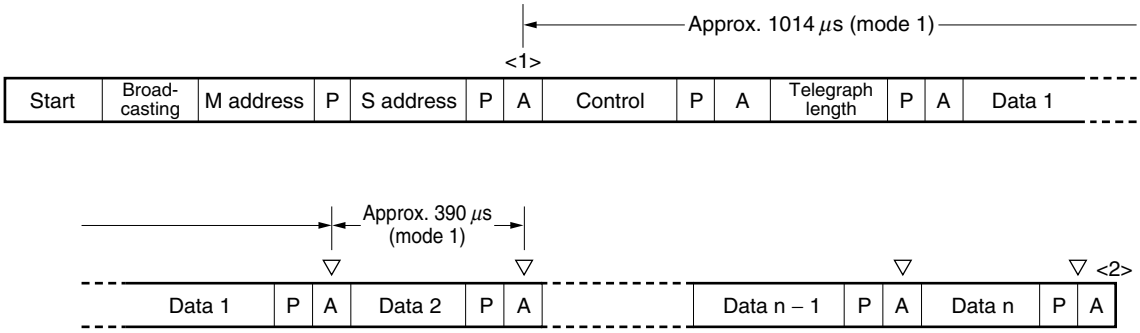
Initial preparation processing:

Sets a unit address, slave address, and control data.

Communication start processing:

Sets the bus control register (enables communication and master request).

Figure 18-18. 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 **18.5.2 (2) Frame end processing**)

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **18.5.2 (1) Interrupt (INTIE1) occurrence**)
The receive data stored in 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, but the same data is retransmitted from the slave.

If the receive data is not read by the time the next data is received, the hardware automatically transmits NACK.

(2) Frame end processing

The vector interrupt servicing in <2> in Figure 18-18 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.

18.5.3 Slave transmission

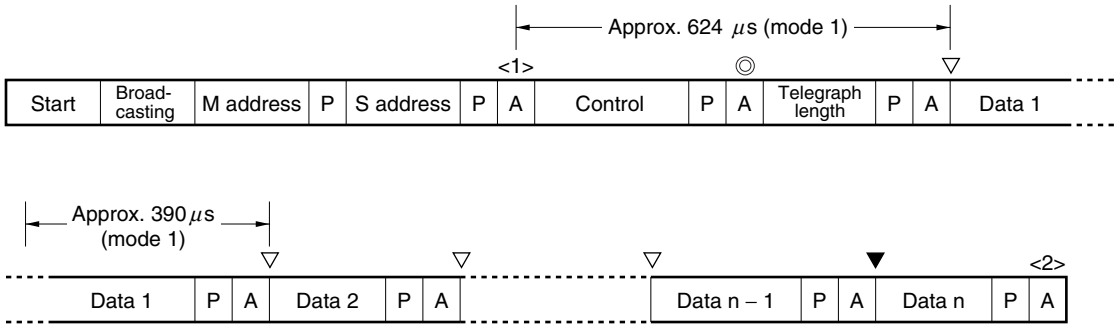
Initial preparation processing:

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 18-19. 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 **18.5.3 (2) Frame end processing**)

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **18.5.3 (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 the locked state, refer to **18.3.2 (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, but the same data is retransmitted by hardware.

If the transmit data is not written within the period of writing the next data, a communication error interrupt occurs due to the occurrence of an underrun, and communication ends abnormally.

(2) Frame end processing

The vector interrupt processing in <2> in Figure 18-19 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.

18.5.4 Slave reception

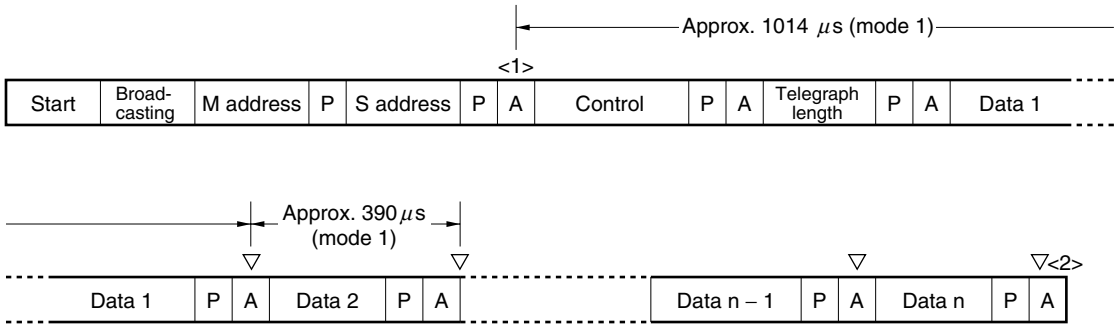
Initial preparation processing:

Sets a unit address.

Communication start processing:

Sets the bus control register (enables communication, disables slave transmission, and enables slave reception).

Figure 18-20. 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 **18.5.4 (2) Frame end processing**).

Remarks 1. ▽: Interrupt (INTIE1) occurrence (See **18.5.4 (1) Interrupt (INTIE1) occurrence**).
The receive data stored in 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, but the same data is retransmitted from the master.

If the receive data is not read in by the time the next data is received, NACK is automatically transmitted.

(2) Frame end processing

The vector interrupt processing in <2> in Figure 18-20 judges whether the data has been correctly received within one frame.

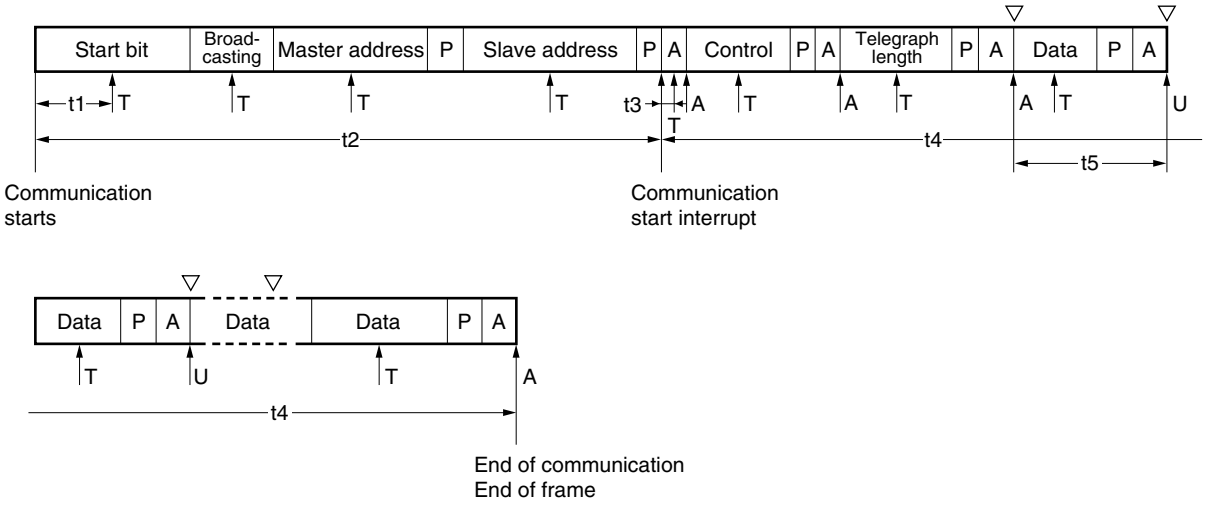
18.5.5 Interval of occurrence of interrupt for IEBus control

Each control interrupt must occur at each point of communication and perform the necessary processing by the time 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 18-21. Master Transmission (Interval of Interrupt Occurrence)



Remarks 1. T: Timing error
A: $\overline{\text{ACK}}$ error
U: Underrun error
▽: Data set interrupt (INTIE1)

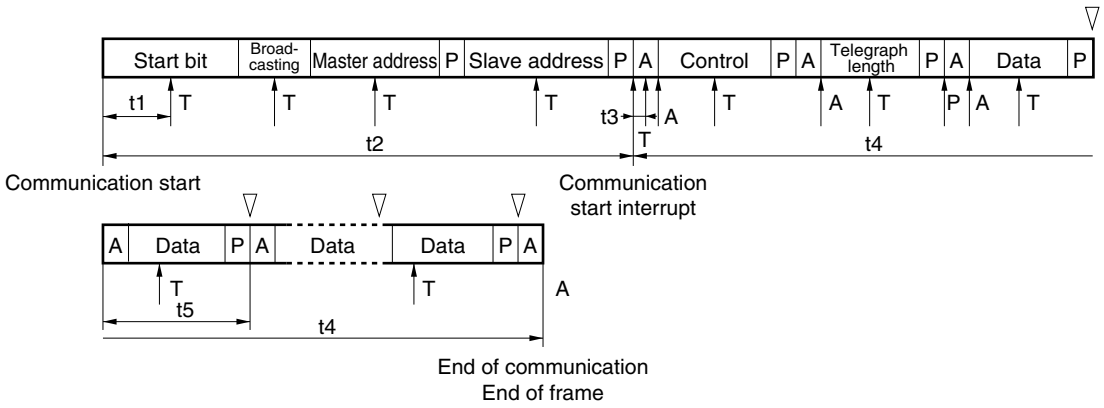
2. End of frame occurs at the end of 32-byte data.

(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 18-22. Master Reception (Interval of Interrupt Occurrence)



Remarks 1. T: Timing error

P: Parity error

A: $\overline{\text{ACK}}$ error

▽: Data set interrupt (INTIE1)

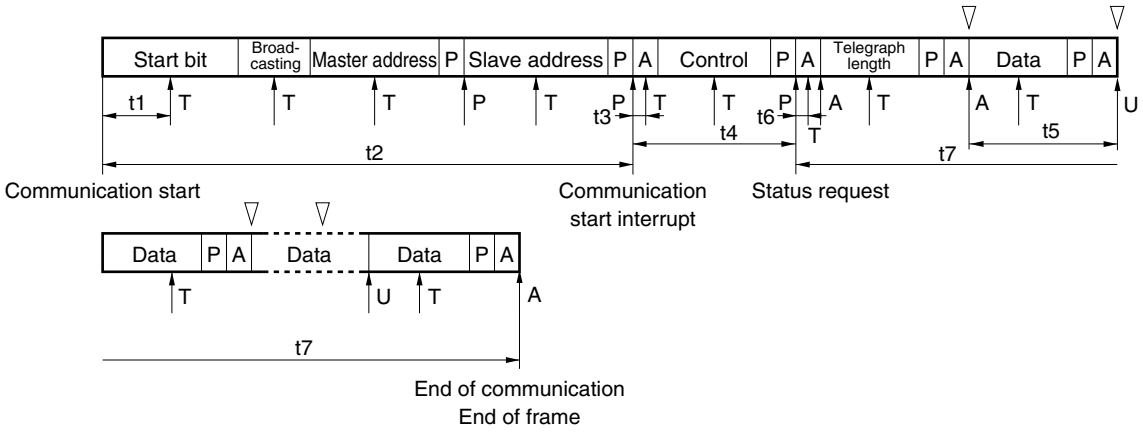
2. End of frame occurs at the end of 32-byte data.

(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 18-23. 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)

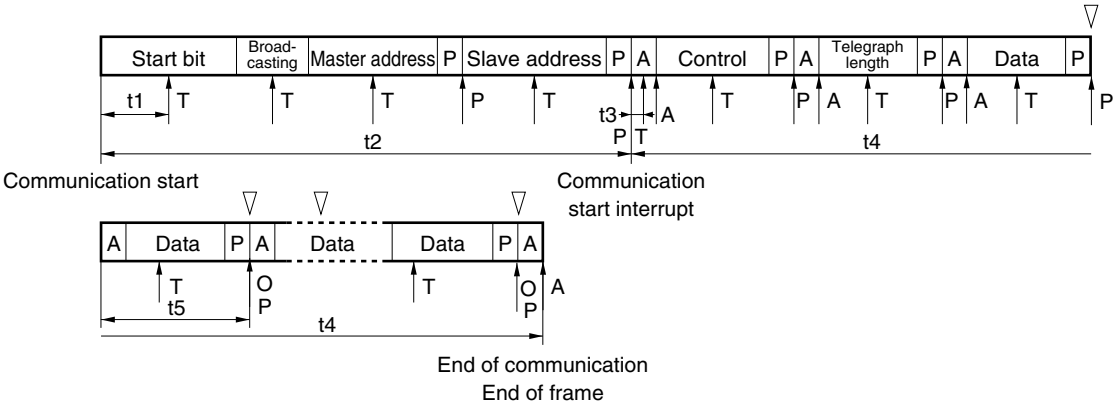
2. End of frame occurs at the end of 32-byte data.

(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 18-24. 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

CHAPTER 19 FCAN CONTROLLER (V850/SC3)

The V850/SC3 features an on-chip FCAN (Full Controller Area Network) controller that complies with CAN specification Ver. 2.0 PartB active. (The V850/SC3 product line includes the μPD703089Y and μPD70F3089Y as two-channel devices and the μPD703088Y as a single-channel device.)

★ 19.1 Features

- CAN specification Ver.2.0 PartB active
- Standard frame and expanded frame transmission/reception enabled
- Remote frame automatic transmission enabled
- Transfer rate: 1 Mbps max.
- 32 message buffers

19.2 Overview of Functions

Table 19-1 presents an overview of FCAN controller functions.

Table 19-1. Overview of Functions

Function	Description
Protocol	CAN Protocol Ver. 2.0, PartB active (standard and extended frame transmission/reception)
★ Baud rate	Maximum 1 Mbps
Data storage	• Allocated to common access-enabled RAM area • RAM that is mapped to an unused message buffer can be used for CPU processing or other processing
Mask functions	• Mask setting of four patterns is possible for each CAN module • Global masks and local masks can be used in common
Message configuration	Can be declared as transmit message or receive message
No. of messages	32 messages
Message storage method	• Storage in message buffer with unique ID • Storage in buffer specified by receive mask function
Remote reception	• Remote frames can be received in either the receive message buffer or the transmit message buffer • If a remote frame is received by a transmit message buffer, there is a choice between having the remote request processed by the CPU or starting the auto transmit function.
Remote transmission	The remote frame can be sent either by setting the transmit message's RTR bit (M_CTRLn register) or by setting the receive message's send request.
Time stamp function	A time stamp function can be set for receive messages and transmit messages.
Diagnostic functions	• Read-enabled error counter • "Valid protocol operation flag" for verification of bus connections • Receive-only mode (with auto baud rate detection) • Diagnostic processing mode
Low-power mode	• CAN sleep mode (wakeup function using CAN bus enabled) • CAN stop mode (wakeup function using CAN bus disabled)

Remark n = 00 to 31

19.3 Configuration

FCAN is composed of the following four blocks.

(1) NPB interface

This functional block provides an NPB (NEC peripheral I/O bus) interface and a means of transmitting and receiving signals.

(2) MAC (Memory Access Controller)

This functional block controls access to the CAN module within the FCAN and to the CAN RAM.

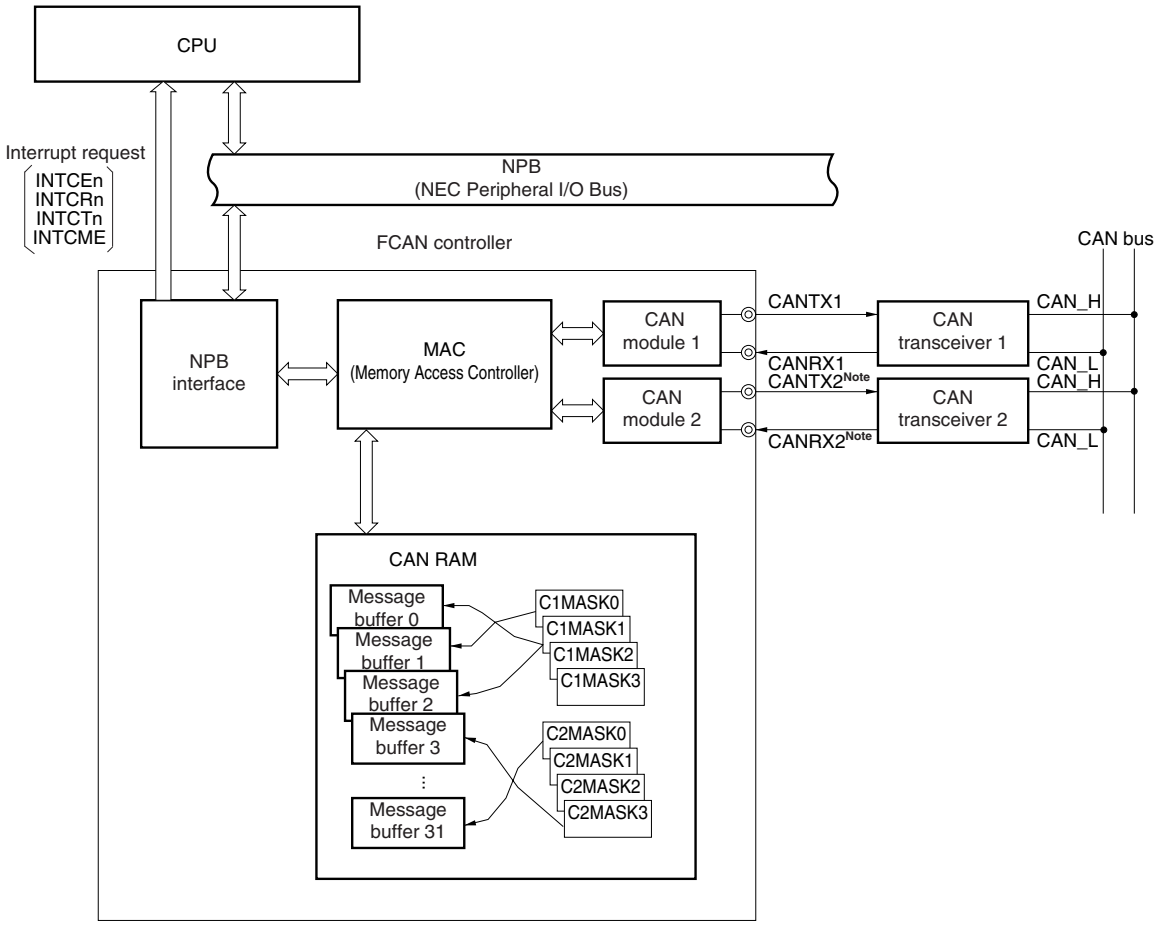
(3) CAN module

This functional block is involved in the operation of the CAN protocol layer and its related settings.

(4) CAN RAM

This is the CAN memory functional block, which is used to store message IDs, message data, etc.

Figure 19-1. Block Diagram of FCAN



Note μPD703089Y and 70F3089Y only

Cautions 1. When P114/CANTX1, P115/CANRX1, P116/CANTX2, P117/CANRX2 are used during FCAN transmission/reception, they can be used as FCAN pin functions (CANTX1, CANRX1, CANTX2, CANRX2) by setting the port alternate function control register (PAC) (refer to 5.2.10 (2) (b) Port alternate-function control register (PAC)).

2. When the P114/CANTX1 and P116/CANTX2 pins are used as CANTX1, CANTX2, set both the P11 and PM11 registers to 0 (refer to 5.3 Using Port Pins as Alternate-Function Pins).

3. When the P115/CANRX1 and P117/CANRX2 pins are used as CANRX1 and CANRX2, set the P11 register to 0 and the PM11 register to 1.

4. If the FCAN register is read/written when the external bus interface function is used, an address/data control signal is output to the external expansion pins (ports 4, 5, 6, 9), so read/write of xx3FF800H to xx3FFFFFH, which is the FCAN address area, should not be performed for the external devices connected to the external expansion pins.

5. If the wait function and idle function are set when the external bus interface function is used, these functions are enabled even reading/writing the FCAN register.

6. Since no clock is supplied from the subclock to FCAN, when stopping the main clock and setting the subclock operation, do not read/write the FCAN register.

Remark n = 1, 2

19.4 Internal Registers of FCAN Controller

19.4.1 Configuration of messages and buffers

Table 19-2. Configuration of Messages and Buffers

Address	Register Name
xx3FF800H to xx3FF81FH	Message buffer 0 field
xx3FF820H to xx3FF83FH	Message buffer 1 field
xx3FF840H to xx3FF85FH	Message buffer 2 field
xx3FF860H to xx3FF87FH	Message buffer 3 field
xx3FF880H to xx3FF89FH	Message buffer 4 field
xx3FF8A0H to xx3FF8BFH	Message buffer 5 field
xx3FF8C0H to xx3FF8DFH	Message buffer 6 field
xx3FF8E0H to xx3FF8FFH	Message buffer 7 field
xx3FF900H to xx3FF91FH	Message buffer 8 field
xx3FF920H to xx3FF93FH	Message buffer 9 field
xx3FF940H to xx3FF95FH	Message buffer 10 field
xx3FF960H to xx3FF97FH	Message buffer 11 field
xx3FF980H to xx3FF99FH	Message buffer 12 field
xx3FF9A0H to xx3FF9BFH	Message buffer 13 field
xx3FF9C0H to xx3FF9DFH	Message buffer 14 field
xx3FF9E0H to xx3FF9FFH	Message buffer 15 field
xx3FFA00H to xx3FFA1FH	Message buffer 16 field
xx3FFA20H to xx3FFA3FH	Message buffer 17 field
xx3FFA40H to xx3FFA5FH	Message buffer 18 field
xx3FFA60H to xx3FFA7FH	Message buffer 19 field
xx3FFA80H to xx3FFA9FH	Message buffer 20 field
xx3FFAA0H to xx3FFABFH	Message buffer 21 field
xx3FFAC0H to xx3FFADFH	Message buffer 22 field
xx3FFAE0H to xx3FFAFFH	Message buffer 23 field
xx3FFB00H to xx3FFB1FH	Message buffer 24 field
xx3FFB20H to xx3FFB3FH	Message buffer 25 field
xx3FFB40H to xx3FFB5FH	Message buffer 26 field
xx3FFB60H to xx3FFB7FH	Message buffer 27 field
xx3FFB80H to xx3FFB9FH	Message buffer 28 field
xx3FFBA0H to xx3FFBBFH	Message buffer 29 field
xx3FFBC0H to xx3FFBDFH	Message buffer 30 field
xx3FFBE0H to xx3FFBFFH	Message buffer 31 field

Caution The addresses xx3FF800H to xx3FFFFFH are specified as a physical FCAN address area, whose image is visible in addresses xxnFF800H to xxnFFFFFH (n = 7, B).
Accessing addresses from xxnFF800H to xxnFFFFFH is prohibited.

Remark For details of message buffers, see **19.4.2 List of FCAN registers**.

19.4.2 List of FCAN registers

(1/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FF804H	CAN message data length register 00	M_DLC00	R/W		√		Undefined
xx3FF805H	CAN message control register 00	M_CTRL00			√		
xx3FF806H	CAN message time stamp register 00	M_TIME00				√	
xx3FF808H	CAN message data register 000	M_DATA000			√		
xx3FF809H	CAN message data register 001	M_DATA001			√		
xx3FF80AH	CAN message data register 002	M_DATA002			√		
xx3FF80BH	CAN message data register 003	M_DATA003			√		
xx3FF80CH	CAN message data register 004	M_DATA004			√		
xx3FF80DH	CAN message data register 005	M_DATA005			√		
xx3FF80EH	CAN message data register 006	M_DATA006			√		
xx3FF80FH	CAN message data register 007	M_DATA007			√		
xx3FF810H	CAN message ID register L00	M_IDL00				√	
xx3FF812H	CAN message ID register H00	M_IDH00				√	
xx3FF814H	CAN message configuration register 00	M_CONF00			√		
xx3FF815H	CAN message status register 00	M_STAT00	R		√		
xx3FF816H	CAN status set/clear register 00	SC_STAT00	W			√	0000H
xx3FF824H	CAN message data length register 01	M_DLC01	R/W		√		Undefined
xx3FF825H	CAN message control register 01	M_CTRL01			√		
xx3FF826H	CAN message time stamp register 01	M_TIME01				√	
xx3FF828H	CAN message data register 010	M_DATA010			√		
xx3FF829H	CAN message data register 011	M_DATA011			√		
xx3FF82AH	CAN message data register 012	M_DATA012			√		
xx3FF82BH	CAN message data register 013	M_DATA013			√		
xx3FF82CH	CAN message data register 014	M_DATA014			√		
xx3FF82DH	CAN message data register 015	M_DATA015			√		
xx3FF82EH	CAN message data register 016	M_DATA016			√		
xx3FF82FH	CAN message data register 017	M_DATA017			√		
xx3FF830H	CAN message ID register L01	M_IDL01				√	
xx3FF832H	CAN message ID register H01	M_IDH01				√	
xx3FF834H	CAN message configuration register 01	M_CONF01			√		
xx3FF835H	CAN message status register 01	M_STAT01	R		√		
xx3FF836H	CAN status set/clear register 01	SC_STAT01	W			√	0000H
xx3FF844H	CAN message data length register 02	M_DLC02	R/W		√		Undefined
xx3FF845H	CAN message control register 02	M_CTRL02			√		
xx3FF846H	CAN message time stamp register 02	M_TIME02				√	
xx3FF848H	CAN message data register 020	M_DATA020			√		
xx3FF849H	CAN message data register 021	M_DATA021			√		
xx3FF84AH	CAN message data register 022	M_DATA022			√		
xx3FF84BH	CAN message data register 023	M_DATA023			√		
xx3FF84CH	CAN message data register 024	M_DATA024			√		
xx3FF84DH	CAN message data register 025	M_DATA025			√		

(2/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FF84EH	CAN message data register 026	M_DATA026	R/W		√		Undefined
xx3FF84FH	CAN message data register 027	M_DATA027			√		
xx3FF850H	CAN message ID register L02	M_IDL02				√	
xx3FF852H	CAN message ID register H02	M_IDH02				√	
xx3FF854H	CAN message configuration register 02	M_CONF02			√		
xx3FF855H	CAN message status register 02	M_STAT02	R		√		
xx3FF856H	CAN status set/clear register 02	SC_STAT02	W			√	0000H
xx3FF864H	CAN message data length register 03	M_DLC03	R/W		√		Undefined
xx3FF865H	CAN message control register 03	M_CTRL03			√		
xx3FF866H	CAN message time stamp register 03	M_TIME03				√	
xx3FF868H	CAN message data register 030	M_DATA030			√		
xx3FF869H	CAN message data register 031	M_DATA031			√		
xx3FF86AH	CAN message data register 032	M_DATA032			√		
xx3FF86BH	CAN message data register 033	M_DATA033			√		
xx3FF86CH	CAN message data register 034	M_DATA034			√		
xx3FF86DH	CAN message data register 035	M_DATA035			√		
xx3FF86EH	CAN message data register 036	M_DATA036			√		
xx3FF86FH	CAN message data register 037	M_DATA037			√		
xx3FF870H	CAN message ID register L03	M_IDL03				√	
xx3FF872H	CAN message ID register H03	M_IDH03				√	
xx3FF874H	CAN message configuration register 03	M_CONF03			√		
xx3FF875H	CAN message status register 03	M_STAT03	R		√		
xx3FF876H	CAN status set/clear register 03	SC_STAT03	W			√	0000H
xx3FF884H	CAN message data length register 04	M_DLC04	R/W		√		Undefined
xx3FF885H	CAN message control register 04	M_CTRL04			√		
xx3FF886H	CAN message time stamp register 04	M_TIME04				√	
xx3FF888H	CAN message data register 040	M_DATA040			√		
xx3FF889H	CAN message data register 041	M_DATA041			√		
xx3FF88AH	CAN message data register 042	M_DATA042			√		
xx3FF88BH	CAN message data register 043	M_DATA043			√		
xx3FF88CH	CAN message data register 044	M_DATA044			√		
xx3FF88DH	CAN message data register 045	M_DATA045			√		
xx3FF88EH	CAN message data register 046	M_DATA046			√		
xx3FF88FH	CAN message data register 047	M_DATA047			√		
xx3FF890H	CAN message ID register L04	M_IDL04				√	
xx3FF892H	CAN message ID register H04	M_IDH04				√	
xx3FF894H	CAN message configuration register 04	M_CONF04			√		
xx3FF895H	CAN message status register 04	M_STAT04	R		√		
xx3FF896H	CAN status set/clear register 04	SC_STAT04	W			√	0000H
xx3FF8A4H	CAN message data length register 05	M_DLC05	R/W		√		Undefined
xx3FF8A5H	CAN message control register 05	M_CTRL05			√		
xx3FF8A6H	CAN message time stamp register 05	M_TIME05				√	
xx3FF8A8H	CAN message data register 050	M_DATA050			√		
xx3FF8A9H	CAN message data register 051	M_DATA051			√		

(3/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FF8AAH	CAN message data register 052	M_DATA052	R/W		√		Undefined
xx3FF8ABH	CAN message data register 053	M_DATA053			√		
xx3FF8ACH	CAN message data register 054	M_DATA054			√		
xx3FF8ADH	CAN message data register 055	M_DATA055			√		
xx3FF8AEH	CAN message data register 056	M_DATA056			√		
xx3FF8AFH	CAN message data register 057	M_DATA057			√		
xx3FF8B0H	CAN message ID register L05	M_IDL05				√	
xx3FF8B2H	CAN message ID register H05	M_IDH05				√	
xx3FF8B4H	CAN message configuration register 05	M_CONF05			√		
xx3FF8B5H	CAN message status register 05	M_STAT05	R		√		
xx3FF8B6H	CAN status set/clear register 05	SC_STAT05	W			√	0000H
xx3FF8C4H	CAN message data length register 06	M_DLC06	R/W		√		Undefined
xx3FF8C5H	CAN message control register 06	M_CTRL06			√		
xx3FF8C6H	CAN message time stamp register 06	M_TIME06				√	
xx3FF8C8H	CAN message data register 060	M_DATA060			√		
xx3FF8C9H	CAN message data register 061	M_DATA061			√		
xx3FF8CAH	CAN message data register 062	M_DATA062			√		
xx3FF8CBH	CAN message data register 063	M_DATA063			√		
xx3FF8CCH	CAN message data register 064	M_DATA064			√		
xx3FF8CDH	CAN message data register 065	M_DATA065			√		
xx3FF8CEH	CAN message data register 066	M_DATA066			√		
xx3FF8CFH	CAN message data register 067	M_DATA067			√		
xx3FF8D0H	CAN message ID register L06	M_IDL06				√	
xx3FF8D2H	CAN message ID register H06	M_IDH06				√	
xx3FF8D4H	CAN message configuration register 06	M_CONF06			√		
xx3FF8D5H	CAN message status register 06	M_STAT06	R		√		
xx3FF8D6H	CAN status set/clear register 06	SC_STAT06	W			√	0000H
xx3FF8E4H	CAN message data length register 07	M_DLC07	R/W		√		Undefined
xx3FF8E5H	CAN message control register 07	M_CTRL07			√		
xx3FF8E6H	CAN message time stamp register 07	M_TIME07				√	
xx3FF8E8H	CAN message data register 070	M_DATA070			√		
xx3FF8E9H	CAN message data register 071	M_DATA071			√		
xx3FF8EAH	CAN message data register 072	M_DATA072			√		
xx3FF8EBH	CAN message data register 073	M_DATA073			√		
xx3FF8ECH	CAN message data register 074	M_DATA074			√		
xx3FF8EDH	CAN message data register 075	M_DATA075			√		
xx3FF8EEH	CAN message data register 076	M_DATA076			√		
xx3FF8EFH	CAN message data register 077	M_DATA077			√		
xx3FF8F0H	CAN message ID register L07	M_IDL07				√	
xx3FF8F2H	CAN message ID register H07	M_IDH07				√	
xx3FF8F4H	CAN message configuration register 07	M_CONF07			√		
xx3FF8F5H	CAN message status register 07	M_STAT07	R		√		
xx3FF8F6H	CAN status set/clear register 07	SC_STAT07	W			√	0000H
xx3FF904H	CAN message data length register 08	M_DLC08	R/W		√		Undefined

(4/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FF905H	CAN message control register 08	M_CTRL08	R/W		√		Undefined
xx3FF906H	CAN message time stamp register 08	M_TIME08				√	
xx3FF908H	CAN message data register 080	M_DATA080			√		
xx3FF909H	CAN message data register 081	M_DATA081			√		
xx3FF90AH	CAN message data register 082	M_DATA082			√		
xx3FF90BH	CAN message data register 083	M_DATA083			√		
xx3FF90CH	CAN message data register 084	M_DATA084			√		
xx3FF90DH	CAN message data register 085	M_DATA085			√		
xx3FF90EH	CAN message data register 086	M_DATA086			√		
xx3FF90FH	CAN message data register 087	M_DATA087			√		
xx3FF910H	CAN message ID register L08	M_IDL08				√	
xx3FF912H	CAN message ID register H08	M_IDH08				√	
xx3FF914H	CAN message configuration register 08	M_CONF08			√		
xx3FF915H	CAN message status register 08	M_STAT08	R		√		
xx3FF916H	CAN status set/clear register 08	SC_STAT08	W			√	0000H
xx3FF924H	CAN message data length register 09	M_DLC09	R/W		√		Undefined
xx3FF925H	CAN message control register 09	M_CTRL09			√		
xx3FF926H	CAN message time stamp register 09	M_TIME09				√	
xx3FF928H	CAN message data register 090	M_DATA090			√		
xx3FF929H	CAN message data register 091	M_DATA091			√		
xx3FF92AH	CAN message data register 092	M_DATA092			√		
xx3FF92BH	CAN message data register 093	M_DATA093			√		
xx3FF92CH	CAN message data register 094	M_DATA094			√		
xx3FF92DH	CAN message data register 095	M_DATA095			√		
xx3FF92EH	CAN message data register 096	M_DATA096			√		
xx3FF92FH	CAN message data register 097	M_DATA097			√		
xx3FF930H	CAN message ID register L09	M_IDL09				√	
xx3FF932H	CAN message ID register H09	M_IDH09				√	
xx3FF934H	CAN message configuration register 09	M_CONF09			√		
xx3FF935H	CAN message status register 09	M_STAT09	R		√		
xx3FF936H	CAN status set/clear register 09	SC_STAT09	W			√	0000H
xx3FF944H	CAN message data length register 10	M_DLC10	R/W		√		Undefined
xx3FF945H	CAN message control register 10	M_CTRL10			√		
xx3FF946H	CAN message time stamp register 10	M_TIME10				√	
xx3FF948H	CAN message data register 100	M_DATA100			√		
xx3FF949H	CAN message data register 101	M_DATA101			√		
xx3FF94AH	CAN message data register 102	M_DATA102			√		
xx3FF94BH	CAN message data register 103	M_DATA103			√		
xx3FF94CH	CAN message data register 104	M_DATA104			√		
xx3FF94DH	CAN message data register 105	M_DATA105			√		
xx3FF94EH	CAN message data register 106	M_DATA106			√		
xx3FF94FH	CAN message data register 107	M_DATA107			√		
xx3FF950H	CAN message ID register L10	M_IDL10				√	
xx3FF952H	CAN message ID register H10	M_IDH10				√	

(5/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FF954H	CAN message configuration register 10	M_CONF10	R/W		√		Undefined
xx3FF955H	CAN message status register 10	M_STAT10	R		√		
xx3FF956H	CAN status set/clear register 10	SC_STAT10	W			√	0000H
xx3FF964H	CAN message data length register 11	M_DLC11	R/W		√		Undefined
xx3FF965H	CAN message control register 11	M_CTRL11			√		
xx3FF966H	CAN message time stamp register 11	M_TIME11				√	
xx3FF968H	CAN message data register 110	M_DATA110			√		
xx3FF969H	CAN message data register 111	M_DATA111			√		
xx3FF96AH	CAN message data register 112	M_DATA112			√		
xx3FF96BH	CAN message data register 113	M_DATA113			√		
xx3FF96CH	CAN message data register 114	M_DATA114			√		
xx3FF96DH	CAN message data register 115	M_DATA115			√		
xx3FF96EH	CAN message data register 116	M_DATA116			√		
xx3FF96FH	CAN message data register 117	M_DATA117			√		
xx3FF970H	CAN message ID register L11	M_IDL11				√	
xx3FF972H	CAN message ID register H11	M_IDH11				√	
xx3FF974H	CAN message configuration register 11	M_CONF11			√		
xx3FF975H	CAN message status register 11	M_STAT11	R		√		
xx3FF976H	CAN status set/clear register 11	SC_STAT11	W			√	0000H
xx3FF984H	CAN message data length register 12	M_DLC12	R/W		√		Undefined
xx3FF985H	CAN message control register 12	M_CTRL12			√		
xx3FF986H	CAN message time stamp register 12	M_TIME12				√	
xx3FF988H	CAN message data register 120	M_DATA120			√		
xx3FF989H	CAN message data register 121	M_DATA121			√		
xx3FF98AH	CAN message data register 122	M_DATA122			√		
xx3FF98BH	CAN message data register 123	M_DATA123			√		
xx3FF98CH	CAN message data register 124	M_DATA124			√		
xx3FF98DH	CAN message data register 125	M_DATA125			√		
xx3FF98EH	CAN message data register 126	M_DATA126			√		
xx3FF98FH	CAN message data register 127	M_DATA127			√		
xx3FF990H	CAN message ID register L12	M_IDL12				√	
xx3FF992H	CAN message ID register H12	M_IDH12				√	
xx3FF994H	CAN message configuration register 12	M_CONF12			√		
xx3FF995H	CAN message status register 12	M_STAT12	R		√		
xx3FF996H	CAN status set/clear register 12	SC_STAT12	W			√	0000H
xx3FF9A4H	CAN message data length register 13	M_DLC13	R/W		√		Undefined
xx3FF9A5H	CAN message control register 13	M_CTRL13			√		
xx3FF9A6H	CAN message time stamp register 13	M_TIME13				√	
xx3FF9A8H	CAN message data register 130	M_DATA130			√		
xx3FF9A9H	CAN message data register 131	M_DATA131			√		
xx3FF9AAH	CAN message data register 132	M_DATA132			√		
xx3FF9ABH	CAN message data register 133	M_DATA133			√		
xx3FF9ACH	CAN message data register 134	M_DATA134			√		
xx3FF9ADH	CAN message data register 135	M_DATA135			√		

(6/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FF9AEH	CAN message data register 136	M_DATA136	R/W		√		Undefined
xx3FF9AFH	CAN message data register 137	M_DATA137			√		
xx3FF9B0H	CAN message ID register L13	M_IDL13					
xx3FF9B2H	CAN message ID register H13	M_IDH13					
xx3FF9B4H	CAN message configuration register 13	M_CONF13			√		
xx3FF9B5H	CAN message status register 13	M_STAT13	R		√		
xx3FF9B6H	CAN status set/clear register 13	SC_STAT13	W			√	0000H
xx3FF9C4H	CAN message data length register 14	M_DLC14	R/W		√		Undefined
xx3FF9C5H	CAN message control register 14	M_CTRL14			√		
xx3FF9C6H	CAN message time stamp register 14	M_TIME14				√	
xx3FF9C8H	CAN message data register 140	M_DATA140			√		
xx3FF9C9H	CAN message data register 141	M_DATA141			√		
xx3FF9CAH	CAN message data register 142	M_DATA142			√		
xx3FF9CBH	CAN message data register 143	M_DATA143			√		
xx3FF9CCH	CAN message data register 144	M_DATA144			√		
xx3FF9CDH	CAN message data register 145	M_DATA145			√		
xx3FF9CEH	CAN message data register 146	M_DATA146			√		
xx3FF9CFH	CAN message data register 147	M_DATA147			√		
xx3FF9D0H	CAN message ID register L14	M_IDL14				√	
xx3FF9D2H	CAN message ID register H14	M_IDH14				√	
xx3FF9D4H	CAN message configuration register 14	M_CONF14			√		
xx3FF9D5H	CAN message status register 14	M_STAT14	R		√		
xx3FF9D6H	CAN status set/clear register 14	SC_STAT14	W			√	0000H
xx3FF9E4H	CAN message data length register 15	M_DLC15	R/W		√		Undefined
xx3FF9E5H	CAN message control register 15	M_CTRL15			√		
xx3FF9E6H	CAN message time stamp register 15	M_TIME15				√	
xx3FF9E8H	CAN message data register 150	M_DATA150			√		
xx3FF9E9H	CAN message data register 151	M_DATA151			√		
xx3FF9EAH	CAN message data register 152	M_DATA152			√		
xx3FF9EBH	CAN message data register 153	M_DATA153			√		
xx3FF9ECH	CAN message data register 154	M_DATA154			√		
xx3FF9EDH	CAN message data register 155	M_DATA155			√		
xx3FF9EEH	CAN message data register 156	M_DATA156			√		
xx3FF9EFH	CAN message data register 157	M_DATA157			√		
xx3FF9F0H	CAN message ID register L15	M_IDL15				√	
xx3FF9F2H	CAN message ID register H15	M_IDH15				√	
xx3FF9F4H	CAN message configuration register 15	M_CONF15			√		
xx3FF9F5H	CAN message status register 15	M_STAT15	R		√		
xx3FF9F6H	CAN status set/clear register 15	SC_STAT15	W			√	0000H
xx3FFA04H	CAN message data length register 16	M_DLC16	R/W		√		Undefined
xx3FFA05H	CAN message control register 16	M_CTRL16			√		
xx3FFA06H	CAN message time stamp register 16	M_TIME16				√	
xx3FFA08H	CAN message data register 160	M_DATA160			√		
xx3FFA09H	CAN message data register 161	M_DATA161			√		

(7/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FFA0AH	CAN message data register 162	M_DATA162	R/W		√		Undefined
xx3FFA0BH	CAN message data register 163	M_DATA163			√		
xx3FFA0CH	CAN message data register 164	M_DATA164			√		
xx3FFA0DH	CAN message data register 165	M_DATA165			√		
xx3FFA0EH	CAN message data register 166	M_DATA166			√		
xx3FFA0FH	CAN message data register 167	M_DATA167			√		
xx3FFA10H	CAN message ID register L16	M_IDL16				√	
xx3FFA12H	CAN message ID register H16	M_IDH16				√	
xx3FFA14H	CAN message configuration register 16	M_CONF16			√		
xx3FFA15H	CAN message status register 16	M_STAT16	R		√		
xx3FFA16H	CAN status set/clear register 16	SC_STAT16	W			√	0000H
xx3FFA24H	CAN message data length register 17	M_DLC17	R/W		√		Undefined
xx3FFA25H	CAN message control register 17	M_CTRL17			√		
xx3FFA26H	CAN message time stamp register 17	M_TIME17				√	
xx3FFA28H	CAN message data register 170	M_DATA170			√		
xx3FFA29H	CAN message data register 171	M_DATA171			√		
xx3FFA2AH	CAN message data register 172	M_DATA172			√		
xx3FFA2BH	CAN message data register 173	M_DATA173			√		
xx3FFA2CH	CAN message data register 174	M_DATA174			√		
xx3FFA2DH	CAN message data register 175	M_DATA175			√		
xx3FFA2EH	CAN message data register 176	M_DATA176			√		
xx3FFA2FH	CAN message data register 177	M_DATA177			√		
xx3FFA30H	CAN message ID register L17	M_IDL17				√	
xx3FFA32H	CAN message ID register H17	M_IDH17				√	
xx3FFA34H	CAN message configuration register 17	M_CONF17			√		
xx3FFA35H	CAN message status register 17	M_STAT17	R		√		
xx3FFA36H	CAN status set/clear register 17	SC_STAT17	W			√	0000H
xx3FFA44H	CAN message data length register 18	M_DLC18	R/W		√		Undefined
xx3FFA45H	CAN message control register 18	M_CTRL18			√		
xx3FFA46H	CAN message time stamp register 18	M_TIME18				√	
xx3FFA48H	CAN message data register 180	M_DATA180			√		
xx3FFA49H	CAN message data register 181	M_DATA181			√		
xx3FFA4AH	CAN message data register 182	M_DATA182			√		
xx3FFA4BH	CAN message data register 183	M_DATA183			√		
xx3FFA4CH	CAN message data register 184	M_DATA184			√		
xx3FFA4DH	CAN message data register 185	M_DATA185			√		
xx3FFA4EH	CAN message data register 186	M_DATA186			√		
xx3FFA4FH	CAN message data register 187	M_DATA187			√		
xx3FFA50H	CAN message ID register L18	M_IDL18				√	
xx3FFA52H	CAN message ID register H18	M_IDH18				√	
xx3FFA54H	CAN message configuration register 18	M_CONF18			√		
xx3FFA55H	CAN message status register 18	M_STAT18	R		√		
xx3FFA56H	CAN status set/clear register 18	SC_STAT18	W			√	0000H
xx3FFA64H	CAN message data length register 19	M_DLC19	R/W		√		Undefined

(8/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FFA65H	CAN message control register 19	M_CTRL19	R/W		√		Undefined
xx3FFA66H	CAN message time stamp register 19	M_TIME19				√	
xx3FFA68H	CAN message data register 190	M_DATA190			√		
xx3FFA69H	CAN message data register 191	M_DATA191			√		
xx3FFA6AH	CAN message data register 192	M_DATA192			√		
xx3FFA6BH	CAN message data register 193	M_DATA193			√		
xx3FFA6CH	CAN message data register 194	M_DATA194			√		
xx3FFA6DH	CAN message data register 195	M_DATA195			√		
xx3FFA6EH	CAN message data register 196	M_DATA196			√		
xx3FFA6FH	CAN message data register 197	M_DATA197			√		
xx3FFA70H	CAN message ID register L19	M_IDL19				√	
xx3FFA72H	CAN message ID register H19	M_IDH19				√	
xx3FFA74H	CAN message configuration register 19	M_CONF19			√		
xx3FFA75H	CAN message status register 19	M_STAT19	R		√		
xx3FFA76H	CAN status set/clear register 19	SC_STAT19	W			√	0000H
xx3FFA84H	CAN message data length register 20	M_DLC20	R/W		√		Undefined
xx3FFA85H	CAN message control register 20	M_CTRL20			√		
xx3FFA86H	CAN message time stamp register 20	M_TIME20				√	
xx3FFA88H	CAN message data register 200	M_DATA200			√		
xx3FFA89H	CAN message data register 201	M_DATA201			√		
xx3FFA8AH	CAN message data register 202	M_DATA202			√		
xx3FFA8BH	CAN message data register 203	M_DATA203			√		
xx3FFA8CH	CAN message data register 204	M_DATA204			√		
xx3FFA8DH	CAN message data register 205	M_DATA205			√		
xx3FFA8EH	CAN message data register 206	M_DATA206			√		
xx3FFA8FH	CAN message data register 207	M_DATA207			√		
xx3FFA90H	CAN message ID register L20	M_IDL20				√	
xx3FFA92H	CAN message ID register H20	M_IDH20				√	
xx3FFA94H	CAN message configuration register 20	M_CONF20			√		
xx3FFA95H	CAN message status register 20	M_STAT20	R		√		
xx3FFA96H	CAN status set/clear register 20	SC_STAT20	W			√	0000H
xx3FFAA4H	CAN message data length register 21	M_DLC21	R/W		√		Undefined
xx3FFAA5H	CAN message control register 21	M_CTRL21			√		
xx3FFAA6H	CAN message time stamp register 21	M_TIME21				√	
xx3FFAA8H	CAN message data register 210	M_DATA210			√		
xx3FFAA9H	CAN message data register 211	M_DATA211			√		
xx3FFAAAH	CAN message data register 212	M_DATA212			√		
xx3FFAABH	CAN message data register 213	M_DATA213			√		
xx3FFAACH	CAN message data register 214	M_DATA214			√		
xx3FFAADH	CAN message data register 215	M_DATA215			√		
xx3FFAAEH	CAN message data register 216	M_DATA216			√		
xx3FFAAFH	CAN message data register 217	M_DATA217			√		
xx3FFAB0H	CAN message ID register L21	M_IDL21				√	
xx3FFAB2H	CAN message ID register H21	M_IDH21				√	

(9/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FFAB4H	CAN message configuration register 21	M_CONF21	R/W		√		Undefined
xx3FFAB5H	CAN message status register 21	M_STAT21	R		√		
xx3FFAB6H	CAN status set/clear register 21	SC_STAT21	W			√	0000H
xx3FFAC4H	CAN message data length register 22	M_DLC22	R/W		√		Undefined
xx3FFAC5H	CAN message control register 22	M_CTRL22			√		
xx3FFAC6H	CAN message time stamp register 22	M_TIME22				√	
xx3FFAC8H	CAN message data register 220	M_DATA220			√		
xx3FFAC9H	CAN message data register 221	M_DATA221			√		
xx3FFACAH	CAN message data register 222	M_DATA222			√		
xx3FFACBH	CAN message data register 223	M_DATA223			√		
xx3FFACCH	CAN message data register 224	M_DATA224			√		
xx3FFACDH	CAN message data register 225	M_DATA225			√		
xx3FFACEH	CAN message data register 226	M_DATA226			√		
xx3FFACFH	CAN message data register 227	M_DATA227			√		
xx3FFAD0H	CAN message ID register L22	M_IDL22				√	
xx3FFAD2H	CAN message ID register H22	M_IDH22				√	
xx3FFAD4H	CAN message configuration register 22	M_CONF22			√		
xx3FFAD5H	CAN message status register 22	M_STAT22	R		√		
xx3FFAD6H	CAN status set/clear register 22	SC_STAT22	W			√	0000H
xx3FFAE4H	CAN message data length register 23	M_DLC23	R/W		√		Undefined
xx3FFAE5H	CAN message control register 23	M_CTRL23			√		
xx3FFAE6H	CAN message time stamp register 23	M_TIME23				√	
xx3FFAE8H	CAN message data register 230	M_DATA230			√		
xx3FFAE9H	CAN message data register 231	M_DATA231			√		
xx3FFAEAH	CAN message data register 232	M_DATA232			√		
xx3FFAEBH	CAN message data register 233	M_DATA233			√		
xx3FFAECH	CAN message data register 234	M_DATA234			√		
xx3FFAEDH	CAN message data register 235	M_DATA235			√		
xx3FFAEEH	CAN message data register 236	M_DATA236			√		
xx3FFAEFH	CAN message data register 237	M_DATA237			√		
xx3FFAF0H	CAN message ID register L23	M_IDL23				√	
xx3FFAF2H	CAN message ID register H23	M_IDH23				√	
xx3FFAF4H	CAN message configuration register 23	M_CONF23			√		
xx3FFAF5H	CAN message status register 23	M_STAT23	R		√		
xx3FFAF6H	CAN status set/clear register 23	SC_STAT23	W			√	0000H
xx3FFB04H	CAN message data length register 24	M_DLC24	R/W		√		Undefined
xx3FFB05H	CAN message control register 24	M_CTRL24			√		
xx3FFB06H	CAN message time stamp register 24	M_TIME24				√	
xx3FFB08H	CAN message data register 240	M_DATA240			√		
xx3FFB09H	CAN message data register 241	M_DATA241			√		
xx3FFB0AH	CAN message data register 242	M_DATA242			√		
xx3FFB0BH	CAN message data register 243	M_DATA243			√		
xx3FFB0CH	CAN message data register 244	M_DATA244			√		
xx3FFB0DH	CAN message data register 245	M_DATA245			√		

(10/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FFB0EH	CAN message data register 246	M_DATA246	R/W		√		Undefined
xx3FFB0FH	CAN message data register 247	M_DATA247			√		
xx3FFB10H	CAN message ID register L24	M_IDL24				√	
xx3FFB12H	CAN message ID register H24	M_IDH24				√	
xx3FFB14H	CAN message configuration register 24	M_CONF24			√		
xx3FFB15H	CAN message status register 24	M_STAT24	R		√		
xx3FFB16H	CAN status set/clear register 24	SC_STAT24	W			√	0000H
xx3FFB24H	CAN message data length register 25	M_DLC25	R/W		√		Undefined
xx3FFB25H	CAN message control register 25	M_CTRL25			√		
xx3FFB26H	CAN message time stamp register 25	M_TIME25				√	
xx3FFB28H	CAN message data register 250	M_DATA250			√		
xx3FFB29H	CAN message data register 251	M_DATA251			√		
xx3FFB2AH	CAN message data register 252	M_DATA252			√		
xx3FFB2BH	CAN message data register 253	M_DATA253			√		
xx3FFB2CH	CAN message data register 254	M_DATA254			√		
xx3FFB2DH	CAN message data register 255	M_DATA255			√		
xx3FFB2EH	CAN message data register 256	M_DATA256			√		
xx3FFB2FH	CAN message data register 257	M_DATA257			√		
xx3FFB30H	CAN message ID register L25	M_IDL25				√	
xx3FFB32H	CAN message ID register H25	M_IDH25				√	
xx3FFB34H	CAN message configuration register 25	M_CONF25			√		
xx3FFB35H	CAN message status register 25	M_STAT25	R		√		
xx3FFB36H	CAN status set/clear register 25	SC_STAT25	W			√	0000H
xx3FFB44H	CAN message data length register 26	M_DLC26	R/W		√		Undefined
xx3FFB45H	CAN message control register 26	M_CTRL26			√		
xx3FFB46H	CAN message time stamp register 26	M_TIME26				√	
xx3FFB48H	CAN message data register 260	M_DATA260			√		
xx3FFB49H	CAN message data register 261	M_DATA261			√		
xx3FFB4AH	CAN message data register 262	M_DATA262			√		
xx3FFB4BH	CAN message data register 263	M_DATA263			√		
xx3FFB4CH	CAN message data register 264	M_DATA264			√		
xx3FFB4DH	CAN message data register 265	M_DATA265			√		
xx3FFB4EH	CAN message data register 266	M_DATA266			√		
xx3FFB4FH	CAN message data register 267	M_DATA267			√		
xx3FFB50H	CAN message ID register L26	M_IDL26				√	
xx3FFB52H	CAN message ID register H26	M_IDH26				√	
xx3FFB54H	CAN message configuration register 26	M_CONF26			√		
xx3FFB55H	CAN message status register 26	M_STAT26	R		√		
xx3FFB56H	CAN status set/clear register 26	SC_STAT26	W			√	0000H
xx3FFB64H	CAN message data length register 27	M_DLC27	R/W		√		Undefined
xx3FFB65H	CAN message control register 27	M_CTRL27			√		
xx3FFB66H	CAN message time stamp register 27	M_TIME27				√	
xx3FFB68H	CAN message data register 270	M_DATA270			√		
xx3FFB69H	CAN message data register 271	M_DATA271			√		

(11/13)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
xx3FFB6AH	CAN message data register 272	M_DATA272	R/W		√		Undefined
xx3FFB6BH	CAN message data register 273	M_DATA273			√		
xx3FFB6CH	CAN message data register 274	M_DATA274			√		
xx3FFB6DH	CAN message data register 275	M_DATA275			√		
xx3FFB6EH	CAN message data register 276	M_DATA276			√		
xx3FFB6FH	CAN message data register 277	M_DATA277			√		
xx3FFB70H	CAN message ID register L27	M_IDL27				√	
xx3FFB72H	CAN message ID register H27	M_IDH27				√	
xx3FFB74H	CAN message configuration register 27	M_CONF27			√		
xx3FFB75H	CAN message status register 27	M_STAT27	R		√		
xx3FFB76H	CAN status set/clear register 27	SC_STAT27	W			√	0000H
xx3FFB84H	CAN message data length register 28	M_DLC28	R/W		√		Undefined
xx3FFB85H	CAN message control register 28	M_CTRL28			√		
xx3FFB86H	CAN message time stamp register 28	M_TIME28				√	
xx3FFB88H	CAN message data register 280	M_DATA280			√		
xx3FFB89H	CAN message data register 281	M_DATA281			√		
xx3FFB8AH	CAN message data register 282	M_DATA282			√		
xx3FFB8BH	CAN message data register 283	M_DATA283			√		
xx3FFB8CH	CAN message data register 284	M_DATA284			√		
xx3FFB8DH	CAN message data register 285	M_DATA285			√		
xx3FFB8EH	CAN message data register 286	M_DATA286			√		
xx3FFB8FH	CAN message data register 287	M_DATA287			√		
xx3FFB90H	CAN message ID register L28	M_IDL28				√	
xx3FFB92H	CAN message ID register H28	M_IDH28				√	
xx3FFB94H	CAN message configuration register 28	M_CONF28			√		
xx3FFB95H	CAN message status register 28	M_STAT28	R		√		
xx3FFB96H	CAN status set/clear register 28	SC_STAT28	W			√	0000H
xx3FFBA4H	CAN message data length register 29	M_DLC29	R/W		√		Undefined
xx3FFBA5H	CAN message control register 29	M_CTRL29			√		
xx3FFBA6H	CAN message time stamp register 29	M_TIME29				√	
xx3FFBA8H	CAN message data register 290	M_DATA290			√		
xx3FFBA9H	CAN message data register 291	M_DATA291			√		
xx3FFBAAH	CAN message data register 292	M_DATA292			√		
xx3FFBABH	CAN message data register 293	M_DATA293			√		
xx3FFBACH	CAN message data register 294	M_DATA294			√		
xx3FFBADH	CAN message data register 295	M_DATA295			√		
xx3FFBAEH	CAN message data register 296	M_DATA296			√		
xx3FFBAFH	CAN message data register 297	M_DATA297			√		
xx3FFBB0H	CAN message ID register L29	M_IDL29				√	
xx3FFBB2H	CAN message ID register H29	M_IDH29				√	
xx3FFBB4H	CAN message configuration register 29	M_CONF29			√		
xx3FFBB5H	CAN message status register 29	M_STAT29	R		√		
xx3FFBB6H	CAN status set/clear register 29	SC_STAT29	W			√	0000H
xx3FFBC4H	CAN message data length register 30	M_DLC30	R/W		√		Undefined

(12/13)

	Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
	xx3FFBC5H	CAN message control register 30	M_CTRL30	R/W		√		Undefined
	xx3FFBC6H	CAN message time stamp register 30	M_TIME30				√	
	xx3FFBC8H	CAN message data register 300	M_DATA300			√		
	xx3FFBC9H	CAN message data register 301	M_DATA301			√		
	xx3FFBCAH	CAN message data register 302	M_DATA302			√		
	xx3FFBCBH	CAN message data register 303	M_DATA303			√		
	xx3FFBCCH	CAN message data register 304	M_DATA304			√		
	xx3FFBCDH	CAN message data register 305	M_DATA305			√		
	xx3FFBCEH	CAN message data register 306	M_DATA306			√		
	xx3FFBCFH	CAN message data register 307	M_DATA307			√		
	xx3FFBD0H	CAN message ID register L30	M_IDL30				√	
	xx3FFBD2H	CAN message ID register H30	M_IDH30				√	
	xx3FFBD4H	CAN message configuration register 30	M_CONF30			√		
	xx3FFBD5H	CAN message status register 30	M_STAT30	R		√		
	xx3FFBD6H	CAN status set/clear register 30	SC_STAT30	W			√	0000H
	xx3FFBE4H	CAN message data length register 31	M_DLC31	R/W		√		Undefined
	xx3FFBE5H	CAN message control register 31	M_CTRL31			√		
	xx3FFBE6H	CAN message time stamp register 31	M_TIME31				√	
	xx3FFBE8H	CAN message data register 310	M_DATA310			√		
	xx3FFBE9H	CAN message data register 311	M_DATA311			√		
	xx3FFBEAH	CAN message data register 312	M_DATA312			√		
	xx3FFBEBH	CAN message data register 313	M_DATA313			√		
	xx3FFBECH	CAN message data register 314	M_DATA314			√		
	xx3FFBEDH	CAN message data register 315	M_DATA315			√		
	xx3FFBEEH	CAN message data register 316	M_DATA316			√		
	xx3FFBEFH	CAN message data register 317	M_DATA317			√		
	xx3FFBF0H	CAN message ID register L31	M_IDL31				√	
	xx3FFBF2H	CAN message ID register H31	M_IDH31				√	
	xx3FFBF4H	CAN message configuration register 31	M_CONF31			√		
	xx3FFBF5H	CAN message status register 31	M_STAT31	R		√		
	xx3FFBF6H	CAN status set/clear register 31	SC_STAT31	W			√	0000H
	xx3FFC00H	CAN stop register	CSTOP	R/W			√	
	xx3FFC04H	CAN interrupt pending register	CCINTP	R			√	
★	xx3FFC10H	CAN global status register	CGST	R/W			√	
★	xx3FFC12H	CAN global interrupt enable register	CGIE				√	
	xx3FFC14H	CAN main clock select register	CGCS				√	7F05H
	xx3FFC18H	CAN time stamp count register	CGTSC	R			√	0000H
	xx3FFC1AH	CAN message search start register	CGMSS	W			√	
		CAN message search result register	CGMSR	R			√	
★	xx3FFC20H	CAN global interrupt pending register	CGINTP	R/W		√	√	0000H
★	xx3FFC22H	CAN1 interrupt pending register	C1INTP			√	√	
★	xx3FFC24H	CAN2 interrupt pending register[Note]	C2INTP			√	√	

Note μPD703089Y and 70F3089Y only

(13/13)

	Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
	xx3FFC40H	CAN1 address mask 0 register L	C1MASKL0	R/W			√	Undefined
	xx3FFC42H	CAN1 address mask 0 register H	C1MASKH0				√	
	xx3FFC44H	CAN1 address mask 1 register L	C1MASKL1				√	
	xx3FFC46H	CAN1 address mask 1 register H	C1MASKH1				√	
	xx3FFC48H	CAN1 address mask 2 register L	C1MASKL2				√	
	xx3FFC4AH	CAN1 address mask 2 register H	C1MASKH2				√	
	xx3FFC4CH	CAN1 address mask 3 register L	C1MASKL3				√	
	xx3FFC4EH	CAN1 address mask 3 register H	C1MASKH3				√	
	xx3FFC50H	CAN1 control register	C1CTRL				√	0101H
★	xx3FFC52H	CAN1 definition register	C1DEF				√	0000H
	xx3FFC54H	CAN1 information register	C1LAST	R			√	00FFH
	xx3FFC56H	CAN1 error count register	C1ERC				√	0000H
★	xx3FFC58H	CAN1 interrupt enable register	C1IE	R/W			√	
	xx3FFC5AH	CAN1 bus active register	C1BA	R			√	00FFH
	xx3FFC5CH	CAN1 bit rate prescaler register	C1BRP	R/W			√	0000H
		CAN1 bus diagnostic information register	C1DINF	R			√	
	xx3FFC5EH	CAN1 synchronization control register	C1SYNC	R/W			√	0218H
	xx3FFC80H	CAN2 address mask 0 register L[Note]	C2MASKL0				√	Undefined
	xx3FFC82H	CAN2 address mask 0 register H[Note]	C2MASKH0				√	
	xx3FFC84H	CAN2 address mask 1 register L[Note]	C2MASKL1				√	
	xx3FFC86H	CAN2 address mask 1 register H[Note]	C2MASKH1				√	
	xx3FFC88H	CAN2 address mask 2 register L[Note]	C2MASKL2				√	
	xx3FFC8AH	CAN2 address mask 2 register H[Note]	C2MASKH2				√	
	xx3FFC8CH	CAN2 address mask 3 register L[Note]	C2MASKL3				√	
	xx3FFC8EH	CAN2 address mask 3 register H[Note]	C2MASKH3				√	
	xx3FFC90H	CAN2 control register[Note]	C2CTRL				√	0101H
★	xx3FFC92H	CAN2 definition register[Note]	C2DEF				√	0000H
	xx3FFC94H	CAN2 information register[Note]	C2LAST	R			√	00FFH
	xx3FFC96H	CAN2 error count register[Note]	C2ERC				√	0000H
★	xx3FFC98H	CAN2 interrupt enable register[Note]	C2IE	R/W			√	
	xx3FFC9AH	CAN2 bus active register[Note]	C2BA	R			√	00FFH
	xx3FFC9CH	CAN2 bit rate prescaler register[Note]	C2BRP	R/W			√	0000H
		CAN2 bus diagnostic information register[Note]	C2DINF	R			√	
	xx3FFC9EH	CAN2 synchronization control register[Note]	C2SYNC	R/W			√	0218H

Note μPD703089Y and 70F3089Y only

19.5 Control Registers

19.5.1 CAN message data length registers 00 to 31 (M_DLC00 to M_DLC31)

The M_DLCn register sets the byte count in the data field of CAN message buffer n (n = 00 to 31). When receiving, the receive data field's byte count is set (1).

These registers can be read/written in 8-bit units.

After reset: Undefined R/W Address: See **Table 19-3**

★

	7	6	5	4	3	2	1	0
M_DLCn	RFU[Note]	RFU[Note]	RFU[Note]	RFU[Note]	DLC3	DLC2	DLC1	DLC0

(n = 00 to 31)

DLC3	DLC2	DLC1	DLC0	Data length code of transmit/receive message
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of DLC value

★ **Note** RFU (Reserved for Future Use) indicates a reserved bit. Always set this bit to 0 when writing the M_DLCn register.

Table 19-3. Addresses of M_DLCn (n = 00 to 31)

Register Name	Address	Register Name	Address
M_DLC00	xx3FF804H	M_DLC16	xx3FFA04H
M_DLC01	xx3FF824H	M_DLC17	xx3FFA24H
M_DLC02	xx3FF844H	M_DLC18	xx3FFA44H
M_DLC03	xx3FF864H	M_DLC19	xx3FFA64H
M_DLC04	xx3FF884H	M_DLC20	xx3FFA84H
M_DLC05	xx3FF8A4H	M_DLC21	xx3FFAA4H
M_DLC06	xx3FF8C4H	M_DLC22	xx3FFAC4H
M_DLC07	xx3FF8E4H	M_DLC23	xx3FFAE4H
M_DLC08	xx3FF904H	M_DLC24	xx3FFB04H
M_DLC09	xx3FF924H	M_DLC25	xx3FFB24H
M_DLC10	xx3FF944H	M_DLC26	xx3FFB44H
M_DLC11	xx3FF964H	M_DLC27	xx3FFB64H
M_DLC12	xx3FF984H	M_DLC28	xx3FFB84H
M_DLC13	xx3FF9A4H	M_DLC29	xx3FFBA4H
M_DLC14	xx3FF9C4H	M_DLC30	xx3FFBC4H
M_DLC15	xx3FF9E4H	M_DLC31	xx3FFBE4H

19.5.2 CAN message control registers 00 to 31 (M_CTRL00 to M_CTRL31)

The M_CTRLn register is used to control operation of CAN message buffer n (n = 00 to 31).

These registers can be read/written in 8-bit units.

After reset: Undefined R/W Address: See **Table 19-4**

★

M_CTRLn (n = 00 to 31)	7	6	5	4	3	2	1	0
	RMDE1	RMDE0	ATS	IE	MOVR	RFU[Notes1, 2]	RFU[Notes1, 3]	RTR

RMDE1	Specifies operation of DN flag when remote frame is received on a transmit message buffer
0	DN flag not set (1) when remote frame is received
1	DN flag set (1) when remote frame is received

- When the RMDE1 bit is set, the setting of the RMDE0 bit is irrelevant.
- If a remote frame is received on the transmit message buffer when the RMDE1 and RMDE0 bits have not been set (1), the CPU is not notified, nor are other operations performed.

RMDE0	Specification of set/clear status of remote frame auto acknowledge function
0	Remote frame auto acknowledge function cleared
1	Remote frame auto acknowledge function set

- The RMDE0 bit's setting is used only for transmit message buffers.
- When the RTR bit has been set (1) (when the receive message or transmit message has a remote frame), the RMDE0 bit is processed as RMDE0 = 0. This prevents transmission of the same remote frame during remote frame reception. If the same remote frame is transmitted under the worst conditions, the bus load reaches 100%.

ATS	Specifies whether or not to add a time stamp when transmitting
0	Time stamp not added when transmitting
1	Time stamp added when transmitting

- The ATS bit is used only for transmit messages.
- When the ATS bit has been set (1) and the data length code specifies at least two bytes, the last two bytes are replaced by a time stamp (see **Table 19-12**). The added time stamp counter value is sent to the bus via the message's SOF. When this occurs, the last two bytes (which are defined as a data field) are ignored.

★ **Notes 1.** RFU (Reserved for Future Use) indicates a reserved bit. Always set this bit to 0 when writing.

2. The value of the r1 bit on the CAN bus is set during reception.

3. The value of the r0 bit on the CAN bus is set during reception.

Remark DN: Bit 2 of M_STATm (see **19.5.7 CAN message status registers 00 to 31 (M_STAT00 to M_STAT31)**.)

(2/2)

IE	Specifies the enable/disable setting for interrupt requests
0	Interrupt requests disabled
1	Interrupt requests enabled

- An interrupt request occurs when the IE bit is 1 under the following conditions.
  - When a message is sent from the transmit message buffer
  - When a message is received by the receive message buffer
  - When a remote frame has been transmitted from the receive message buffer
  - When a remote frame is received by the transmit message buffer when the auto acknowledge function has not been set (RMDE0 bit = 0).
- An interrupt request does not occur when the IE bit is 1 under the following conditions.
  - When a remote frame is received by the transmit message buffer when the auto acknowledge function has been set (RMDE0 bit = 1)
- An interrupt request occurs when the IE bit is 0 under the following conditions.
  - When a remote frame is received by the receive message buffer when the auto acknowledge function has not been set (RMDE0 bit = 0).

MOVR	Message buffer overwrite
0	Overwrite does not occur after DN bit is cleared
1	Overwrite occurs at least once after DN bit is cleared

- An overwrite of the message buffer occurs when the CAN module writes new data to the message buffer or when the DN bit has already been set (1). The MOVR bit is updated each time new data is stored in the message buffer.

RTR	Specification of frame type
0	Data frame transmit/receive
1	Remote frame transmit/receive

- When the RTR bit has been set (1) for a transmit message, a remote frame is transmitted instead of a data frame.

Remark DN: Bit 2 of M_STATm (see **19.5.7 CAN message status registers 00 to 31 (M_STAT00 to M_STAT31)**.)

Table 19-4. Addresses of M_CTRLn (n = 00 to 31)

Register Name	Address	Register Name	Address
M_CTRL00	xx3FF805H	M_CTRL16	xx3FFA05H
M_CTRL01	xx3FF825H	M_CTRL17	xx3FFA25H
M_CTRL02	xx3FF845H	M_CTRL18	xx3FFA45H
M_CTRL03	xx3FF865H	M_CTRL19	xx3FFA65H
M_CTRL04	xx3FF885H	M_CTRL20	xx3FFA85H
M_CTRL05	xx3FF8A5H	M_CTRL21	xx3FFAA5H
M_CTRL06	xx3FF8C5H	M_CTRL22	xx3FFAC5H
M_CTRL07	xx3FF8E5H	M_CTRL23	xx3FFAE5H
M_CTRL08	xx3FF905H	M_CTRL24	xx3FFB05H
M_CTRL09	xx3FF925H	M_CTRL25	xx3FFB25H
M_CTRL10	xx3FF945H	M_CTRL26	xx3FFB45H
M_CTRL11	xx3FF965H	M_CTRL27	xx3FFB65H
M_CTRL12	xx3FF985H	M_CTRL28	xx3FFB85H
M_CTRL13	xx3FF9A5H	M_CTRL29	xx3FFBA5H
M_CTRL14	xx3FF9C5H	M_CTRL30	xx3FFBC5H
M_CTRL15	xx3FF9E5H	M_CTRL31	xx3FFBE5H

19.5.3 CAN message time stamp registers 00 to 31 (M_TIME00 to M_TIME31)

The M_TIMEn register is the area where the time stamp counter value is written upon completion of data reception (n = 00 to 31).

These registers can be read/written in 16-bit units.

When a data frame or remote frame is received in the receive message buffer, the new data is stored in the message buffer and a 16-bit time tag (time stamp counter value) is stored in the M_TIMEn register. This time tag is set according to the FCAN's time stamp setting, which is either the time stamp counter value that was captured when the SOF was sent on the CAN bus or the value captured when the CAN module writes data to the message buffer.

After reset: Undefined R/W Address: See **Table 19-5**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_TIMEn (n = 00 to 31)	TS 15	TS 14	TS 13	TS 12	TS 11	TS 10	TS 09	TS 08	TS 07	TS 06	TS 05	TS 04	TS 03	TS 02	TS 01	TS 00

Table 19-5. Addresses of M_TIMEn (n = 00 to 31)

Register Name	Address	Register Name	Address
M_TIME00	xx3FF806H	M_TIME16	xx3FFA06H
M_TIME01	xx3FF826H	M_TIME17	xx3FFA26H
M_TIME02	xx3FF846H	M_TIME18	xx3FFA46H
M_TIME03	xx3FF866H	M_TIME19	xx3FFA66H
M_TIME04	xx3FF886H	M_TIME20	xx3FFA86H
M_TIME05	xx3FF8A6H	M_TIME21	xx3FFAA6H
M_TIME06	xx3FF8C6H	M_TIME22	xx3FFAC6H
M_TIME07	xx3FF8E6H	M_TIME23	xx3FFAE6H
M_TIME08	xx3FF906H	M_TIME24	xx3FFB06H
M_TIME09	xx3FF926H	M_TIME25	xx3FFB26H
M_TIME10	xx3FF946H	M_TIME26	xx3FFB46H
M_TIME11	xx3FF966H	M_TIME27	xx3FFB66H
M_TIME12	xx3FF986H	M_TIME28	xx3FFB86H
M_TIME13	xx3FF9A6H	M_TIME29	xx3FFBA6H
M_TIME14	xx3FF9C6H	M_TIME30	xx3FFBC6H
M_TIME15	xx3FF9E6H	M_TIME31	xx3FFBE6H

19.5.4 CAN message data registers n0 to n7 (M_DATAn0 to M_DATAn7)

The M_DATAn0 to M_DATAn7 registers are areas where up to 8 bytes of transmit or receive message data is stored.

These registers can be read/written in 8-bit units.

The M_DATAn0 to M_DATAn7 registers are used to hold receive message data and transmit message data. When data is transmitted, the number of messages defined by the DLC3 to DLC0 bits in the M_DLCn register are transmitted via the CAN bus.

When the M_CTRLn register's ATS bit has been set (1) and the value of the DLC3 to DLC0 bits in the M_DLCn register is at least two bytes, the last two bytes that are sent normally via the CAN bus are ignored and the time stamp value is sent.

When a new message is received, all data fields are updated, even when the value of the DLC3 to DLC0 bits in the M_DLCn register is less than 8 bytes. The values of data bytes that have not been received on the CAN bus may be updated, but they are ignored.

Remark n = 00 to 31, x = 0 to 7

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn0	D07	D06	D05	D04	D03	D02	D01	D00	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn1	D17	D16	D15	D14	D13	D12	D11	D10	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn2	D27	D26	D25	D24	D23	D22	D21	D20	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn3	D37	D36	D35	D34	D33	D32	D31	D30	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn4	D47	D46	D45	D44	D43	D42	D41	D40	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn5	D57	D56	D55	D54	D53	D52	D51	D50	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn6	D67	D66	D65	D64	D63	D62	D61	D60	See **Table 19-6**	Undefined

(n = 00 to 31)

	7	6	5	4	3	2	1	0	Address	After reset
M_DATAn7	D77	D76	D75	D74	D73	D72	D71	D70	See **Table 19-6**	Undefined

(n = 00 to 31)

Table 19-6. Addresses of M_DATAnx (n = 00 to 31, x = 0 to 7)

Register Name / n	M_DATAn0	M_DATAn1	M_DATAn2	M_DATAn3	M_DATAn4	M_DATAn5	M_DATAn6	M_DATAn7
00	xx3FF808H	xx3FF809H	xx3FF80AH	xx3FF80BH	xx3FF80CH	xx3FF80DH	xx3FF80EH	xx3FF80FH
01	xx3FF828H	xx3FF829H	xx3FF82AH	xx3FF82BH	xx3FF82CH	xx3FF82DH	xx3FF82EH	xx3FF82FH
02	xx3FF848H	xx3FF849H	xx3FF84AH	xx3FF84BH	xx3FF84CH	xx3FF84DH	xx3FF84EH	xx3FF84FH
03	xx3FF868H	xx3FF869H	xx3FF86AH	xx3FF86BH	xx3FF86CH	xx3FF86DH	xx3FF86EH	xx3FF86FH
04	xx3FF888H	xx3FF889H	xx3FF88AH	xx3FF88BH	xx3FF88CH	xx3FF88DH	xx3FF88EH	xx3FF88FH
05	xx3FF8A8H	xx3FF8A9H	xx3FF8AAH	xx3FF8ABH	xx3FF8ACH	xx3FF8ADH	xx3FF8AEH	xx3FF8AFH
06	xx3FF8C8H	xx3FF8C9H	xx3FF8CAH	xx3FF8CBH	xx3FF8CCH	xx3FF8CDH	xx3FF8CEH	xx3FF8CFH
07	xx3FF8E8H	xx3FF8E9H	xx3FF8EAH	xx3FF8EBH	xx3FF8ECH	xx3FF8EDH	xx3FF8EEH	xx3FF8EFH
08	xx3FF908H	xx3FF909H	xx3FF90AH	xx3FF90BH	xx3FF90CH	xx3FF90DH	xx3FF90EH	xx3FF90FH
09	xx3FF928H	xx3FF929H	xx3FF92AH	xx3FF92BH	xx3FF92CH	xx3FF92DH	xx3FF92EH	xx3FF92FH
10	xx3FF948H	xx3FF949H	xx3FF94AH	xx3FF94BH	xx3FF94CH	xx3FF94DH	xx3FF94EH	xx3FF94FH
11	xx3FF968H	xx3FF969H	xx3FF96AH	xx3FF96BH	xx3FF96CH	xx3FF96DH	xx3FF96EH	xx3FF96FH
12	xx3FF988H	xx3FF989H	xx3FF98AH	xx3FF98BH	xx3FF98CH	xx3FF98DH	xx3FF98EH	xx3FF98FH
13	xx3FF9A8H	xx3FF9A9H	xx3FF9AAH	xx3FF9ABH	xx3FF9ACH	xx3FF9ADH	xx3FF9AEH	xx3FF9AFH
14	xx3FF9C8H	xx3FF9C9H	xx3FF9CAH	xx3FF9CBH	xx3FF9CCH	xx3FF9CDH	xx3FF9CEH	xx3FF9CFH
15	xx3FF9E8H	xx3FF9E9H	xx3FF9EAH	xx3FF9EBH	xx3FF9ECH	xx3FF9EDH	xx3FF9EEH	xx3FF9EFH
16	xx3FFA08H	xx3FFA09H	xx3FFA0AH	xx3FFA0BH	xx3FFA0CH	xx3FFA0DH	xx3FFA0EH	xx3FFA0FH
17	xx3FFA28H	xx3FFA29H	xx3FFA2AH	xx3FFA2BH	xx3FFA2CH	xx3FFA2DH	xx3FFA2EH	xx3FFA2FH
18	xx3FFA48H	xx3FFA49H	xx3FFA4AH	xx3FFA4BH	xx3FFA4CH	xx3FFA4DH	xx3FFA4EH	xx3FFA4FH
19	xx3FFA68H	xx3FFA69H	xx3FFA6AH	xx3FFA6BH	xx3FFA6CH	xx3FFA6DH	xx3FFA6EH	xx3FFA6FH
20	xx3FFA88H	xx3FFA89H	xx3FFA8AH	xx3FFA8BH	xx3FFA8CH	xx3FFA8DH	xx3FFA8EH	xx3FFA8FH
21	xx3FFAA8H	xx3FFAA9H	xx3FFAAAH	xx3FFAABH	xx3FFAACH	xx3FFAADH	xx3FFAAEH	xx3FFAAFH
22	xx3FFAC8H	xx3FFAC9H	xx3FFACAH	xx3FFACBH	xx3FFACCH	xx3FFACDH	xx3FFACEH	xx3FFACFH
23	xx3FFAE8H	xx3FFAE9H	xx3FFAEAH	xx3FFAEBH	xx3FFAECH	xx3FFAEDH	xx3FFAEEH	xx3FFAEFH
24	xx3FFB08H	xx3FFB09H	xx3FFB0AH	xx3FFB0BH	xx3FFB0CH	xx3FFB0DH	xx3FFB0EH	xx3FFB0FH
25	xx3FFB28H	xx3FFB29H	xx3FFB2AH	xx3FFB2BH	xx3FFB2CH	xx3FFB2DH	xx3FFB2EH	xx3FFB2FH
26	xx3FFB48H	xx3FFB49H	xx3FFB4AH	xx3FFB4BH	xx3FFB4CH	xx3FFB4DH	xx3FFB4EH	xx3FFB4FH
27	xx3FFB68H	xx3FFB69H	xx3FFB6AH	xx3FFB6BH	xx3FFB6CH	xx3FFB6DH	xx3FFB6EH	xx3FFB6FH
28	xx3FFB88H	xx3FFB89H	xx3FFB8AH	xx3FFB8BH	xx3FFB8CH	xx3FFB8DH	xx3FFB8EH	xx3FFB8FH
29	xx3FFBA8H	xx3FFBA9H	xx3FFBAAH	xx3FFBABH	xx3FFBACH	xx3FFBADH	xx3FFBAEH	xx3FFBAFH
30	xx3FFBC8H	xx3FFBC9H	xx3FFBCAH	xx3FFBCBH	xx3FFBCCH	xx3FFBCDH	xx3FFBCEH	xx3FFBCFH
31	xx3FFBE8H	xx3FFBE9H	xx3FFBEAH	xx3FFBEBH	xx3FFBECH	xx3FFBEDH	xx3FFBEEH	xx3FFBEFH

19.5.5 CAN message ID registers L00 to L31 and H00 to H31 (M_IDL00 to M_IDL31 and M_IDH00 to M_IDH31)

The M_IDLn and M_IDHn registers are areas used to set identifiers (n = 00 to 31).

These registers can be read/written in 16-bit units.

When in standard format mode, any data can be stored in the following areas.

ID17 to ID10: First byte of receive data[Note] is stored.

ID9 to ID12: Second byte of receive data[Note] is stored.

ID1, ID0: Third byte (higher two bits) of receive data[Note] is stored.

Note See **19.5.4 CAN message data registers n0 to n7 (M_DATAn0 to M_DATAn7)**.

After reset: Undefined R/W Address: See **Table 19-7**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_IDHn (n = 00 to 31)	IDE	0	0	ID 28	ID 27	ID 26	ID 25	ID 24	ID 23	ID 22	ID 21	ID 20	ID 19	ID 18	ID 17	ID 16

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_IDLn (n = 00 to 31)	ID 15	ID 14	ID 13	ID 12	ID 11	ID 10	ID 9	ID 8	ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID 0

IDE	Specification of format setting mode
0	Standard format mode (ID28 to ID18: 11 bits)
1	Extended format mode (ID28 to ID0: 29 bits)

Table 19-7. Addresses of M_IDLn and M_IDHn (n = 00 to 31)

Register Name	Address	Register Name	Address
M_IDL00	xx3FF810H	M_IDL16	xx3FFA10H
M_IDH00	xx3FF812H	M_IDH16	xx3FFA12H
M_IDL01	xx3FF830H	M_IDL17	xx3FFA30H
M_IDH01	xx3FF832H	M_IDH17	xx3FFA32H
M_IDL02	xx3FF850H	M_IDL18	xx3FFA50H
M_IDH02	xx3FF852H	M_IDH18	xx3FFA52H
M_IDL03	xx3FF870H	M_IDL19	xx3FFA70H
M_IDH03	xx3FF872H	M_IDH19	xx3FFA72H
M_IDL04	xx3FF890H	M_IDL20	xx3FFA90H
M_IDH04	xx3FF892H	M_IDH20	xx3FFA92H
M_IDL05	xx3FF8B0H	M_IDL21	xx3FFAB0H
M_IDH05	xx3FF8B2H	M_IDH21	xx3FFAB2H
M_IDL06	xx3FF8D0H	M_IDL22	xx3FFAD0H
M_IDH06	xx3FF8D2H	M_IDH22	xx3FFAD2H
M_IDL07	xx3FF8F0H	M_IDL23	xx3FFAF0H
M_IDH07	xx3FF8F2H	M_IDH23	xx3FFAF2H
M_IDL08	xx3FF910H	M_IDL24	xx3FFB10H
M_IDH08	xx3FF912H	M_IDH24	xx3FFB12H
M_IDL09	xx3FF930H	M_IDL25	xx3FFB30H
M_IDH09	xx3FF932H	M_IDH25	xx3FFB32H
M_IDL10	xx3FF950H	M_IDL26	xx3FFB50H
M_IDH10	xx3FF952H	M_IDH26	xx3FFB52H
M_IDL11	xx3FF970H	M_IDL27	xx3FFB70H
M_IDH11	xx3FF972H	M_IDH27	xx3FFB72H
M_IDL12	xx3FF990H	M_IDL28	xx3FFB90H
M_IDH12	xx3FF992H	M_IDH28	xx3FFB92H
M_IDL13	xx3FF9B0H	M_IDL29	xx3FFBB0H
M_IDH13	xx3FF9B2H	M_IDH29	xx3FFBB2H
M_IDL14	xx3FF9D0H	M_IDL30	xx3FFBD0H
M_IDH14	xx3FF9D2H	M_IDH30	xx3FFBD2H
M_IDL15	xx3FF9F0H	M_IDL31	xx3FFBF0H
M_IDH15	xx3FF9F2H	M_IDH31	xx3FFBF2H

19.5.6 CAN message configuration registers 00 to 31 (M_CONF00 to M_CONF31)

The M_CONFn register is used to specify the message buffer type and mask setting (n = 00 to 31).

These registers can be read/written in 8-bit units.

After reset: Undefined R/W Address: See **Table 19-8**

	7	6	5	4	3	2	1	0
M_CONFn (n = 00 to 31)	0	0	MT2	MT1	MT0	MA2	MA1	MA0

MT2	MT1	MT0	Specification of message type and mask setting
0	0	0	Transmit message
0	0	1	Receive message (no mask setting)
0	1	0	Receive message (mask 0 is set)
0	1	1	Receive message (mask 1 is set)
1	0	0	Receive message (mask 2 is set)
1	0	1	Receive message (mask 3 is set)
1	1	0	Setting prohibited
1	1	1	Receive message (used in diagnostic processing mode)

- When bits MT2 to MT0 have been set as “111”, processing can be performed only when the FCAN has been set to diagnostic processing mode. In such cases, all messages received are stored regardless of the following conditions.
  - Storage to other message buffer
  - Identifier type (standard frame or extended frame)
  - Data frame or remote frame

MA2	MA1	MA0	Link settings of message buffer and CAN module
0	0	0	Message buffer is not used
0	0	1	Used as CAN1 module’s message buffer
0	1	0	Used as CAN2 module’s message buffer[Note]
Other than above			Setting prohibited

- When bits MA2, MA1, and MA0 have been set to “000”, message buffer area is used for application RAM or for event processing as a temporary buffer.
- ★ In the unused message buffers, always set the MA2, MA1, and MA0 bits to 000.

Note μPD703089Y and 70F3089Y only

Table 19-8. Addresses of M_CONFn (n = 00 to 31)

Register Name	Address	Register Name	Address
M_CONF00	xx3FF814H	M_CONF16	xx3FFA14H
M_CONF01	xx3FF834H	M_CONF17	xx3FFA34H
M_CONF02	xx3FF854H	M_CONF18	xx3FFA54H
M_CONF03	xx3FF874H	M_CONF19	xx3FFA74H
M_CONF04	xx3FF894H	M_CONF20	xx3FFA94H
M_CONF05	xx3FF8B4H	M_CONF21	xx3FFAB4H
M_CONF06	xx3FF8D4H	M_CONF22	xx3FFAD4H
M_CONF07	xx3FF8F4H	M_CONF23	xx3FFAF4H
M_CONF08	xx3FF914H	M_CONF24	xx3FFB14H
M_CONF09	xx3FF934H	M_CONF25	xx3FFB34H
M_CONF10	xx3FF954H	M_CONF26	xx3FFB54H
M_CONF11	xx3FF974H	M_CONF27	xx3FFB74H
M_CONF12	xx3FF994H	M_CONF28	xx3FFB94H
M_CONF13	xx3FF9B4H	M_CONF29	xx3FFBB4H
M_CONF14	xx3FF9D4H	M_CONF30	xx3FFBD4H
M_CONF15	xx3FF9F4H	M_CONF31	xx3FFBF4H

19.5.7 CAN message status registers 00 to 31 (M_STAT00 to M_STAT31)

The M_STATn register indicates the transmit/receive status information of each message buffer (n = 00 to 31). These registers are read-only, in 8-bit units.

Cautions 1. Writing directly to M_STATn register is not possible. Use CAN status set/clear register n (SC_STATn).

2. Messages are transmitted only when the M_STATn registers' TRQ and RDY bits have been set (1).

After reset: Undefined R Address: See **Table 19-9**

	7	6	5	4	3	2	1	0
★ M_STATn (n = 00 to 31)	0	0	0	0	RFU[Note]	DN	TRQ	RDY

DN	Message update flag
0	No message was received after DN bit was cleared
1	At least one message was received after DN bit was cleared
• When the DN bit has been set (1) by the transmit message buffer, it indicates that the message buffer has received a remote frame. When this message is sent, the DN bit is automatically cleared (0). • When a frame is again received in the message buffer for which the DN bit has been set (1), an overwrite condition occurs and the M_CTRLn register's MOVR bit is set (1).	

TRQ	Transmit request flag
0	Message transmission prohibited
1	Message transmission enabled
★ • A transmit request is processed as a CAN module only when the RDY bit is set to 1. ★ • A remote frame is transmitted to the receive message buffer in which the TRQ bit is set to 1.	

RDY	Message ready flag
0	Message is not ready
1	Message is ready
• A receive operation is performed only for a message buffer in which the RDY bit is set to 1 during reception. • A transmit operation is performed only for a message buffer in which the RDY bit is set to 1 and the TRQ bit is set to 1 during transmission.	

★ **Note** RFU (Reserved for Future Use) indicates a reserved bit. 0 or 1 is read from this bit regardless of the message buffer setting.

Table 19-9. Addresses of M_STATn (n = 00 to 31)

Register Name	Address	Register Name	Address
M_STAT00	xx3FF815H	M_STAT16	xx3FFA15H
M_STAT01	xx3FF835H	M_STAT17	xx3FFA35H
M_STAT02	xx3FF855H	M_STAT18	xx3FFA55H
M_STAT03	xx3FF875H	M_STAT19	xx3FFA75H
M_STAT04	xx3FF895H	M_STAT20	xx3FFA95H
M_STAT05	xx3FF8B5H	M_STAT21	xx3FFAB5H
M_STAT06	xx3FF8D5H	M_STAT22	xx3FFAD5H
M_STAT07	xx3FF8F5H	M_STAT23	xx3FFAF5H
M_STAT08	xx3FF915H	M_STAT24	xx3FFB15H
M_STAT09	xx3FF935H	M_STAT25	xx3FFB35H
M_STAT10	xx3FF955H	M_STAT26	xx3FFB55H
M_STAT11	xx3FF975H	M_STAT27	xx3FFB75H
M_STAT12	xx3FF995H	M_STAT28	xx3FFB95H
M_STAT13	xx3FF9B5H	M_STAT29	xx3FFBB5H
M_STAT14	xx3FF9D5H	M_STAT30	xx3FFBD5H
M_STAT15	xx3FF9F5H	M_STAT31	xx3FFBF5H

19.5.8 CAN status set/clear registers 00 to 31 (SC_STAT00 to SC_STAT31)

The SC_STATn register is used to set/clear the transmit/receive status information (n = 00 to 31).

These registers are write-only, in 16-bit units.

After reset: 0000H W Address: See **Table 19-10**

	15	14	13	12	11	10	9	8
SC_STATn (n = 00 to 31)	0	0	0	0	0	set DN	set TRQ	set RDY
	7	6	5	4	3	2	1	0
	0	0	0	0	0	clear DN	clear TRQ	clear RDY

set DN	clear DN	Message update flag setting
0	1	Clear (Clear (0) DN bit)
1	0	Set (Set (1) DN bit)
Other than above		No change in DN bit value

set TRQ	clear TRQ	Transmit request flag setting
0	1	Clear (Clear (0) TRQ bit)
1	0	Set (Set (1) TRQ bit)
Other than above		No change in TRQ bit value

set RDY	clear RDY	Message ready flag setting
0	1	Clear (Clear (0) RDY bit)
1	0	Set (Set (1) RDY bit)
Other than above		No change in RDY bit value

Remark DN: Bit 2 of CAN message status register n (M_STATn)
TRQ: Bit 1 of CAN message status register n (M_STATn)
RDY: Bit 0 of CAN message status register n (M_STATn)

Table 19-10. Addresses of SC_STATn (n = 00 to 31)

Register name	Address	Register name	Address
SC_STAT00	xx3FF816H	SC_STAT16	xx3FFA16H
SC_STAT01	xx3FF836H	SC_STAT17	xx3FFA36H
SC_STAT02	xx3FF856H	SC_STAT18	xx3FFA56H
SC_STAT03	xx3FF876H	SC_STAT19	xx3FFA76H
SC_STAT04	xx3FF896H	SC_STAT20	xx3FFA96H
SC_STAT05	xx3FF8B6H	SC_STAT21	xx3FFAB6H
SC_STAT06	xx3FF8D6H	SC_STAT22	xx3FFAD6H
SC_STAT07	xx3FF8F6H	SC_STAT23	xx3FFAF6H
SC_STAT08	xx3FF916H	SC_STAT24	xx3FFB16H
SC_STAT09	xx3FF936H	SC_STAT25	xx3FFB36H
SC_STAT10	xx3FF956H	SC_STAT26	xx3FFB56H
SC_STAT11	xx3FF976H	SC_STAT27	xx3FFB76H
SC_STAT12	xx3FF996H	SC_STAT28	xx3FFB96H
SC_STAT13	xx3FF9B6H	SC_STAT29	xx3FFBB6H
SC_STAT14	xx3FF9D6H	SC_STAT30	xx3FFBD6H
SC_STAT15	xx3FF9F6H	SC_STAT31	xx3FFBF6H

19.5.9 CAN interrupt pending register (CCINTP)

The CCINTP register is used to confirm the pending status of various interrupts.

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

After reset: 0000H R Address: xx3FFC04H

	15	14	13	12	11	10	9	8
CCINTP	0	INTMAC	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CAN2ERR	CAN2REC	CAN2TRX	CAN1ERR	CAN1REC	CAN1TRX

INTMAC	Pending status of MAC error[Note 1] interrupts (GINT2, GINT1)
0	Not pending
1	Pending

CAN2ERR[Note 2]	Pending status of CAN2 access error interrupt (C2INT6 to C2INT2)
0	Not pending
1	Pending

CAN2REC[Note 2]	Pending status of CAN2 receive completion interrupt (C2INT1)
0	Not pending
1	Pending

CAN2TRX[Note 2]	Pending status of CAN2 transmit completion interrupt (C2INT0)
0	Not pending
1	Pending

CAN1ERR	Pending status of CAN1 access error interrupt (C1INT6 to C1INT2)
0	Not pending
1	Pending

CAN1REC	Pending status of CAN1 receive completion interrupt (C1INT1)
0	Not pending
1	Pending

CAN1TRX	Pending status of CAN1 transmit completion interrupt (C1INT0)
0	Not pending
1	Pending

Notes 1. MAC (Memory Access Control) errors are errors that are set only when an interrupt source has occurred for the CAN global interrupt pending register (CGINTP).

2. μPD703089Y and 70F3089Y only

Remark GINT2, GINT1: Bits 2 and 1 of the CAN global interrupt pending register (CGINTP)

CnINT6 to CnINT0: Bits 6 to 0 of the CANn interrupt pending register (CnINTP) (n=1, 2)

19.5.10 CAN global interrupt pending register (CGINTP)

The CGINTP register is used to confirm the pending status of MAC access error interrupts.

★ This register can be read/written in 8-bit or 16-bit units.

Cautions 1. When "1" is written to a bit in the CGINTP register, that bit is cleared (0). When "0" is written to it, the bit's value does not change.

2. An interrupt occurs when the corresponding interrupt request is enabled and when no interrupt pending bit has been set (1) for a new interrupt.
The interrupt pending bit can be set (1) only when the interrupt enable bit has been set (1) by the CAN global interrupt enable register (CGIE). However, the interrupt pending bit is not automatically cleared (0) just because the interrupt enable bit has been cleared (0).
Use software processing to clear the interrupt pending bit (0).
Whether the interrupt pending bit (1) is cleared (0) at the appropriate timing or not is controlled by an interrupt service routine. The earlier the interrupt service routine clears (0) the interrupt pending bit, the more quickly the interrupt occurs without losing any new interrupts of the same type.

Remark For details of interrupt sources generated at GINT1 and GINT2, see **19.15.2 Interrupts that occur for global CAN interface**.

★ After reset: 0000H R/W Address: xx3FFC20H

★

	15	14	13	12	11	10	9	8
CGINTP	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	GINT3	GINT2	GINT1	0

GINT3	Pending status of wakeup interrupt from CAN sleep mode with clock supply to FCAN stopped
0	Not pending
1	Pending

★

GINT2	Pending status of CAN module register[Note] write access error interrupt when GOM bit is 0 or pending status of temporary buffer write access error interrupt when GOM bit is 1
0	Not pending
1	Pending

★

GINT1	Pending status of invalid global macro shutdown occurrence interrupt or pending status of unusable memory address access error interrupt
0	Not pending
1	Pending

Note Register with a name starting with "Cn" (n = 1, 2)

Remark GOM: Bit 0 of CAN global status register (CGST)

19.5.11 CANn interrupt pending register (CnINTP)

The CnINTP register is used to confirm the pending status of interrupts issued to the FCAN.

★ This register can be read/written in 8-bit or 16-bit units.

The CAN2 interrupt pending register (C2INTP) is valid only in models μPD703089Y and 70F3089Y.

Cautions 1. When "1" is written to a bit in the CnINTP register, that bit is cleared (0). When "0" is written to it, the bit's value does not change.

2. An interrupt occurs when the corresponding interrupt request is enabled and when no interrupt pending bit has been set (1) for a new interrupt.

The interrupt pending bit can be set (1) only when the interrupt ready bit has been set (1) by the CANn interrupt enable register (CnIE). However, the interrupt pending bit is not automatically cleared (0) just because the interrupt enable bit has been cleared (0). Use software processing to clear the interrupt pending bit (0).

Whether the interrupt pending bit (1) is cleared (0) at the appropriate timing or not is controlled by an interrupt service routine. The earlier the interrupt service routine clears (0) the interrupt pending bit, the more quickly the interrupt occurs without losing any new interrupts of the same type.

Remark n = 1, 2

★ After reset: 0000H R/W Addresses: C1INTP: xx3FFC22H
C2INTP: xx3FFC24H

	15	14	13	12	11	10	9	8
CnINTP	0	0	0	0	0	0	0	0
(n = 1, 2)	7	6	5	4	3	2	1	0
	0	CnINT6	CnINT5	CnINT4	CnINT3	CnINT2	CnINT1	CnINT0

CnINT6	Pending status of CAN module error interrupt
0	Not pending
1	Pending

CnINT5	Pending status of CAN bus error interrupt
0	Not pending
1	Pending

CnINT4	Pending status of wakeup interrupt (from CAN sleep mode)
0	Not pending
1	Pending

CnINT3	Pending status of CAN receive error passive status interrupt
0	Not pending
1	Pending

CnINT2	Pending status of CAN transmit error passive or bus off status interrupt
0	Not pending
1	Pending

CnINT1	Pending status of CAN receive completion interrupt
0	Not pending
1	Pending

CnINT0	Pending status of CAN transmit completion interrupt
0	Not pending
1	Pending

19.5.12 CAN stop register (CSTOP)

The CSTOP register controls clock supply to the entire CAN system.

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

Cautions 1. Be sure to set the CSTP bit (1) if the FCAN function will not be used.

2. When the CSTP bit is set (1), access to FCAN registers other than the CSTOP register is prohibited. Access to FCAN (other than the CSTOP register) is possible only when the CSTP bit is cleared (0). If accessed while the CSTP bit is set (1), undefined value is read and writing is not possible.

3. When the CSTP bit is set (1), wakeup from the CAN sleep mode (SLEEP bit of CANn control register (CnCTRL) = 1) can be performed in accordance with a change on the CAN bus.

★ **4. If the CAN main clock (f_{MEM1}) is stopped in other than CAN sleep mode, first set the CAN module to initial mode (INIT bit of CnCTRL register = 1), clear (0) the GOM bit of the CGST register, and then set (1) the CSTP bit.**

After reset: 0000H R/W Address: xx3FFC00H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSTOP	CSTP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSTP	Controls clock supply to FCAN
0	FCAN in operation (supplies clock to FCAN blocks)
1	FCAN is stopped (access to FCAN blocks is not possible)

19.5.13 CAN global status register (CGST)

The CGST register indicates global status information.

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

Cautions 1. Both bitwise writing and direct writing to the CGST register are prohibited. Attempts to write directly to this register may result in operation faults, so be sure to follow the sequence described in 19.6 Cautions Regarding Bit Set/Clear Function.

2. When writing to the CGST register, set or clear bits according to the register configuration shown in part (b) Write of the following figure.

(1/2)

After reset: 0000H R/W Address: xx3FFC10H

(a) Read	15	14	13	12	11	10	9	8
CGST	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	MERR	0	0	0	EFSD	TSM	0	GOM

(b) Write	15	14	13	12	11	10	9	8
CGST	0	0	0	0	set EFSD	set TSM	0	set GOM
	7	6	5	4	3	2	1	0
	clear MERR	0	0	0	clear EFSD	clear TSM	0	clear GOM

(a) Read

MERR	MAC error status flag
0	Error does not occur after the MERR bit has been cleared
1	Error occurs at least once after MERR bit has been cleared

★

- MAC errors occur under the following conditions.
  - When invalid address is accessed
  - When access prohibited by MAC is performed
  - When the GOM bit is cleared (0) before the INIT bit of the CnCTRL register is set (1)

EFSD	Shutdown request
0	Shutdown prohibited
1	Shutdown enabled

- Be sure to set the EFSD bit (1) before clearing the GOM bit (0) (must be accessed twice). The EFSD bit will be cleared (0) automatically when the CGST register is accessed again.

TSM	Operation status of time stamp counter[Note]
0	Time stamp counter is stopped
1	Time stamp counter is operating

Note Refer to **19.5.16 CAN time stamp count register (CGTSC)**.

(2/2)

(a) Read

GOM	Status of global operation mode
0	CAN module is reset and access to CAN module register[Note 1] is prohibited
1	CAN module operation is enabled and access to CAN module register[Note 1] is enabled

- The GOM bit controls the method the memory is accessed by the MAC and CAN module operation status.
  - When GOM bit = 0
    - All CAN modules are reset
    - Access to CAN module register disabled (if accessed, a MAC error interrupt occurs)[Note 2]
    - Access to temporary buffer enabled
    - Access to message buffer area enabled
  - When GOM bit = 1
    - Access to CAN module register enabled[Note 3]
    - Access to temporary buffer prohibited (if accessed, a MAC error interrupt occurs)
    - Access to message buffer area enabled
- The GOM bit is cleared (0) only when all the CAN modules are in the initial mode (the INIT bit of the CnCTRL register is 1). Even if the GOM bit is cleared when there is a CAN module not in the initial mode, the GOM bit remains set (1).
- ★ To clear (0) the GOM bit, first set (1) the INIT bit of the CnCTRL register, and then set (1) the EFSD bit. Do not manipulate the GOM bit and EFSD bit simultaneously.

Notes 1. Register with a name starting with “Cn” (n = 1, 2)

★ **2.** The CGCS register can be accessed.
Write accessing the CGMSS register is prohibited. If the CGMSS register is write-accessed, the wrong search result is reflected in the CGMSR register.

★ **3.** Write accessing the CGCS register is prohibited.
Write accessing the CGMSS register is possible.

(b) Write

set EFSD	clear EFSD	EFSD bit setting
0	1	EFSD bit cleared (0)
1	0	EFSD bit set (1)
Other than above		No change in EFSD bit's value

set TSM	clear TSM	TSM bit setting
0	1	TSM bit cleared (0)
1	0	TSM bit set (1)
Other than above		No change in TSM bit's value

set GOM	clear GOM	GOM bit setting
0	1	GOM bit cleared (0)
1	0	GOM bit set (1)
Other than above		No change in GOM bit's value

clear MERR	MERR bit setting
0	No change in MERR bit's value
1	MERR bit cleared (1)

19.5.14 CAN global interrupt enable register (CGIE)

The CGIE register is used to issue interrupt requests for global interrupts.

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

Cautions 1. Both bitwise writing and direct writing to the CGIE register are prohibited. Attempts to write directly to this register may result in operation faults, so be sure to follow the sequence described in 19.6 Cautions Regarding Bit Set/Clear Function.

2. When writing to the CGIE register, set or clear bits according to the register configuration shown in part (b) Write of the following figure.

After reset: 0000H R/W Address: xx3FFC12H

(a) Read

CGIE	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	G_IE2	G_IE1	0

(b) Write

CGIE	15	14	13	12	11	10	9	8
	0	0	0	0	0	set G_IE2	set G_IE1	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	clear G_IE2	clear G_IE1	0

(a) Read

★

G_IE2	Write access error interrupt enable status for CAN module register[Note] when GOM bit is 0 or write access error interrupt enable status for temporary buffer when the GOM bit is 1
0	Interrupt disabled
1	Interrupt enabled

★

G_IE1	Invalid global macro shutdown occurrence interrupt enable status or unusable memory address access error interrupt enable status
0	Interrupt disabled
1	Interrupt enabled

Note Register with a name starting with “Cn” (n = 1, 2)

(b) Write

set G_IEn	clear G_IEn	Setting of G_IEn bit
0	1	Clear (0) G_IEn bit
1	0	Set (1) G_IEn bit
Other than above		No change

Remarks 1. n = 1, 2

2. GOM: Bit 0 of the CAN global status register (CGST)

19.5.15 CAN main clock select register (CGCS)

The CGCS register is used to select the CAN main clock.

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

★ **Caution When the GOM bit of the CGST register is 1, write accessing the CGCS register is prohibited.**

After reset: 7F05H R/W Address: xx3FFC14H

	15	14	13	12	11	10	9	8
CGCS	CGTS7	CGTS6	CGTS5	CGTS4	CGTS3	CGTS2	CGTS1	CGTS0
	7	6	5	4	3	2	1	0
	GTCS1	GTCS0	0	0[Note 1]	MCP3	MCP2	MCP1	MCP0

n	CGTS 7	CGTS 6	CGTS 5	CGTS 4	CGTS 3	CGTS 2	CGTS 1	CGTS 0	System timer prescaler selection $f_{GTS} = f_{GTS1}/(n + 1)$
0	0	0	0	0	0	0	0	0	$f_{GTS} = f_{GTS1}/1$
1	0	0	0	0	0	0	0	1	$f_{GTS} = f_{GTS1}/2$
					:				$f_{GTS} = f_{GTS1}/(n + 1)$
127	0	1	1	1	1	1	1	1	$f_{GTS} = f_{GTS1}/128$ (after reset)
					:				$f_{GTS} = f_{GTS1}/(n + 1)$
254	1	1	1	1	1	1	1	0	$f_{GTS} = f_{GTS1}/255$
255	1	1	1	1	1	1	1	1	$f_{GTS} = f_{GTS1}/256$
The global timer system clock (f_{GTS}) is the source clock for the time stamp counter[Note 2] that is used for the time stamp function.									

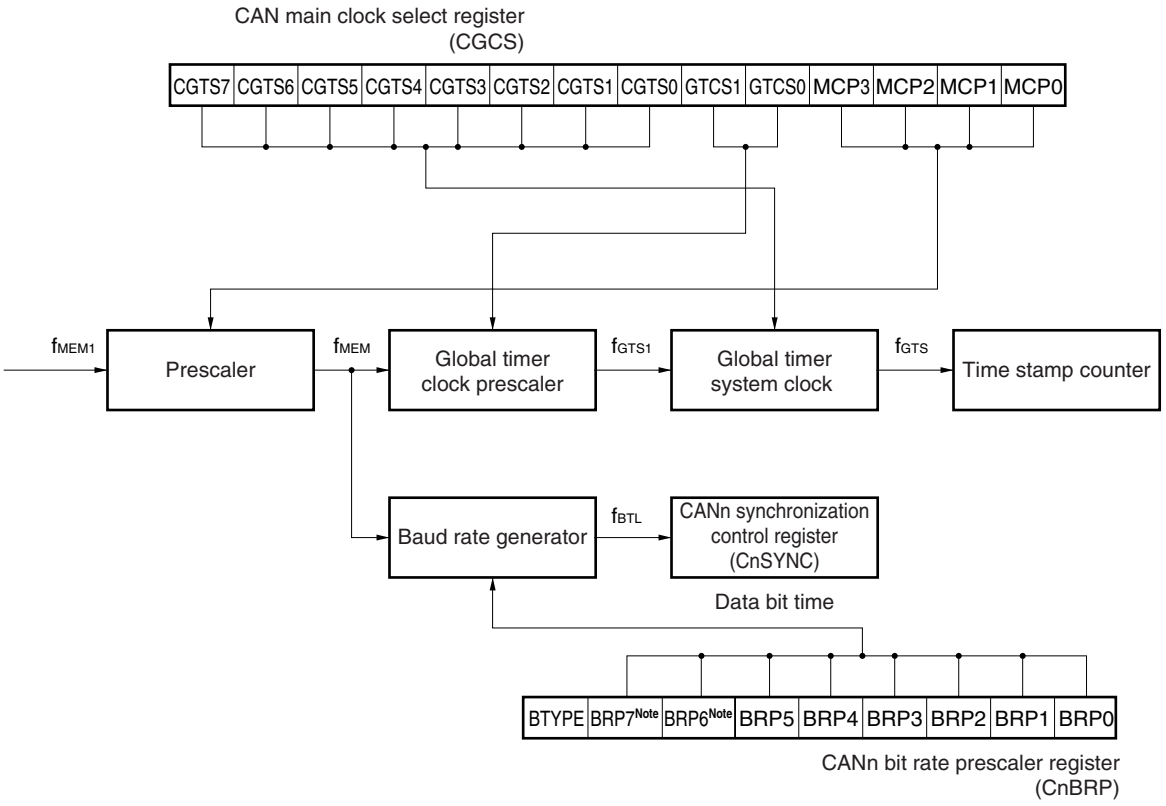
GTCS1	GTCS0	Global timer clock selection (f_{GTS1})
0	0	$f_{MEM}/2$
0	1	$f_{MEM}/4$
1	0	$f_{MEM}/8$
1	1	$f_{MEM}/16$

n	MCP3	MCP2	MCP1	MCP0	Selection of clock to memory access controller (f_{MEM})
0	0	0	0	0	f_{MEM1}
1	0	0	0	1	$f_{MEM1}/2$
2	0	0	1	0	$f_{MEM1}/3$
			:		$f_{MEM1}/(n+1)$
14	1	1	1	0	$f_{MEM1}/15$
15	1	1	1	1	$f_{MEM1}/16$
★ Once the values of the MCP3 to MCP0 bits are set after reset is released, do not change these values.					

★ **Notes 1.** When writing to this bit, always set it to 0.

2. See **19.5.16 CAN time stamp count register (CGTSC)**.

★ **Figure 19-2. FCAN Clocks**



Note Only when the TLM bit of the CANn bit rate prescaler register (CnBRP) is 1.

Remarks 1. $f_{MEM1} = f_{XX}$ = Clock supply to CAN

2. n = 1, 2

19.5.16 CAN time stamp count register (CGTSC)

The CGTSC register indicates the contents of the time stamp counter.

This register can be read at any time.

This register can be written to only when clearing bits. The clear function writes 0 to all bits in the CGTSC register.

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

After reset: 0000H R Address: xx3FFC18H

	15	14	13	12	11	10	9	8
CGTSC	TSC15	TSC14	TSC13	TSC12	TSC11	TSC10	TSC9	TSC8
	7	6	5	4	3	2	1	0
	TSC7	TSC6	TSC5	TSC4	TSC3	TSC2	TSC1	TSC0

19.5.17 CAN message search start/result register (CGMSS/CGMSR)

The CGMSS/CGMSR register indicates the message search start/result status. Messages in the message buffer that match the specified search criteria can be searched quickly.

These registers can be read/written in 16-bit units.

★ **Cautions 1. Execute the search by writing only once.**

★ **2. Always set the SMNO2 bit of the CGMSS register to 0.**
If 1 is set, operation is not guaranteed.

(1/2)

After reset: 0000H R/W Address: xx3FFC1AH

(a) Read

CGMSR	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	MM	AM
	7	6	5	4	3	2	1	0
	0	0	0	MFND4	MFND3	MFND2	MFND1	MFND0

(b) Write

★

CGMSS	15	14	13	12	11	10	9	8
	CIDE	0	CTRQ	CMSK	CDN	SMNO2	SMNO1	SMNO0
	7	6	5	4	3	2	1	0
	0	0	0	STRT4	STRT3	STRT2	STRT1	STRT0

(a) Read

MM	Confirmation of multiple hits from message search
0	No messages or only one message meets the search criteria
1	Several messages meet the search criteria
If several message buffers that meet the search criteria are detected, the MM bit is set.	

AM	Confirmation of hits from message search
0	No messages meet the search criteria
1	At least one message meets the search criteria

MFND4 to MFND0	Searched message number
This indicates the number (0 to 31) of the searched message. • When multiple message buffer numbers match as a result of a search (MM = 1), the return value of bits MFND4 to MFND0 is the lowest message buffer number. When no message buffer numbers match (AM = 0), the return value of bits MFND4 to MFND0 is “message buffer number −1”.	

(b) Write

CIDE	Message identifier (ID) format flag check
0	Message identifier format flag not checked
1	Message with standard format identifier checked

(2/2)

(b) Write

CTRQ	Transmit request and message ready flag check
0	Transmit request and message ready flags not checked
1	Transmit request and message ready flags checked

CMSK	Message check
0	Checked regardless of mask setting
1	Only unmasked messages checked

CDN	Status check of M_STATn register's DN flag (n = 00 to 31)
0	Status of M_STATn register's DN flag not checked
1	Status of M_STATn register's DN flag checked

★

SMNO2	SMNO1	SMNO0	Search module setting
0	0	0	No search module setting
0	0	1	CAN module 1 is set as the searched target
0	1	0	CAN module 2 is set as the searched target
Other than above			Setting prohibited

STRTn	Message search start position (n = 0 to 4)
0 to 31	Message search start position (message number)

- Search starts from the message number defined by bits STRT4 to STRT0. Search continues until it reaches the message buffer having the highest number among the usable message buffers. If the search results include several message buffer numbers among the matching messages, the message buffer with the lowest message buffer number is selected. To fetch the next message buffer number without changing the search criteria, "(MFND4 to MFND0) + 1" must be set as the values of bits STRT4 to STRT0.

19.5.18 CANn address mask a registers L and H (CnMASKLa and CnMASKHa)

The CnMASKLa and CnMASKHa registers are used to extend the number of receivable messages by masking part of the message's identifier (ID) and then ignoring the masked parts (a = 0 to 3, n = 1, 2).

These registers can be read/written in 16-bit units.

The C2MASKLa and C2MASKHa registers are valid only in models μPD703089Y and 70F3089Y.

After reset: Undefined R/W Address: See **Table 19-11**

	15	14	13	12	11	10	9	8
CnMASKHa	CMIDE	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
(a = 0 to 3, n = 1, 2)	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16
	15	14	13	12	11	10	9	8
CnMASKLa	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
(a = 0 to 3, n = 1, 2)	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

CMIDE	Mask setting for identifier (ID) format
0	ID format (standard or extended) checked
1	ID format (standard or extended) not checked
When the CMIDE bit is set (1), the higher 11 bits of ID are compared. The receive message and ID format stored in a message buffer are not compared.	

★

CMID0 to CMID28	Mask setting for identifier (ID) bits
0	ID bit in message buffer linked to bits CMID28 to CMID0 compared with received ID bit.
1	ID bit in message buffer linked to bits CMID28 to CMID0 not compared with received ID bit (i.e., masked).
A mask is always defined by an ID length of 29 bits. When a mask is assigned to the standard ID, always set the CMID17 to CMID0 bits to 1. The received ID is masked only by CMID28 to CMID18. The same mask can be used for standard and extended IDs.	

★

Table 19-11. Addresses of CnMASKLa and CnMASKHa (a = 0 to 3, n = 1, 2)

Register Name	Address	Register Name	Address
C1MASKL0	xx3FFC40H	C2MASKL0	xx3FFC80H
C1MASKH0	xx3FFC42H	C2MASKH0	xx3FFC82H
C1MASKL1	xx3FFC44H	C2MASKL1	xx3FFC84H
C1MASKH1	xx3FFC46H	C2MASKH1	xx3FFC86H
C1MASKL2	xx3FFC48H	C2MASKL2	xx3FFC88H
C1MASKH2	xx3FFC4AH	C2MASKH2	xx3FFC8AH
C1MASKL3	xx3FFC4CH	C2MASKL3	xx3FFC8CH
C1MASKH3	xx3FFC4EH	C2MASKH3	xx3FFC8EH

19.5.19 CANn control register (CnCTRL)

The CnCTRL register is used to control the operation of the CAN module.

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

The C2CTRL register is valid only in models μPD703089Y and 70F3089Y.

Cautions 1. Both bitwise writing and direct writing to the CnCTRL register are prohibited. Attempts to write directly to this register may result in operation faults, so be sure to follow the sequence described in 19.6 Cautions Regarding Bit Set/Clear Function.

2. When writing to the CnCTRL register, set or clear bits according to the register configuration shown in part (b) Write of the following figure.

3. When canceling CAN stop mode, CAN sleep mode must be cancelled at the same time.

After reset: 0101H R/W Addresses: C1CTRL: xx3FFC50H
C2CTRL: xx3FFC90H

(a) Read

CnCTRL (n = 1, 2)	15	14	13	12	11	10	9	8
	TECS1	TECS0	RECS1	RECS0	BOFF	TSTAT	RSTAT	ISTAT
	7	6	5	4	3	2	1	0
	0	DLEVR	DLEVT	OVM	TMR	STOP	SLEEP	INIT

(b) Write

CnCTRL (n = 1, 2)	15	14	13	12	11	10	9	8
	0	set DLEVR	set DLEVT	set OVM	set TMR	set STOP	set SLEEP	set INIT
	7	6	5	4	3	2	1	0
	0	clear DLEVR	clear DLEVT	clear OVM	clear TMR	clear STOP	clear SLEEP	clear INIT

(a) Read

TECS1	TECS0	Status of transmit error counter
0	0	Transmit error counter value < 96
0	1	Transmit error counter value = 96 to 127 (warning level)
1	0	Not used
1	1	Transmit error counter value ≥ 128 (error passive)

RECS1	RECS0	Status of receive error counter
0	0	Receive error counter value < 96
0	1	Receive error counter value = 96 to 127 (warning level)
1	0	Not used
1	1	Receive error counter value ≥ 128 (error passive)

BOFF	Bus off status flag
0	Transmit error counter value < 256 (not bus off status)
1	Transmit error counter value ≥ 256 (bus off status)

(2/4)

(a) Read

TSTAT	Transmit status flag
0	Transmit stop status
1	Transmit operating status

RSTAT	Receive status flag
0	Receive stop status
1	Receive operating status

ISTAT	Initialization status flag
0	Normal operating status
1	FCAN is stopped and initialized

- The ISTAT bit is set (1) when the CAN protocol layer acknowledges the setting of the INIT bit. The ISTAT bit is automatically cleared (0) after the INIT bit is cleared (0).
- "Recessive" is output via the CANTXn pin in initialization mode.
- The CnSYNC and CnBRP registers can be written only in initialization mode.
- When shifting from the initialization status to the normal operating status, the error counter (see **19.5.22 CANn error count register (CnERC)**) is cleared (0) and the error status (TECS1, TECS0, RECS1, or RECS0 bit) is reset.

DLEVR	Dominant level control bit for receive pin
0	A low level to a receive pin is acknowledged as dominant
1	A high level to a receive pin is acknowledged as dominant

DLEVT	Dominant level control bit for transmit pin
0	A low level is transmitted from transmit pin as dominant
1	A high level is transmitted from transmit pin as dominant

OVM	Overwrite mode control bit
0	New messages stored in message buffer in which DN bit of M_STATa register is set (a = 00 to 31)
1	New messages in message buffer in which DN bit is set (a = 00 to 31) discarded

TMR	Time stamp control bit for reception
0	When the SOF is detected on the CAN bus, the value of the time stamp counter is captured.
1	When the EOF is detected on the CAN bus (a valid message is confirmed), the value of the time stamp counter is captured.

(3/4)

(a) Read

STOP	CAN stop mode control bit
0	Normal CAN sleep mode
1	CAN stop mode (change in CAN bus does not cause wakeup)
• CAN stop mode can be selected only when the CAN module has been set to CAN sleep mode, i.e., when the SLEEP bit has been set (1). • CAN stop mode can be canceled only by the CPU by clearing the STOP bit (0).	

SLEEP	CAN sleep mode control bit
0	Normal operating mode
1	CAN sleep mode (change in CAN bus causes wakeup)

★ • CAN sleep mode can be set only when the CAN bus is in the idle state.
• CAN sleep mode is canceled under the following conditions.
  • When the CPU has cleared the SLEEP bit (0)
  • When the CAN bus changes (this occurs only when CAN stop mode has not been set)
• The WAKE bit[Note] is set (1) only when CAN sleep mode is cancelled by the change of the CAN bus, and an error interrupt occurs.
★ • To check the settings of the SLEEP bit, read the CnCTRL register.	

INIT	Initialization request bit
0	Normal operation mode
1	Initialization mode
• Be sure to confirm that the CAN module has entered the initialization mode using the ISTAT bit (ISTAT bit = 1) after setting the INIT bit (1). • If the INIT bit is set (1) when the CAN module is in the bus off status (BOFF bit = 1), the CAN module enters initialization mode (ISTAT bit = 1) immediately.	

Note See **19.5.20 CANn definition register (CnDEF)**.

(4/4)

(b) Write

set DLEVR	clear DLEVR	DLEVR bit setting
0	1	DLEVR bit cleared (0)
1	0	DLEVR bit set (1)
Other than above		DLEVR bit not changed

set DLEVT	clear DLEVT	DLEVT bit setting
0	1	DLEVT bit cleared (0)
1	0	DLEVT bit set (1)
Other than above		DLEVT bit not changed

set OVM	clear OVM	OVM bit setting
0	1	OVM bit cleared (0)
1	0	OVM bit set (1)
Other than above		OVM bit not changed

set TMR	clear TMR	TMR bit setting
0	1	TMR bit cleared (0)
1	0	TMR bit set (1)
Other than above		TMR bit not changed

set STOP	clear STOP	STOP bit setting
0	1	STOP bit cleared (0)
1	0	STOP bit set (1)
Other than above		STOP bit not changed

set SLEEP	clear SLEEP	SLEEP bit setting
0	1	SLEEP bit cleared (0)
1	0	SLEEP bit set (1)
Other than above		SLEEP bit not changed

set INIT	clear INIT	INIT bit setting
0	1	INIT bit cleared (0)
1	0	INIT bit set (1)
Other than above		INIT bit not changed

19.5.20 CANn definition register (CnDEF)

The CnDEF register is used to define the operation of the CAN module.

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

The C2DEF register is valid only in models μPD703089Y and 70F3089Y.

Cautions 1. Both bitwise writing and direct writing to the CnDEF register are prohibited. Attempts to write directly to this register may result in operation faults, so be sure to follow the sequence described in 19.6 Cautions Regarding Bit Set/Clear Function.

2. When writing to the CnDEF register, set or clear bits according to the register configuration shown in part (b) Write of the following figure.

(1/4)

After reset: 0000H R/W Addresses: C1DEF: xx3FFC52H
C2DEF: xx3FFC92H

(a) Read

CnDEF (n = 1, 2)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	DGM	MOM	SSHT	PBB	BERR	VALID	WAKE	OVR

(b) Write

CnDEF (n = 1, 2)	15	14	13	12	11	10	9	8
	set DGM	set MOM	set SSHT	set PBB	0	0	0	0
	7	6	5	4	3	2	1	0
	clear DGM	clear MOM	clear SSHT	clear PBB	clear BERR	clear VALID	clear WAKE	clear OVR

(a) Read

DGM	Specification of diagnostic processing mode
0	Valid messages are stored in the message buffer used for diagnostic processing mode[Note] (only when receiving)
1	Valid messages are stored in the same way as in normal operating mode (only when receiving)
• The diagnostic processing mode (MOM bit = 1) is used for CAN baud rate detection and for diagnostic purposes. When this mode is set, the following operations are performed. • When the VALID bit = 1, it indicates that the current receive operation is valid. • Setting the DGM bit confirms whether valid data has been stored in the message buffer used for diagnostic processing mode, or stored in the same way as in normal operating mode.	

Note Bits 5 to 3 (MT2 to MT0) of CAN message configuration register a (M_CONFa) are set as “111” (a = 00 to 31).

(2/4)

(a) Read

MOM	Specification of CAN module's operating mode
0	Normal operating mode
1	Diagnostic processing mode

- When in diagnostic processing mode (MOM bit = 1), the CnBRP register can be accessed only when the CAN module has been set to initialization mode (i.e., when the CnCTRL register's ISTAT bit = INIT bit = 1).
  When the CAN module is operating (i.e., when the CnCTRL register's ISTAT bit = 0) the CnBRP register cannot be used, and the CANn bus diagnostic information register (CnDINF) register can be used instead.
- The CAN protocol layer does not sent ACK, error frame, or transmit messages, nor does it operate an error counter.
  The internal transmit output is fed back to the internal input due to auto baud rate detection.

SSHT	Specification of single shot mode
0	Normal operating mode
1	Single shot mode

- In single shot mode, the CAN module can transmit a message only once. The M_STATa (a = 00 to 31) register's TRQ bit is then cleared (0) regardless of whether or not there are any pending normal transmit operations.
  ★ Also, if a bus error has occurred due to a transmission, it is processed as a transmission error.
- In single shot mode, even if the CAN lost in the arbitration phase, it is handled as a completed message transmission.
  In this mode, the BERR bit is set (1) but the error counter (see **19.5.22 CANn error count register (CnERC)**) value does not change since there are no CAN bus errors.
- ★ In single shot mode, even when transmission is stopped due to error detection or a loss in the arbitration phase, the transmission completion interrupt occurs.
- During the time when the CAN module is active, normal operating mode and single shot mode can be switched without causing any errors on the CAN bus.

PBB	Specification of priority control for transmission
0	Identifier (ID) based priority control
1	Message number based priority control

- Ordinarily, priority for transmission is defined based on message IDs, but when the PBB bit has been set (1) priority becomes based instead on the position of messages, so that messages with lower message numbers have higher priority.

BERR	CAN bus error status
0	CAN bus error was not detected
1	CAN bus error was detected at least once after bit was cleared

VALID	Valid message detection status
0	Valid message was not detected
1	Valid message was detected at least once after bit was cleared

(3/4)

(a) Read

WAKE	CAN sleep mode cancellation status
0	Normal operation
1	Cancel CAN sleep mode

★ • The WAKE bit is set (1) only when the CAN sleep mode is released due to a change of CAN bus and an error interrupt occurs.

★ • While the WAKE bit is set (1), the error interrupt signal holds the active status. Therefore, always clear (0) the WAKE bit after recognition.

OVR	Overrun error status
0	Normal operation
1	Overwrite occurred during RAM access

• When an overrun error has occurred, the OVR bit is set (1) and an error interrupt occurs at the same time.
The source of the overrun error may be that the RAM access clock is slower than the selected CAN baud rate.

(b) Write

set DGM	clear DGM	DGM bit setting
0	1	DGM bit cleared (0)
1	0	DGM bit set (1)
Other than above		DGM bit not changed

set MOM	clear MOM	MOM bit setting
0	1	MOM bit cleared (0)
1	0	MOM bit set (1)
Other than above		MOM bit not changed

set SSHT	clear SSHT	SSHT bit setting
0	1	SSHT bit cleared (0)
1	0	SSHT bit set (1)
Other than above		SSHT bit not changed

set PBB	clear PBB	PBB bit setting
0	1	PBB bit cleared (0)
1	0	PBB bit set (1)
Other than above		PBB bit not changed

clear BERR	BERR bit setting
1	BERR bit cleared (0)
0	BERR bit not changed

clear VALID	VALID bit setting
1	VALID bit cleared (0)
0	VALID bit not changed

clear WAKE	WAKE bit setting
1	WAKE bit cleared (0)
0	WAKE bit not changed

clear OVR	OVR bit setting
1	OVR bit cleared (0)
0	OVR bit not changed

19.5.21 CANn information register (CnLAST)

The CnLAST register indicates the CANn module's error information and the number of the message buffer received last.

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

The C2LAST register is valid only in models μPD703089Y and 70F3089Y.

After reset: 00FFH R Addresses: C1LAST: xx3FFC54H
C2LAST: xx3FFC94H

	15	14	13	12	11	10	9	8
CnLAST	0	0	0	0	LERR3	LERR2	LERR1	LERR0
(n = 1, 2)	7	6	5	4	3	2	1	0
	LREC7	LREC6	LREC5	LREC4	LREC3	LREC2	LREC1	LREC0

LERR3	LERR2	LERR1	LERR0	Last error information
0	0	0	0	Error not detected
0	0	0	1	Bit error
0	0	1	0	Stuff error
0	0	1	1	CRC error
0	1	0	0	Form error
0	1	0	1	ACK error
0	1	1	0	Arbitration lost (only during single shot mode) (CnDEF: SSHT = 1)
0	1	1	1	CAN overwrite error
1	0	0	0	Wakeup from CAN bus
Other than above				Setting prohibited

- Since bits LERR3 to LERR0 cannot be cleared, the current status is retained until the next error occurs.

LREC7 to LREC0	Number of last receive message
0 to 31	Message number of message last received
32 to 255	Not used

19.5.22 CANn error count register (CnERC)

The CnERC register indicates the count values of the transmission/reception error counters.

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

The C2ERC register is valid only in models μPD703089Y and 70F3089Y.

After reset: 0000H R Addresses: C1ERC: xx3FFC56H
C2ERC: xx3FFC96H

	15	14	13	12	11	10	9	8
CnERC	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
(n = 1, 2)	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REC7 to REC0	Reception error counter
0 to 255	Number of reception error counts
• This reflects the current status of the reception error counter. • The count value is defined by the CAN protocol.	

TEC7 to TEC0	Transmission error counter
0 to 255	Number of transmission error counts
• This reflects the current status of the transmission error counter. • The number of counts is defined by the CAN protocol.	

19.5.23 CANn interrupt enable register (CnIE)

The CnIE register is used to enable/disable the CAN module's interrupts.

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

The C2IE register is valid only in models μPD703089Y and 70F3089Y.

Cautions 1. Both bitwise writing and direct writing to the CnIE register are prohibited. Attempts to write directly to this register may result in operation faults, so be sure to follow the sequence described in 19.6 Cautions Regarding Bit Set/Clear Function.

2. When writing to the CnIE register, set or clear bits according to the register configuration shown in part (b) Write of the following figure.

(1/3)

After reset: 0000H R/W Addresses: C1IE: xx3FFC58H
C2IE: xx3FFC98H

(a) Read

CnIE (n = 1, 2)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	E_INT6	E_INT5	E_INT4	E_INT3	E_INT2	E_INT1	E_INT0

(b) Write

CnIE (n = 1, 2)	15	14	13	12	11	10	9	8
	0	set E_INT6	set E_INT5	set E_INT4	set E_INT3	set E_INT2	set E_INT1	set E_INT0
	7	6	5	4	3	2	1	0
	0	clear E_INT6	clear E_INT5	clear E_INT4	clear E_INT3	clear E_INT2	clear E_INT1	clear E_INT0

(a) Read

E_INT6	CAN module error interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled

E_INT5	CAN bus error interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled

E_INT4	Wake up from CAN sleep mode interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled

E_INT3	CAN receive error passive interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled

(2/3)

(a) Read

E_INT2	CAN transmit error passive or bus off interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled

E_INT1	CAN receive completion interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled
• When IE bit of the M_CTRLn register is 1, a reception completion interrupt occurs regardless of the setting of the E_INT1 bit if the transmit message buffer receives a remote frame while the auto response function is not set (RMDE0 bit of the M_CTRLn register = 0).	

★

E_INT0	CAN transmit completion interrupt enable flag
0	Interrupt disabled
1	Interrupt enabled

(3/3)

(b) Write

set E_INT6	clear E_INT6	E_INT6 bit setting
0	1	E_INT6 interrupt cleared (0)
1	0	E_INT6 interrupt set (1)
Other than above		E_INT6 interrupt not changed

set E_INT5	clear E_INT5	E_INT5 bit setting
0	1	E_INT5 interrupt cleared (0)
1	0	E_INT5 interrupt set (1)
Other than above		E_INT5 interrupt not changed

set E_INT4	clear E_INT4	E_INT4 bit setting
0	1	E_INT4 interrupt cleared (0)
1	0	E_INT4 interrupt set (1)
Other than above		E_INT4 interrupt not changed

set E_INT3	clear E_INT3	E_INT3 bit setting
0	1	E_INT3 interrupt cleared (0)
1	0	E_INT3 interrupt set (1)
Other than above		E_INT3 interrupt not changed

set E_INT2	clear E_INT2	E_INT2 bit setting
0	1	E_INT2 interrupt cleared (0)
1	0	E_INT2 interrupt set (1)
Other than above		E_INT2 interrupt not changed

set E_INT1	clear E_INT1	E_INT1 bit setting
0	1	E_INT1 interrupt cleared (0)
1	0	E_INT1 interrupt set (1)
Other than above		E_INT1 interrupt not changed

set E_INT0	clear E_INT0	E_INT0 bit setting
0	1	E_INT0 interrupt cleared (0)
1	0	E_INT0 interrupt set (1)
Other than above		E_INT0 interrupt not changed

19.5.24 CANn bus active register (CnBA)

The CnBA register indicates frame information output via the CAN bus.

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

The C2BA register is valid only in models μPD703089Y and 70F3089Y.

After reset: 00FFH R Addresses: C1BA: xx3FFC5AH
C2BA: xx3FFC9AH

	15	14	13	12	11	10	9	8
CnBA	0	0	0	CACT4	CACT3	CACT2	CACT1	CACT0
(n = 1, 2)	7	6	5	4	3	2	1	0
	TMNO7	TMNO6	TMNO5	TMNO4	TMNO3	TMNO2	TMNO1	TMNO0

CACT4	CACT3	CACT2	CACT1	CACT0	CAN module status
0	0	0	0	0	Reset state
0	0	0	0	1	Bus idle wait
0	0	0	1	0	Bus idle state
0	0	0	1	1	Start of frame
0	0	1	0	0	Standard identifier area
0	0	1	0	1	Data length code area
0	0	1	1	0	Data field area
0	0	1	1	1	CRC field area
0	1	0	0	0	CRC delimiter
0	1	0	0	1	ACK slot
0	1	0	1	0	ACK delimiter
0	1	0	1	1	End of frame area
0	1	1	0	0	Intermission state
0	1	1	0	1	Suspend transmission
0	1	1	1	0	Error frame
0	1	1	1	1	Error delimiter wait
1	0	0	0	0	Error delimiter
1	0	0	0	1	Bus off error
1	0	0	1	0	Extended identifier area
Other than above					Setting prohibited

TMNO7 to TMNO0	Transmission message counter
0 to 31	Message number of message awaiting transmission or being transmitted
32 to 254	Not used
255	No messages awaiting transmission or being transmitted

19.5.25 CANn bit rate prescaler register (CnBRP)

The CnBRP register is used to set the transmission baud rate for the CAN module.

Use the CnBRP register to select the CAN protocol layer basic system clock (f_{BTL}). The baud rate is determined by the value set to the CnSYNC register.

While in normal operating mode (CnDEF register's MOM bit = 0), writing to the CnBRP register is enabled only when the initialization mode has been set (CnCTRL register's INIT bit = 1).

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

The C2BRP register is valid only in models μPD703089Y and 70F3089Y.

Caution While in diagnostic processing mode (CnDEF register's MOM bit = 1), the CnBRP register can be accessed only when the initialization mode has been set.

(1/2)

After reset: 0000H R/W Addresses: C1BRP: xx3FFC5CH
C2BRP: xx3FFC9CH

(a) When TLM = 0

	15	14	13	12	11	10	9	8
CnBRP	TLM	0	0	0	0	0	0	0
(n = 1, 2)	7	6	5	4	3	2	1	0
	0	BTYPE	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0

(b) When TLM = 1

	15	14	13	12	11	10	9	8
CnBRP	TLM	0	0	0	0	0	0	BTYPE
(n = 1, 2)	7	6	5	4	3	2	1	0
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0

(a) When TLM = 0

TLM	Transfer layer mode specification
0	6-bit prescaler mode

BTYPE	CAN bus type specification
0	Low speed (≤ 125 Kbps)
1	High speed (> 125 Kbps)

a	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	CAN protocol layer basic system clock (f_{BTL})
0	0	0	0	0	0	0	$f_{MEM}/2$
1	0	0	0	0	0	1	$f_{MEM}/4$
2	0	0	0	0	1	0	$f_{MEM}/6$
3	0	0	0	0	1	1	$f_{MEM}/8$
:							$f_{MEM}/\{(a + 1) \times 2\}$
60	1	1	1	1	0	0	$f_{MEM}/122$
61	1	1	1	1	0	1	$f_{MEM}/124$
62	1	1	1	1	1	0	$f_{MEM}/126$
63	1	1	1	1	1	1	$f_{MEM}/128$

Remark $f_{BTL} = f_{MEM}/\{(a + 1) \times 2\}$: CAN protocol layer basic system clock
a = 0 to 63 (set by bits BRP5 to BRP0)
f_{MEM} = CAN base clock

(2/2)

(b) When TLM = 1

TLM	Transfer layer mode specification
1	8-bit prescaler mode

BTYPE	CAN bus type specification
0	Low speed (≤ 125 Kbps)
1	High speed (> 125 Kbps)

a	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	CAN protocol layer basic system clock (f_{BTL})
0	0	0	0	0	0	0	0	0	Setting prohibited
1	0	0	0	0	0	0	0	1	$f_{MEM}/2$
2	0	0	0	0	0	0	1	0	$f_{MEM}/3$
3	0	0	0	0	0	0	1	1	$f_{MEM}/4$
:									$f_{MEM}/(a + 1)$
252	1	1	1	1	1	1	0	0	$f_{MEM}/253$
253	1	1	1	1	1	1	0	1	$f_{MEM}/254$
254	1	1	1	1	1	1	1	0	$f_{MEM}/255$
255	1	1	1	1	1	1	1	1	$f_{MEM}/256$

Remark $f_{BTL} = f_{MEM}/(a + 1)$: CAN protocol layer basic system clock
a = 0 to 255 (set by bits BRP7 to BRP0)
f_{MEM} = CAN base clock

19.5.26 CANn bus diagnostic information register (CnDINF)

The CnDINF register indicates all the CAN bus bits, including the stuff bits and delimiters. This information is used only for diagnostic purposes.

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

The C2DINF register is valid only in models μPD703089Y and 70F3089Y.

Cautions 1. The CnDINF register can be accessed only while in diagnostic processing mode (CnDEF register's MOM bit = 1) and in normal operating mode (CnCTRL register's INIT bit = 0).

2. Storage of the last 8 bits is automatically stopped if an error or a valid message (ACK delimiter) is detected on the CAN bus. Storage is automatically reset each time when SOF is detected on the CAN bus.

After reset: 0000H R Addresses: C1DINF: xx3FFC5CH
C2DINF: xx3FFC9CH

	15	14	13	12	11	10	9	8
CnDINF	DINF15	DINF14	DINF13	DINF12	DINF11	DINF10	DINF9	DINF8
(n = 1, 2)	7	6	5	4	3	2	1	0
	DINF7	DINF6	DINF5	DINF4	DINF3	DINF2	DINF1	DINF0

DINFa	CANn bus diagnostic information
DINF15 to DINF8	No. of bits starting from SOF
DINF7 to DINF0	Information from last 8 bits

19.5.27 CANn synchronization control register (CnSYNC)

The CnSYNC register controls the data bit time for transmission speed.

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

The C2SYNC register is valid only in models μPD703089Y and 70F3089Y.

Cautions 1. The CPU is able to read the CnSYNC register at any time.

2. Writing to the CnSYNC register is enabled in initialization mode (when CnCTRL register's INIT bit = 1).

3. The limit values of CAN protocol when setting the SPTRa bit and DBTRa bit are as follows (a = 0 to 4).

- **5 × BTL ≤ SPT (sample point) ≤ 17 × BTL [4 ≤ set values of SPTR4 to SPTR0 ≤ 16]**
- **8 × BTL ≤ DBT (data bit time) ≤ 25 × BTL [7 ≤ set values of DBTR4 to DBTR0 ≤ 24]**
- ★ **SJW (synchronization jump width) ≤ DBT − SPT**
- **2 ≤ (DBT − SPT) ≤ 8**

Remark BTL = 1/f_{BTL} (f_{BTL}: CAN protocol layer basic system clock)

(1/2)

After reset: 0218H R/W Addresses: C1SYNC: xx3FFC5EH
C2SYNC: xx3FFC9EH

CnSYNC (n = 1, 2)

15	14	13	12	11	10	9	8
0	0	0	SAMP	SJWR1	SJWR0	SPTR4	SPTR3
7	6	5	4	3	2	1	0
SPTR2	SPTR1	SPTR0	DBTR4	DBTR3	DBTR2	DBTR1	DBTR0

SAMP	Bit sampling specification
0	Sample data received at the sample point once
1	Sample received data three times and majority value used as sampled value

SJWR1	SJWR0	Synchronization jump width[Note]
0	0	BTL
0	1	BTL × 2
1	0	BTL × 3
1	1	BTL × 4

Note As stipulated in the CAN protocol specification, Ver. 2.0 PartB active.

Remark BTL = 1/f_{BTL} (f_{BTL}: CAN protocol layer basic system clock)

(2/2)

SPTR4	SPTR3	SPTR2	SPTR1	SPTR0	Position of sampling point
0	0	0	1	0	BTL × 3[Note]
0	0	0	1	1	BTL × 4[Note]
0	0	1	0	0	BTL × 5
0	0	1	0	1	BTL × 6
0	0	1	1	0	BTL × 7
0	0	1	1	1	BTL × 8
0	1	0	0	0	BTL × 9
0	1	0	0	1	BTL × 10
0	1	0	1	0	BTL × 11
0	1	0	1	1	BTL × 12
0	1	1	0	0	BTL × 13
0	1	1	0	1	BTL × 14
0	1	1	1	0	BTL × 15
0	1	1	1	1	BTL × 16
1	0	0	0	0	BTL × 17
Other than above					Setting prohibited
Sampling point within bit timing is selected.					

DBTR4	DBTR3	DBTR2	DBTR1	DBTR0	Data bit time
0	0	1	1	1	BTL × 8
0	1	0	0	0	BTL × 9
0	1	0	0	1	BTL × 10
0	1	0	1	0	BTL × 11
0	1	0	1	1	BTL × 12
0	1	1	0	0	BTL × 13
0	1	1	0	1	BTL × 14
0	1	1	1	0	BTL × 15
0	1	1	1	1	BTL × 16
1	0	0	0	0	BTL × 17
1	0	0	0	1	BTL × 18
1	0	0	1	0	BTL × 19
1	0	0	1	1	BTL × 20
1	0	1	0	0	BTL × 21
1	0	1	0	1	BTL × 22
1	0	1	1	0	BTL × 23
1	0	1	1	1	BTL × 24
1	1	0	0	0	BTL × 25
Other than above					Setting prohibited
1-bit data length is set for CAN bus					

Note This setting is reserved for setting sample point extension and is not compliant with the CAN protocol specifications.

Remark BTL = 1/f_{BTL} (f_{BTL}: CAN protocol layer basic system clock)

19.6 Cautions Regarding Bit Set/Clear Function

The FCAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written to directly, so do not directly write (via bit manipulation, read/modify/write, or direct writing of target values) values to them.

- CAN global status register (CGST)
- CAN global interrupt enable register (CGIE)
- CANn control register (CnCTRL)
- CANn definition register (CnDEF)
- CANn interrupt enable register (CnIE)

Remark n = 1, 2

All 16 bits in the above registers can be read via the usual method. Use the procedure described in Figure 19-3 below to set or clear the lower 8 bits in these registers.

Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (see **Figure 19-4**). Figure 19-3 shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 19-3. Example of Bit Setting/Clearing Operations

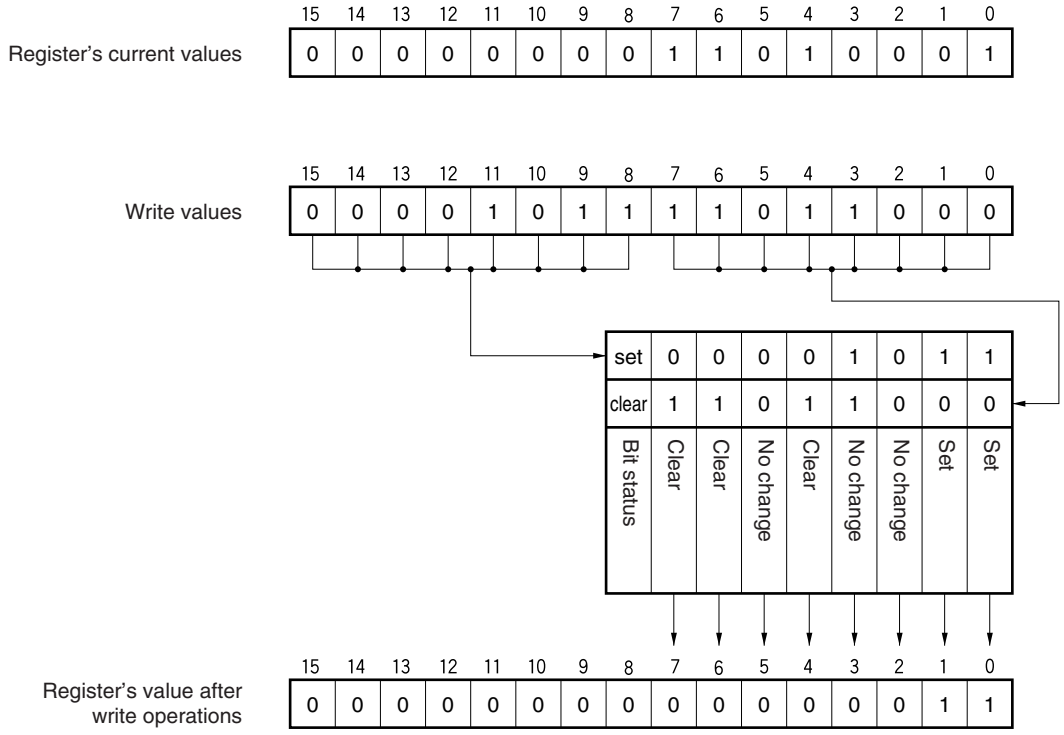


Figure 19-4. 16-Bit Data During Write Operation

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
set 7	set 6	set 5	set 4	set 3	set 2	set 1	set 0	clear 7	clear 6	clear 5	clear 4	clear 3	clear 2	clear 1	clear 0

set n	clear n	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

Remark n = 0 to 7

19.7 Time Stamp Function

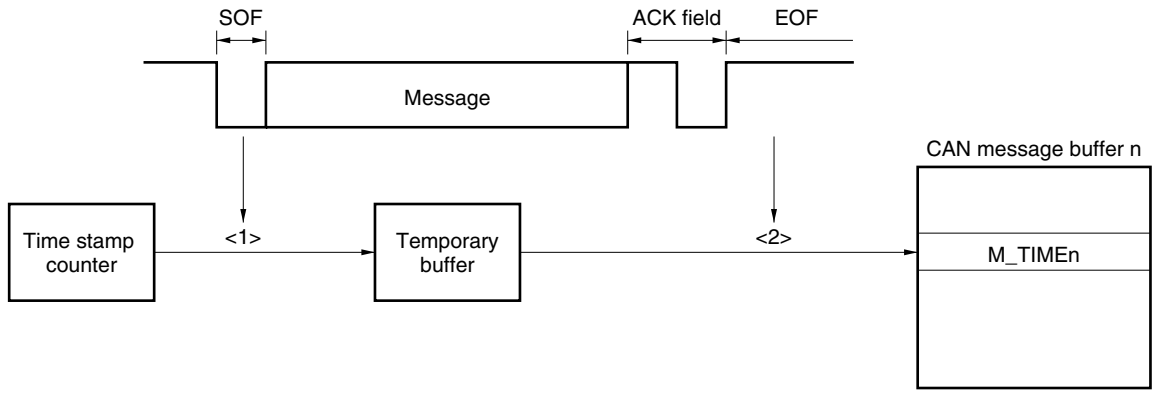
★ **Caution In the V850/SC3, the time stamp function by SOF detection during message transmission/reception cannot be used.**
Only the time stamp function by EOF detection during message reception can be used for the V850/SC3. However, only the value captured by the M_TIME register is valid when the TSM bit of the CGST register is set to 1 and the TMR bit of the CnCTRL register is set to 1.

The FCAN controller supports a time stamp function. This function is needed to build a global time system.

The time stamp function is implemented using a 16-bit free-running time stamp counter.

Two types of time stamp function can be selected for message reception in the FCAN controller. Use bit 3 (TMR) of the CANx control register (CxCTRL) to set the desired time stamp function (x = 1, 2). When the TMRT bit is 0, the time stamp counter value is captured after the SOF is sent via the CAN bus (see **Figure 19-5**) and when the TMR bit is 1, the time stamp counter value is captured after the EOF is sent via the CAN bus (a valid message is confirmed) (see **Figure 19-6**).

Figure 19-5. Time Stamp Function Setting for Message Reception (When CxCTRL Register's TMR Bit = 0)



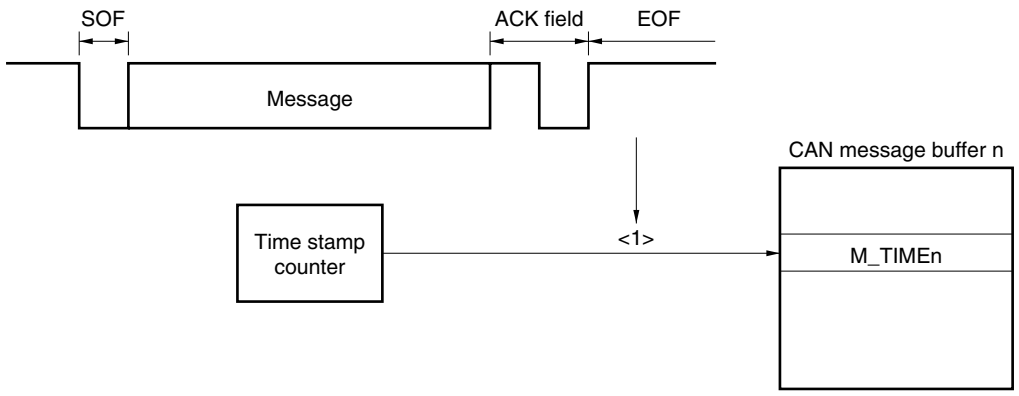
<Explanation>

<1> The time stamp counter value is captured when the SOF is sent via the CAN bus.

<2> A message is stored in CAN message buffer n and the value in the temporary buffer is copied to the M_TIMEn register in CAN message buffer n when the EOF is sent via the CAN bus.

Remark n = 00 to 31
x = 1, 2

Figure 19-6. Time Stamp Function Setting for Message Reception (When CxCTRL Register's TMR Bit = 1)



<Explanation>

<1> When the EOF is sent via the CAN bus (a valid message is acknowledged), the captured time stamp counter value is copied to the M_TIMEn register in CAN message buffer n when a message is stored in CAN message buffer n.

Remark n = 00 to 31
x = 1, 2

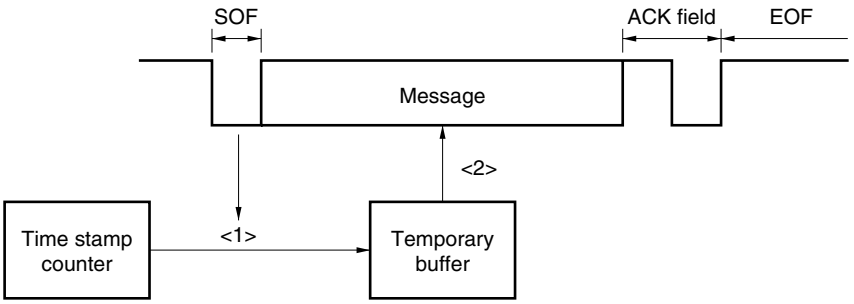
In a global time system, the time value must be captured using the SOF.

In addition, the ability to capture the counter value when data is stored in a message buffer is useful for evaluating the FCAN controller's performance.

The captured time stamp counter value is stored in CAN message buffer n, so CAN message buffer n has its own time stamp function (n = 00 to 31).

When the SOF is sent via the CAN bus while transmitting a message, there is an option to replace the last two bytes of the message with the time stamp counter value by setting bit 5 (ATS) of CAN message control register n (M_CTRLn). This function can be selected for CAN message buffer n on a buffer by buffer basis. Figure 19-7 shows the time stamp setting when the ATS bit = 1.

Figure 19-7. Time Stamp Function Setting for Message Transmission (When M_CTRL Register's ATS Bit = 1)



<Explanation>

<1> The time stamp counter value is captured to the temporary buffer when the SOF is detected on the CAN bus.

<2> The value of the temporary buffer is added to the last 2 bytes of the data length code[Note] specified by bits DLC3 to DLC0 of the M_DLCn register.

Note The ATS bit of the M_CTRLn register must be 1 and the data length must be more than 2 bytes to add the time stamp counter value to the transmit message.

Remark n = 00 to 31

Table 19-12. Example When Adding Captured Time Stamp Counter Value to Last 2 Bytes of Transmit Message

Data Field / DLC Bit Value[Note 1]	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
1	M_DATAn0 register value	–	–	–	–	–	–	–
2	**Note 2**	**Note 3**	–	–	–	–	–	–
3	M_DATAn0 register value	**Note 2**	**Note 3**	–	–	–	–	–
4	M_DATAn0 register value	M_DATAn1 register value	**Note 2**	**Note 3**	–	–	–	–
5	M_DATAn0 register value	M_DATAn1 register value	M_DATAn2 register value	**Note 2**	**Note 3**	–	–	–
6	M_DATAn0 register value	M_DATAn1 register value	M_DATAn2 register value	M_DATAn3 register value	**Note 2**	**Note 3**	–	–
7	M_DATAn0 register value	M_DATAn1 register value	M_DATAn2 register value	M_DATAn3 register value	M_DATAn4 register value	**Note 2**	**Note 3**	–
8	M_DATAn0 register value	M_DATAn1 register value	M_DATAn2 register value	M_DATAn3 register value	M_DATAn4 register value	M_DATAn5 register value	**Note 2**	**Note 3**
9 to 15	M_DATAn0 register value	M_DATAn1 register value	M_DATAn2 register value	M_DATAn3 register value	M_DATAn4 register value	M_DATAn5 register value	**Note 2**	**Note 3**

Notes 1. See **19.5.1 CAN message data length registers 00 to 31 (M_DLC00 to M_DLC31)**.

2. The lower 8 bits of the time stamp counter value when the SOF is detected on the CAN bus

3. The higher 8 bits of the time stamp counter value when the SOF is detected on the CAN bus

Remark n = 00 to 31

★ 19.8 Message Processing

A modular system is used for the FCAN controller. Consequently, messages can be placed at any location within the message area.

The messages can be linked to mask functions that are in turn linked to CAN modules.

19.8.1 Message transmission

The FCAN system is a multiplexed communication system. The priority of message transmission within this system is determined based on message identifiers (IDs).

To facilitate communication processing by application software when there are several messages awaiting transmission, the CAN module uses hardware to check the message IDs and automatically determine whether or not linked messages are prioritized.

This eliminates the need for software-based priority control.

In addition, the priority at transmission can be controlled by setting the PBB bit of the CnDEF register.

- When the PBB bit is set to 0 (see **Figure 19-8**)
  Transmission priority is controlled by the identifier (ID).
  The number[Note] of messages waiting to be transmitted in the message buffer that can be set simultaneously by application software is up to five messages per CAN module.

  Note The number of message buffers when the TRQ bit of the M_STATn register = 1.

- When the PBB bit is set to 1 (see **Figure 19-9**)
  Transmission priority is controlled by the message numbers.
  The number[Note] of messages waiting to be transmitted in the message buffer is not limited by the application software.

Figure 19-8. Message Processing Example (When PBB Bit = 0)

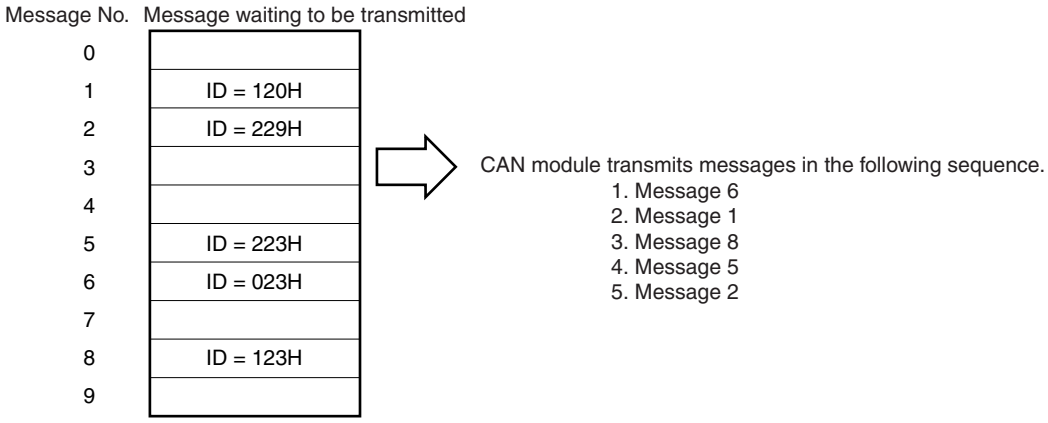
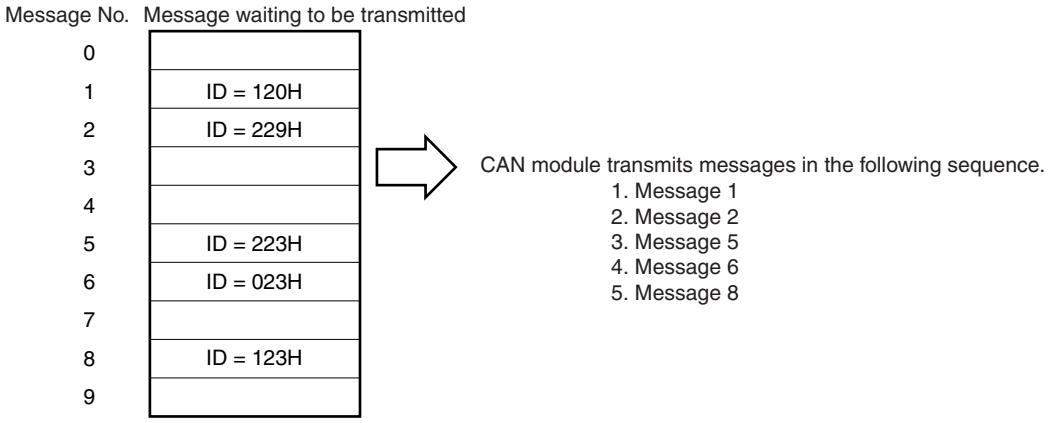


Figure 19-9. Message Processing Example (When PBB Bit = 1)



19.8.2 Message reception

When two or more message buffers of the CAN module receive a message, the storage priority of the received messages is as follows (the storage priority differs between data frames and remote frames).

Table 19-13. Storage Priority for Data Frame Reception

Priority	Conditions
2 (High)	Unmasked message buffer
3	Message buffer linked to mask 0
4	Message buffer linked to mask 1
5	Message buffer linked to mask 2
6 (Low)	Message buffer linked to mask 3

Table 19-14. Storage Priority for Remote Frame Reception

Priority	Conditions
1 (High)	Transmit message buffer
2	Unmasked message buffer
3	Message buffer linked to mask 0
4	Message buffer linked to mask 1
5	Message buffer linked to mask 2
6 (Low)	Message buffer linked to mask 3

A message (data frame or remote frame) is always stored in a receive message buffer with a higher priority, not in a receive buffer with a lower priority. For example, when the unmasked receive message buffer and the message buffer linked to the mask 0 have the same ID, a message is always stored in the unmasked receive message buffer even if the unmasked receive message buffer has already received a message.

When two or more message buffers with the same priority exist in the same CAN module, the priority is as follows.

Table 19-15. Priority of Same Priority Level

Priority	Condition
1 (High)	DN bit of M_STAT register is not set (1)
2 (Low)	DN bit of M_STAT register is set (1)

When two or more message buffers with the same priority exist, the message buffer with the smaller message number takes precedence.

Also, when two or more message buffers with the same ID exist, the message buffer with the smaller message number takes precedence.

19.9 Mask Function

A mask linkage function can be defined for each received message.

This means that there is no need to distinguish between local masks and global masks.

When the mask function is used, the received message's identifier is compared with the message buffer's identifier and the message can be stored in the defined message buffer regardless of whether the mask sets "0" or "1" as a result of the comparison.

When the mask function is operating, a bit whose value is defined as "1" by masking is not subject to the above-mentioned comparison between the received message's identifier and the message buffer's identifier.

However, this comparison is performed for any bit whose value is defined as "0" by masking.

For example, let us assume that all messages that have a standard-format ID in which bit ID27 to ID25 = 0 and bits ID24 and ID22 = 1 are to be stored in message buffer 14 (which is linked by CAN module 1 or mask 1 as was explained in **19.5.6**). The procedure for this example is shown below.

<1> Identifier bits to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

x = don't care

Messages with ID in which bits ID27 to ID25 = 0 and bits ID24 and ID22 = 1 are registered (initialized) in message buffer 14 (see **19.5.5**).

<2> Identifier bits set to message buffer 14 (example)

(Using CAN message ID registers L14 and H14 (M_IDL14 and M_IDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
0	0	0	0	1	0	1	0	0	0	0

ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
0	0	0	0	0	0	0	0	0	0	0

ID6	ID5	ID4	ID3	ID2	ID1	ID0
0	0	0	0	0	0	0

Message buffer 14 is set as a standard-format identifier linked to mask 1 (see **19.5.6**).

<3> Mask setting for CAN module 1 (mask 1) (example)

(Using CAN1 address mask 1 registers L and H (C1MASKL1 and C1MASKH1))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1

CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1

CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
1	1	1	1	1	1	1

1: Do not compare (mask)

0: Compare

Values are written to mask 1 (see **19.5.18**), bits CMID27 to CMID24 and CMID22 = 0 and bits CMID28, CMID23, and CMID21 to CMID0 = 1.

19.10 Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed in the ISO 11898. For details, refer to the **ISO 11898** specifications.

The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link control and medium access control. The composition of these layers is illustrated below.

★ **Figure 19-10. Composition of Layers**

Higher ↕ Lower

Data link layer[Note]	• Logical link control (LLC)	• Acceptance filtering • Overload report • Recovery management
	• Medium access control (MAC)	• Data capsuled/not capsuled • Frame coding (stuffing/not stuffing) • Medium access management • Error detection • Error report • Acknowledgement • Seriated/not seriated
Physical layer		Prescription of signal level and bit description

Note CAN controller specification

19.10.1 Frame format

(1) Standard format frame

In this format 2048 different identifiers can be set.

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2048 messages.

(2) Extended format frame

This format is used to set the identifiers of approx. 5.3 million types.

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers which increases the number of messages that can be handled to 2048×2^{18} messages.
- Extended format frame is set when “recessive: logical level 1” is set for both the SRR and IDE bits in the arbitration field.

19.10.2 Frame types

Following four types of frames are used in CAN protocol.

Table 19-16. Frame Type

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

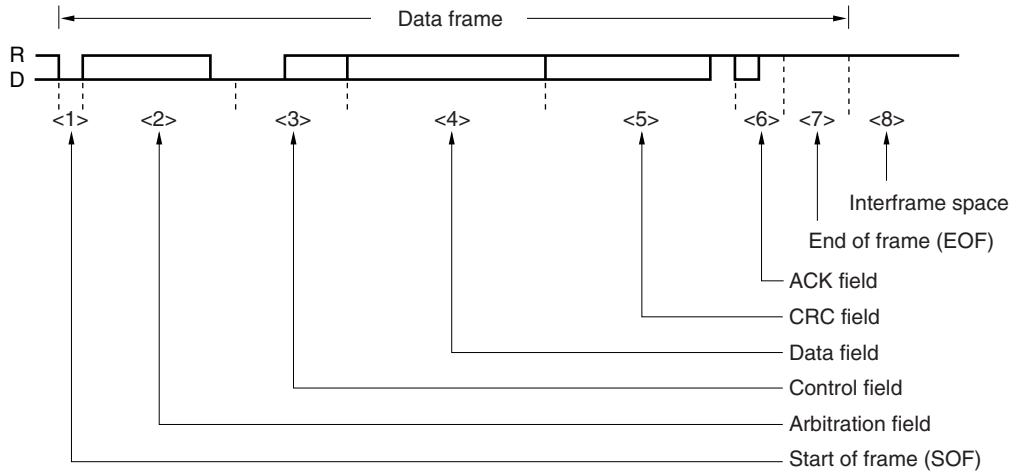
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

19.10.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 19-11. Data Frame

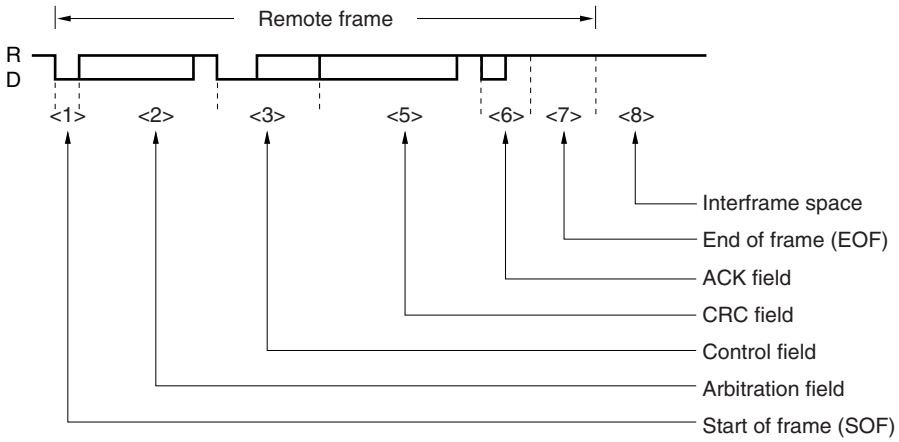


Remark D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 19-12. Remote Frame



Remarks 1. The data field is not transferred even if the control field's data length code is not "0000B".

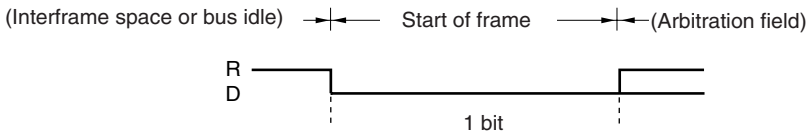
2. D: Dominant = 0
R: Recessive = 1

(3) Description of fields

<1> Start of frame (SOF)

The start of frame field is located at the start of a data frame or remote frame.

Figure 19-13. Start of Frame (SOF)



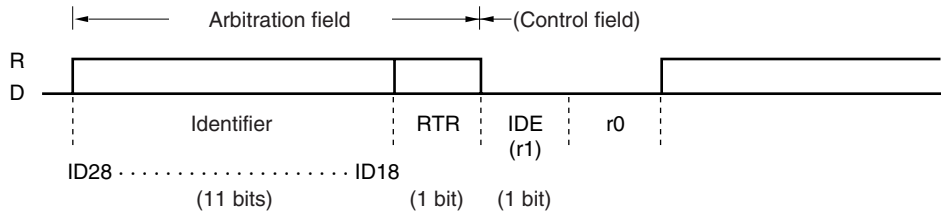
Remark D: Dominant = 0
R: Recessive = 1

- If the dominant level is detected in bus idle mode, the start of frame is recognized.
- If the recessive level is detected at the sample point of the start of frame, it is judged as noise and the bus idle state is entered again.

<2> Arbitration field

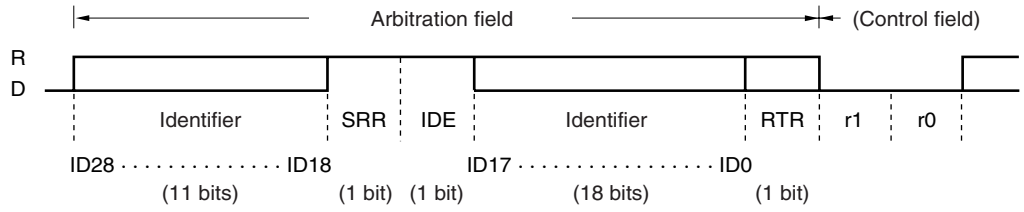
The arbitration field is used to set the priority, data frame or remote frame, and frame format.

Figure 19-14. Arbitration Field (During Standard Format Mode)



Remark D: Dominant = 0
R: Recessive = 1

Figure 19-15. Arbitration Field (in Extended Format Mode)



Cautions 1. ID28 to ID0 are identifier bits.
2. Identifier bits are transferred in MSB-first order.

Remark D: Dominant = 0
R: Recessive = 1

Table 19-17. RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

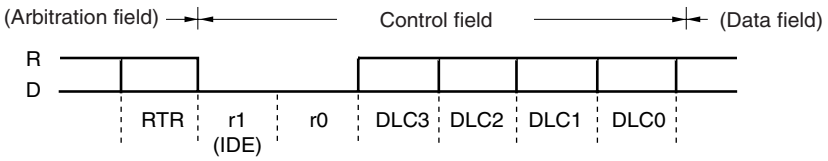
Table 19-18. Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

Frame Format	SRR Bit	IDE Bit	No. of Bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

<3> Control field

The control field sets “N” as the number of data bytes in the data field (N = 0 to 8).

Figure 19-16. Control Field



Remark D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 19-19. Data Length Code Settings

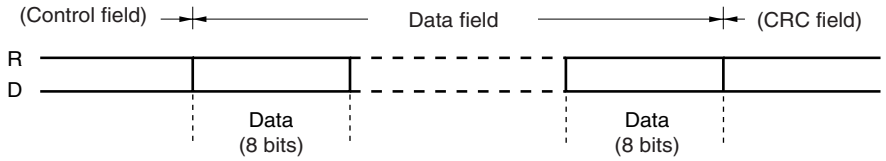
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution In the remote frame, there is no data field even if the data length code is not 0000B.

<4> Data field

The data field contains the amount of data (byte unit) set by the control field. Up to 8 units of data can be set.

Figure 19-17. Data Field

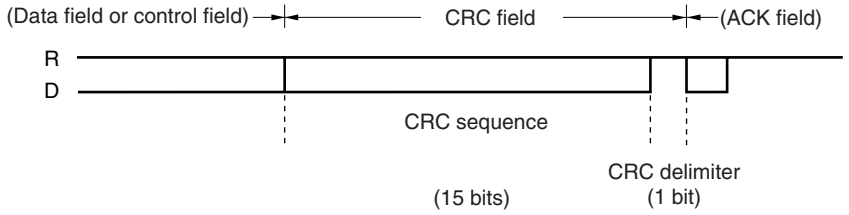


Remark D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 19-18. CRC Field



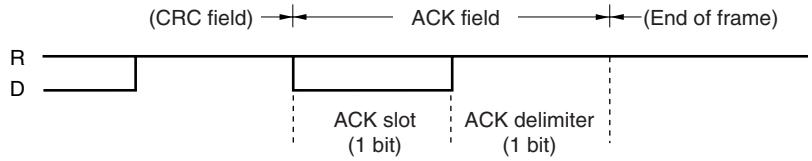
Remark D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.
  $P(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$
- Transmitting node: The CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, or data field is the transferred.
- Receiving node: The CRC sequence calculated using data bits that exclude the stuffing bits in the receive data is compared with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node is transferred to an error frame.

<6> ACK field

The ACK field is used to confirm normal reception.

Figure 19-19. ACK Field



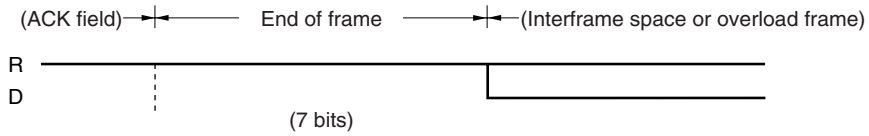
Remark D: Dominant = 0
R: Recessive = 1

- If no CRC error is detected, the receiving node set the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

<7> End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

Figure 19-20. End of Frame (EOF)



Remark D: Dominant = 0
R: Recessive = 1

<8> Interframe space

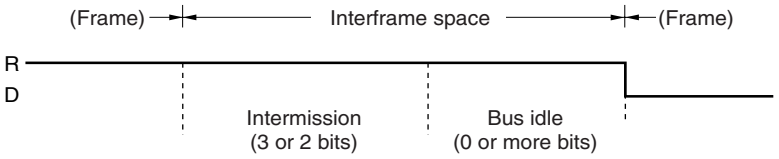
The interframe space is inserted after the data frame, remote frame, error frame, and overload frame to separate one frame from the next.

- The bus status differs depending on the error status.

(a) Error active node

The error active node is composed of a 3-bit intermission field and a bus idle field.

Figure 19-21. Interframe Space (Error Active Node)



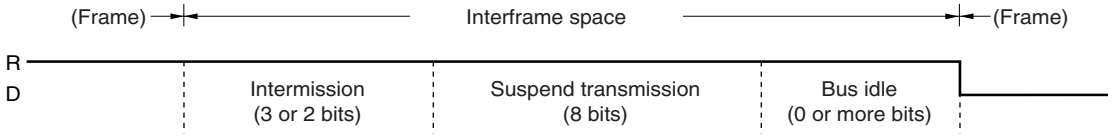
Remarks 1. Bus idle: Status in which the bus is not used by nodes.

2. D: Dominant = 0
R: Recessive = 1

(b) Error passive node

The error passive node is composed of an intermission field, suspend transmission field, and bus idle field.

Figure 19-22. Interframe Space (Error Passive Node)



Remarks 1. Bus idle: Status in which the bus is not used by nodes.
Suspend transmission: 8-bit recessive transmitted from the node in the error passive status.

2. D: Dominant = 0
R: Recessive = 1

- Operation in error status

Table 19-20. Operation in Error Status

Error Status	Operation
Error active	When the bus is idle, the transmit enable mode is set for each node. Transmission then starts from a node.
Error passive	After the 8-bit bus idle field (suspend transmission), the transmit enable mode is set. Receive mode is set if a transmission starts from a different node in bus idle mode (the transmission priority of the local node is lowered).

- Operation when the third bit of the intermission field is dominant level

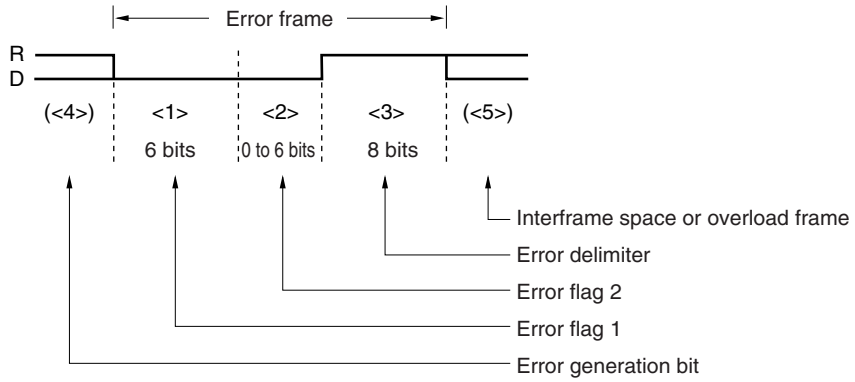
Table 19-21. Operation When Third Bit of Intermission Is Dominant Level

Error Status	Operation
No pending transmissions	A receive operation is performed when start of frame output by the other node is detected.
Pending transmission exists	The identifier is transmitted when start of frame output by the local node is detected.

19.10.4 Error frame

An error frame is output from a node in which an error has been detected.

Figure 19-23. Error Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 19-22. Field Definitions of Error Frame

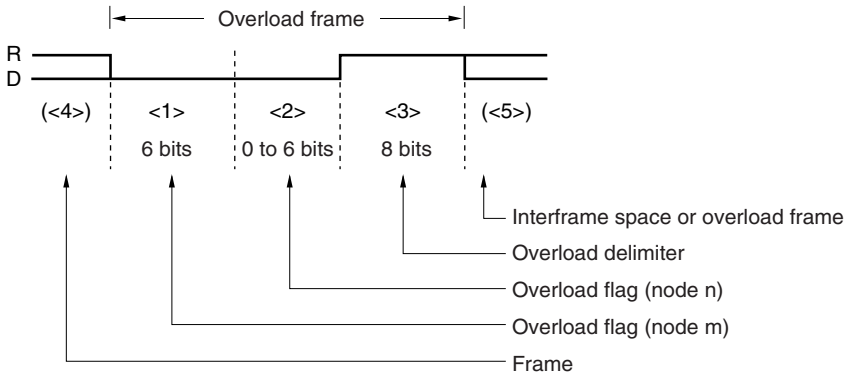
No.	Name	Bit Count	Definition
<1>	Error flag 1	6	Error active node: Consecutive output of 6 dominant-level bits. Error passive node: Consecutive output of 6 recessive-level bits. When the other node outputs a dominant level during output of the passive error flag, the passive error flag does not end until 6 same-level bits are detected consecutively.
<2>	Error flag 2	0 to 6	A node that receives error flag 1 is a node in which bit stuffing errors are detected, after which error flag 2 is output.
<3>	Error delimiter	8	8 consecutive recessive-level bits are output. If a dominant level bit is detected at the eighth bit, an overload frame is transmitted starting at the next bit.
<4>	Error generation bit	–	Bit in which the error is detected. This bit is output following the bit where the error occurred. If the error is a CRC error, it is output following an ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

19.10.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node is not yet ready to receive.
- If a dominant level is detected at the first two bits during intermission mode.
- If a dominant level is detected at the last bit (8th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter.

Figure 19-24. Overload Frame



Remarks 1. Node n/node m: Each node (n ≠ m)

2. D: Dominant = 0

R: Recessive = 1

Table 19-23. Field Definition of Overload Frame

No	Name	Bit count	Definition
<1>	Overload flag starting from node m	6	Consecutive output of 6 dominant-level bits. Output when node m is not ready to receive.
<2>	Overload flag starting from node n	0 to 6	Node n, which has received an overload flag in the interframe space, outputs an overload flag
<3>	Overload delimiter	8	Consecutive output of 8 recessive-level bits. If a dominant level is detected at the eighth bit, an overload frame is sent starting at the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space or overload frame	–	An interframe space or overload frame starts from here.

Remark Node n/node m: Each node (n ≠ m)

19.11 Functions

19.11.1 Determination of bus priority

(1) When one node has starting transmitting

- In bus idle mode, the node that outputs data first starts transmitting.

(2) When several nodes have started transmitting

- The node that outputs the longest string of consecutive dominant-level bits starting from the first bit in the arbitration field has top priority for bus access (dominant-level bits take precedence due to wired-OR bus arbitration).
- The transmitting node compares the arbitration field which it has output and the bus data level.

Table 19-24. Determination of Bus Priority

Matched levels	Transmission continues
Mismatched levels	When a mismatch is detected, data output stops at the next bit, and the operation switches to reception.

(3) Priority between data frame and remote frame

- If a bus conflict occurs between a data frame and a remote frame, the data frame takes priority because its last bit (RTR) is dominant level.

19.11.2 Bit stuffing

Bit stuffing is when one bit of inverted data is added for resynchronization to prevent burst errors when the same level is maintained for five consecutive bits.

Table 19-25. Bit Stuffing

Transmit	When transmitting data frames and remote frames, if the same level is maintained for at least five bits between the start of frame and CRC fields, one bit of data whose level is inverted from the previous level is inserted before the next bit.
Receive	When receiving data frames and remote frames, if the same level is maintained for at least five bits between the start of frame and CRC fields, the next bit of data is deleted before reception is resumed.

19.11.3 Multimasters

Since bus priority is determined based on the identifier, any node can be used as the bus master.

19.11.4 Multi-cast

Even when there is only one transmitting node, the same identifier can be set for several nodes, so that the same data can be received by several nodes at the same time (this is called “multi-casting”).

19.11.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function can be used to set the FCAN controller to sleep (standby) mode to reduce power consumption.

The CAN sleep mode is set via the procedure stipulated in the CAN specifications. The CAN sleep mode can be set to either wake up or not wake up when the bus is operated (this is controlled via CPU access).

19.11.6 Error control function

(1) Types of errors

Table 19-26. Types of Errors

Error Type	Description of Error		Detected Status	
	Detection Method	Detection Condition	Transmit/ Receive	Field/Frame
Bit error	Comparison of output level and bus level (excludes stuff bits)	Mismatch between levels	Transmitting/ receiving nodes	Bus output of data from bits in start of frame to end of frame, error frame, or overload frame
Stuff error	Use stuff bits to check receive data	Six consecutive bits of same-level data	Transmitting/ receiving nodes	Start of frame to CRC sequence
CRC error	Comparison of CRC generated from receive data and received CRC sequence	CRC mismatch	Receiving node	Start of frame to data field
Form error	Check fixed-format field/frame	Detection of inverted fixed format	Receiving node	• CRC delimiter • ACK field • End of frame • Error frame • Overload frame
ACK error	Use transmitting node to check ACK slot	Use ACK slot to detect recessive level	Transmitting node	ACK slot

(2) Error frame output timing

Table 19-27. Error Frame Output Timing

Error Type	Output Timing
Bit error, stuff error, form error, ACK error	Error frame is output at the next bit following the bit where error was detected
CRC error	Error frame is output at the next bit following the ACK delimiter

(3) Handling of errors

The transmitting node retransmits the data frame or remote frame after the error frame has been transmitted.

(4) Error statuses

(a) Types of error statuses

The three types of error statuses are listed below.

Error active
Error passive
Bus off

- The error status is controlled by the transmit error counter and receive error counter (see **19.5.22 CANn error count register (CnERC)**).
- The various error statuses are categorized according to their error counter values.
- When the error counter value reaches 96 or more, the bus status must be tested since the bus may become seriously damaged.
- During start-up, if only one node is active, the error frame and data are repeatedly resent because no ACK is returned even data has been transmitted.

Table 19-28. Types of Error Statuses

Error Status Type	Operation	Error Counter Value	Type of Output Error Flag
Error active	Transmit/ receive	0 to 127	Active error flag (6 consecutive dominant level bits)
Error passive	Transmit	128 to 255	Passive error flag (6 consecutive recessive level bits)
	Receive	128 or more	
Bus off	Transmit	256 or more	Transfer is not possible. When a string of at least 11 consecutive recessive level bits occurs 128 times, the error counter is zero-cleared and error active status can be resumed.

(b) Error counter

The error counter value is incremented each time an error occurs and is decremented when a transmit or receive operation ends normally. The count up/count down timing occurs at the first bit of the error delimiter.

Table 19-29. Error Counter

Status	Transmit Error Counter (TEC7 to TEC0)	Receive Error Counter (REC7 to REC0)
Receiving node has detected an error (except for bit errors that occur in an active error flag or overload flag)	No change	+1
Dominant level is detected following error frame's overload flag output by the receiving node	No change	+8
Transmitting node has sent an error flag [When error counter = ±0] <1> An ACK error was detected in error passive status and a dominant level was not detected during error flag output <2> A stuff error occurs in the arbitration field	+8	No change
Detection of bit error during output of active error flag or overload flag (transmitting node with error active status)	+8	No change
Detection of bit error during output of active error flag or overload flag (receiving node with error active status)	No change	+8
14 consecutive dominant level bits were detected from the start of each node's active error flag or overload flag, followed by detection of eight consecutive dominant level bits. Each node has detected eight consecutive dominant level bits after a passive error flag.	+8	+8
The transmitting node has completed a transmit operation without any errors (±0 if error counter value is 0).	−1	No change
The receiving node has completed a receive operation without any errors.	No change	• −1 (1 ≤ REC7 to REC0 ≤ 127) • ±0 (REC7 to REC0 = 0) • 127 is set (REC7 to REC0 > 127)

(c) Occurrence of bit error during intermission

In this case, an overload frame occurs.

Caution When an error occurs, error control is performed according to the contents of the transmitting and receiving error counters as they existed prior to the error's occurrence. The error counter value is incremented only after an error flag has been output.

19.11.7 Baud rate control function

(1) Prescaler

The FCAN controller of the V850/SC3 includes a prescaler for dividing the clock supplied to the CAN (f_{MEM1}). This prescaler generates a clock (f_{BTL}) that is based on a division ratio ranging from 2 to 128 applied to the CAN base clock (f_{MEM}) when the CnBRP register's TLM bit = 0, and from 2 to 256 when the TLM bit = 1 (see **19.5.25 CANn bit rate prescaler register (CnBRP)**).

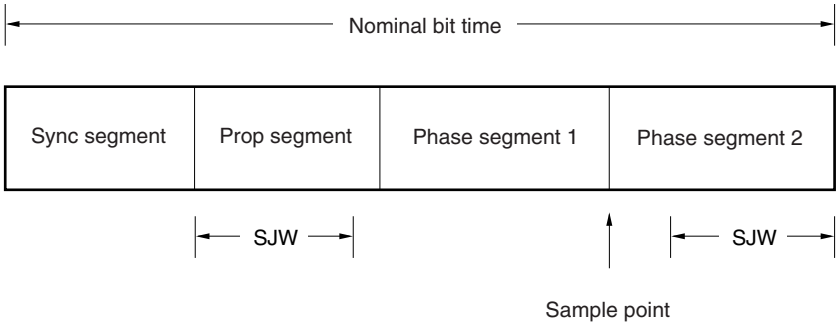
(2) Nominal bit time (8 to 25 time quanta)

The definition of 1 data bit time (1 time quantum) is shown below.

★ **Caution When selecting f_{MEM1} as the clock (f_{MEM}) to the memory access controller using the CAN main clock select register (CGCS) and $f_{MEM}/2$ as the CAN protocol layer base system clock (f_{BTL}) using the CANn bit rate prescaler register (CnBRP), set $f_{XX} \leq 16$ MHz to make one data bit time 8 time quanta. Similarly, set $f_{XX} \leq 18$ MHz to make one data bit time 9 time quanta.**
If used with a larger frequency than above, the baud rate exceeds 1 Mbps, which is the maximum value of the CAN protocol.

Remark 1 time quantum = $1/f_{BTL}$

Figure 19-25. Nominal Bit Time



Segment name	Segment length	Description
Sync segment (Synchronization Segment)	1	This segment begins when resynchronization occurs.
Prop segment (Propagation Segment)	1 to 8 (programmable)	This segment is used to absorb the delays caused by the output buffer, CAN bus, and input buffer. It is set to return an ACK signal until phase segment 1 begins. Prop segment time ≥ (output buffer delay) + (CAN bus delay) + (input buffer delay)
Phase segment 1 (Phase Buffer Segment 1)	1 to 8 (programmable)	This segment is used to compensate for errors in the data bit time. It accommodates a wide margin or error but slows down communication speed.
Phase segment 2 (Phase Buffer Segment 2)	Maximum value from phase segment 1 or IPT[Note] (IPT = 0 to 2)	
SJW (reSynchronization Jump Width)	1 to 4 (programmable)	This sets the range for bit synchronization.

★ **Note** IPT: Information Processing Time

IPT is a period in which the current bit level is referenced and judgement for the next processing is performed.

IPT is indicated by the expression below using the supply clock (f_{MEM1}) to CAN.

$IPT = f_{MEM1} \times 3$

(3) Data bit synchronization

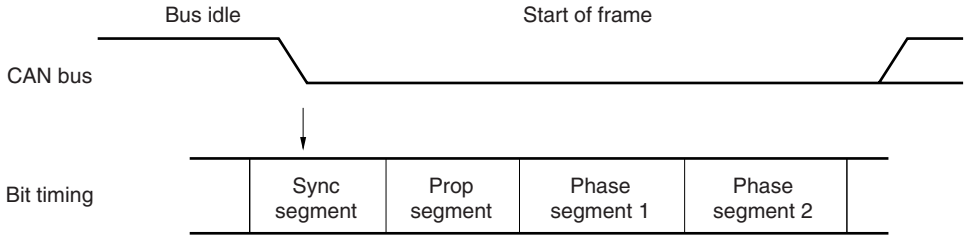
- Since the receiving node has no synchronization signal, synchronization is performed using level changes that occur on the bus.
- As for the transmitting node, data is transmitted in sync with the transmitting node's bit timing.

(a) Hardware synchronization

This is bit synchronization that is performed when the receiving node has detected a start of frame in bus idle mode.

- When a falling edge is detected on the bus, the current bit is assigned to the sync segment and the next bit is assigned to the prop segment. In such cases, synchronization is performed regardless of the SJW (reSynchronization Jump Width).
- Since bit synchronization must be established after a reset or after a wakeup, hardware synchronization is performed only at the first level change that occurs on the bus (for the second and subsequent level changes, bit synchronization is performed as shown below).

Figure 19-26. Coordination of Data Bit Synchronization

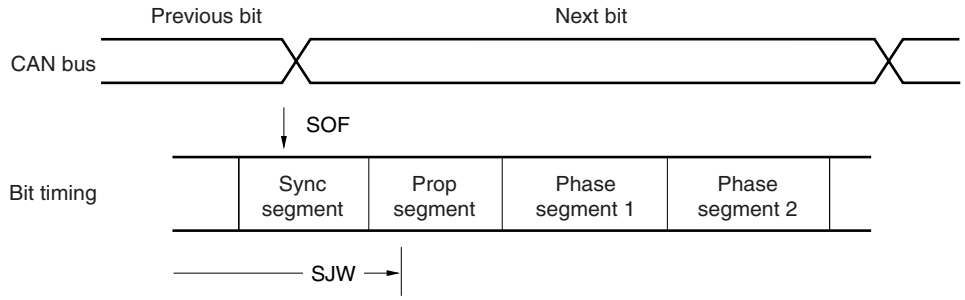


(b) Resynchronization

Resynchronization is performed when a level change is detected on the bus during a receive operation (only when the last sampling was the recessive level).

- The edge's phase error is produced by the relative positions of the detected edge and sync segment.
  <Phase error symbols>
  0: When edge is within sync segment
  Positive: Edge is before sample point (phase error)
  Negative: Edge is after sample point (phase error)
- When the edge is detected as within the bit timing specified by the SJW, synchronization is performed in the same way as hardware synchronization.
- When the edge is detected as extending beyond the bit timing specified by the SJW, synchronization is performed on the following basis.
  When phase error is positive: Phase segment 1 is lengthened to equal the SJW
  When phase error is negative: Phase segment 2 is shortened to equal the SJW
- A "shifting" of the baud rate for the transmitting and receiving nodes moves the relative position of the sample point for data on the receiving node.

Figure 19-27. Resynchronization

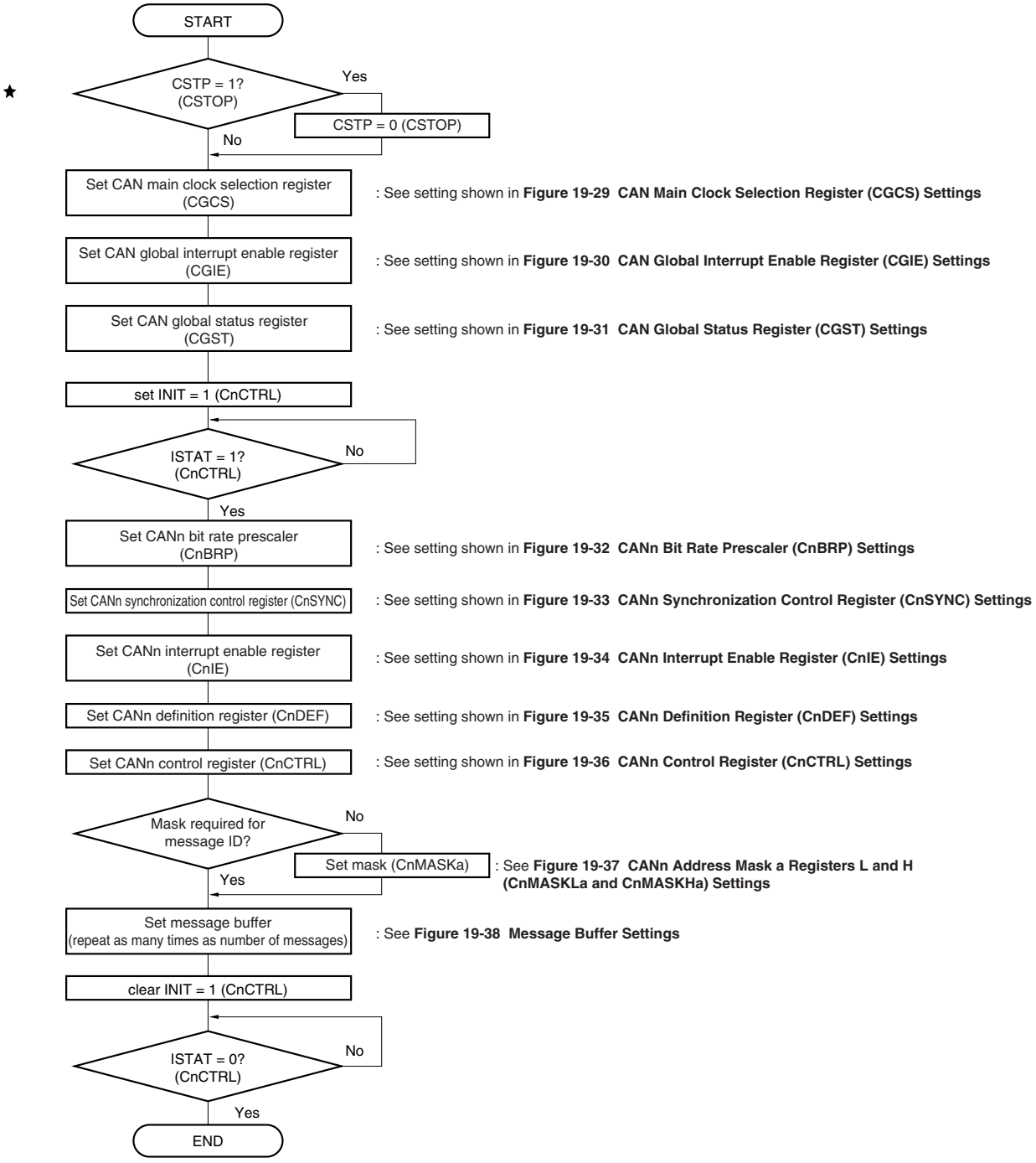


19.12 Operations

19.12.1 Initialization processing

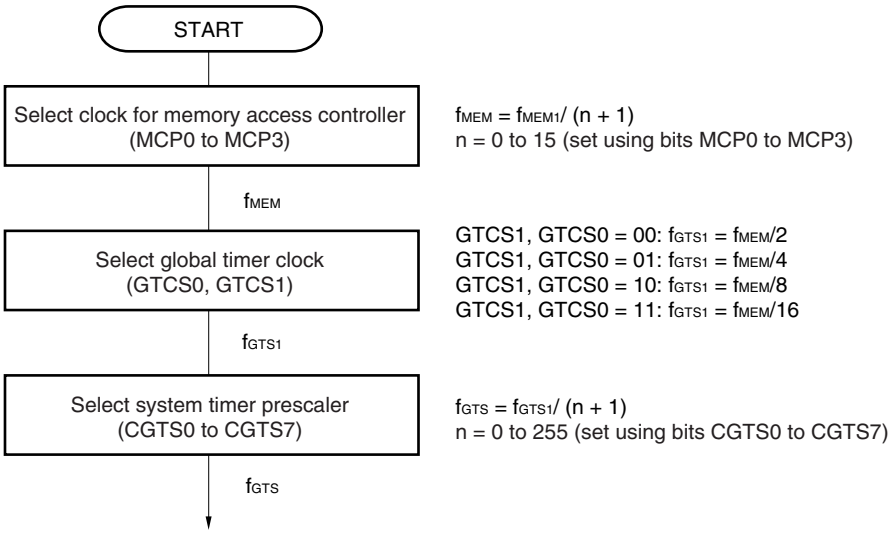
Figure 19-28 shows a flowchart of initialization processing. The register setting flow is shown in Figures 19-29 to 19-41.

Figure 19-28. Initialization Processing



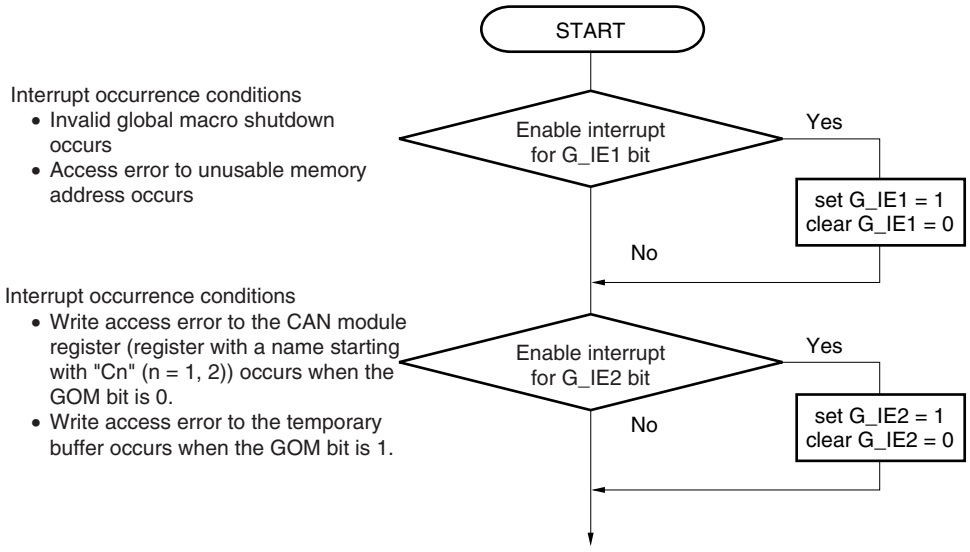
Remark a = 0 to 3
n = 1, 2

Figure 19-29. Setting of CAN Main Clock Select Register (CGCS)



Remark f_{MEM} = CAN base clock
$f_{MEM1} = f_{XX}$: Clock supply to CAN
f_{GTS1} = Global timer clock
f_{GTS} = System timer prescaler

Figure 19-30. Setting of CAN Global Interrupt Enable Register (CGIE)



Remark GOM: Bit of CAN global status register (CGST)
EFSD: Bit of CAN global status register (CGST)
ISTAT: Bit of CANn control register (CnCTRL)

Figure 19-31. Setting of CAN Global Status Register (CGST)

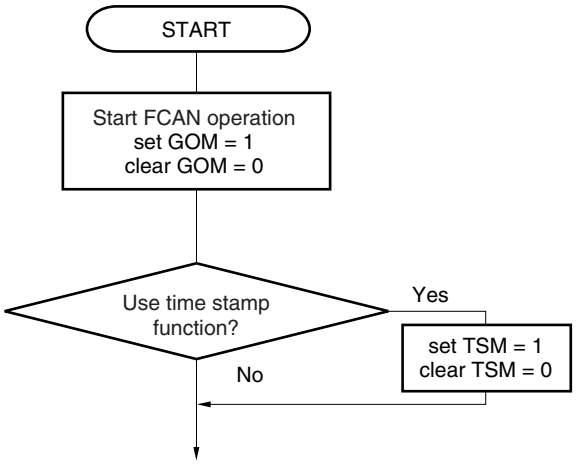
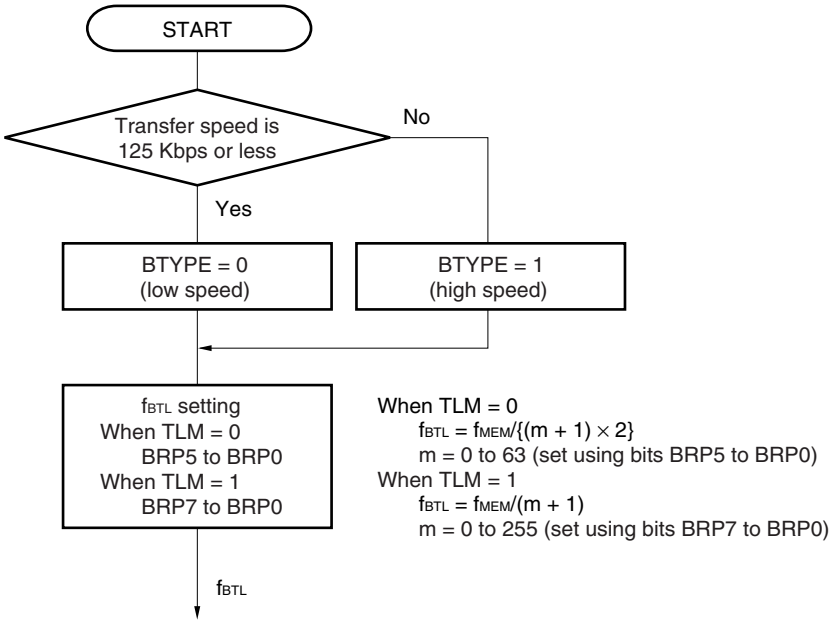


Figure 19-32. Setting of CANn Bit Rate Prescaler (CnBRP)

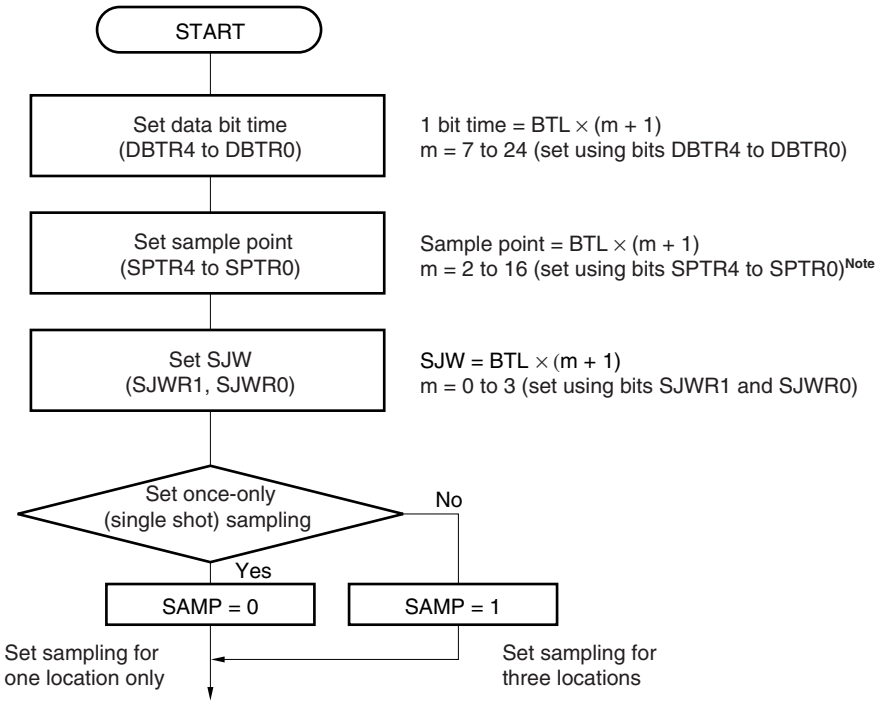


Remarks 1. f_{BTL} = CAN protocol layer basic system clock

f_{MEM} = CAN base clock

2. n = 1, 2

Figure 19-33. Setting of CANn Synchronization Control Register (CnSYNC)

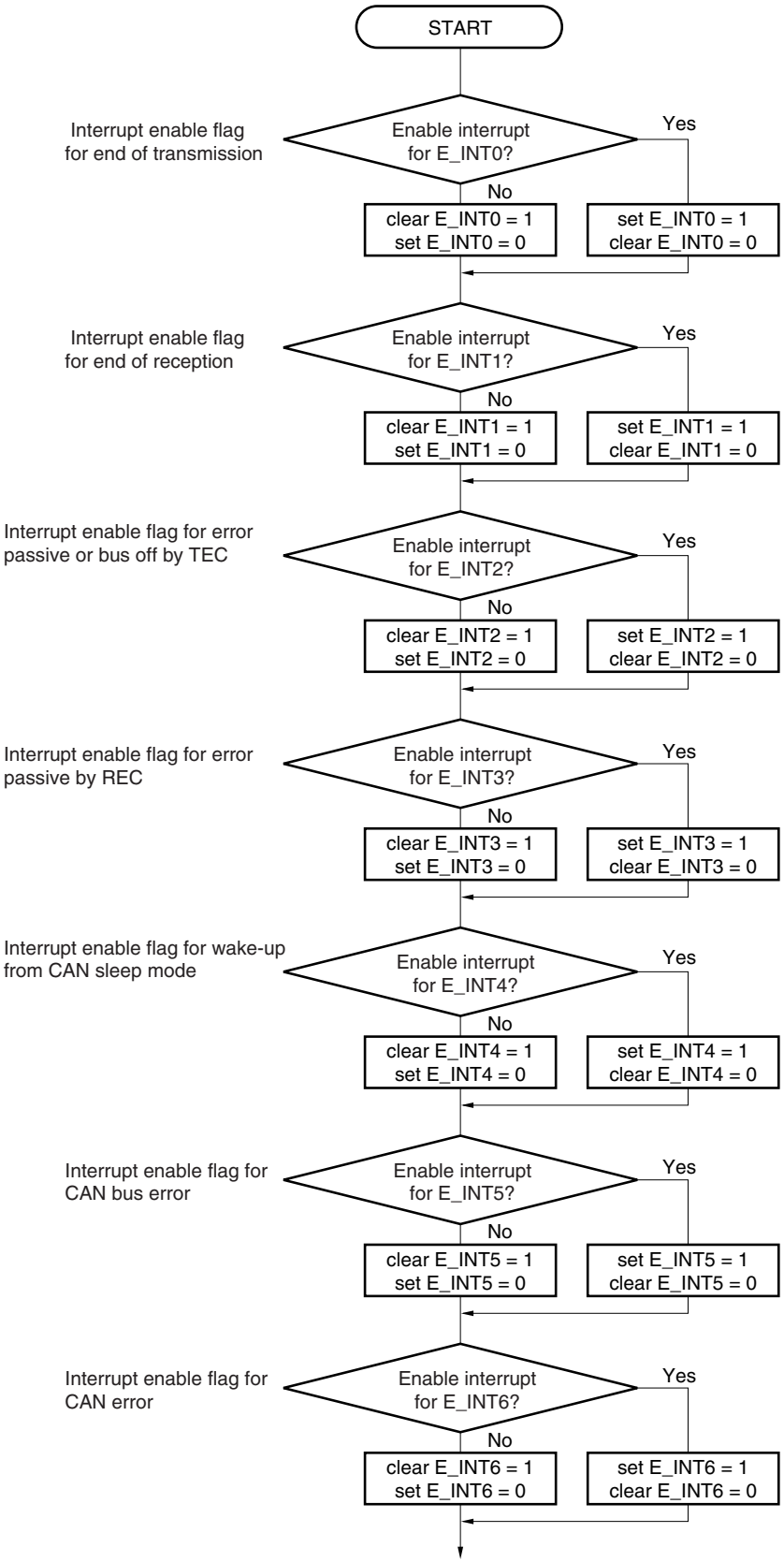


★ **Note** The setting of m = 2, 3 is reserved for setting sample point extension, and is not compliant with the CAN protocol specifications.

Remarks 1. BTL = 1/f_{BTL} (f_{BTL} = CAN protocol layer basic system clock)

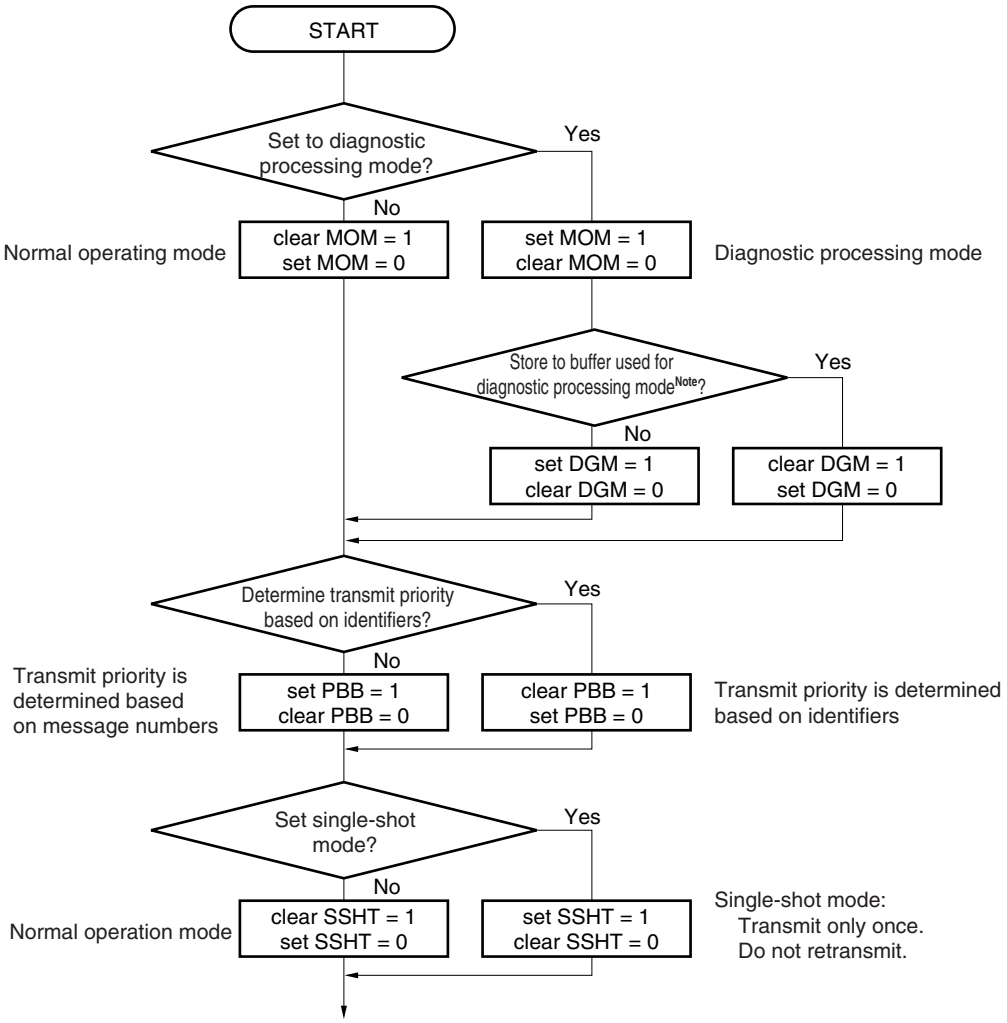
2. n = 1, 2

Figure 19-34. Setting of CANn Interrupt Enable Register (CnIE)



Remark n = 1, 2

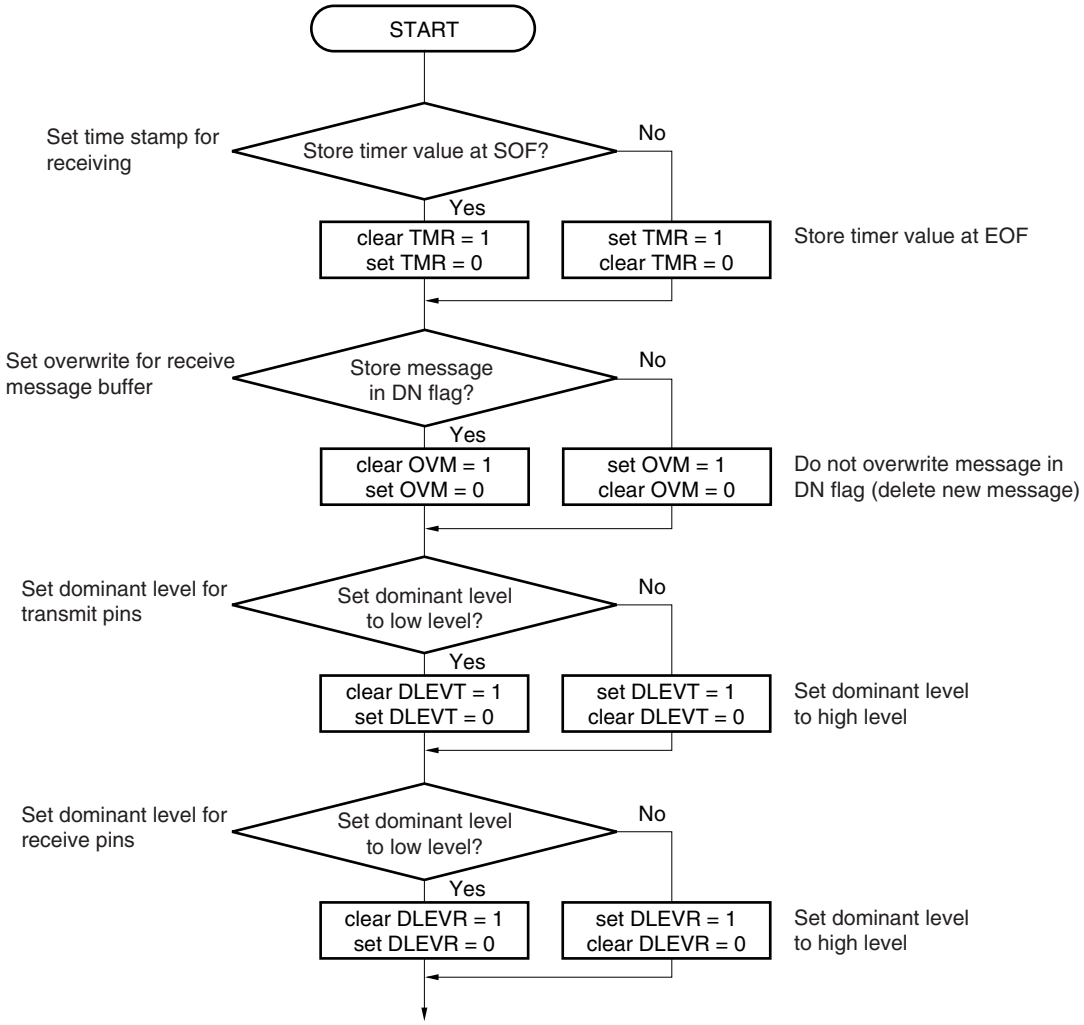
Figure 19-35. Setting of CANn Definition Register (CnDEF)



Note Bits 5 to 3 (MT2 to MT0) in CAN message configuration register m (M_CONFm) are set as “111”

Remark n = 1, 2
m = 00 to 31

Figure 19-36. Setting of CANn Control Register (CnCTRL)



Remark n = 1, 2

Figure 19-37. Setting of CANn Address Mask a Registers L and H (CnMASKLa and CnMASKHa)

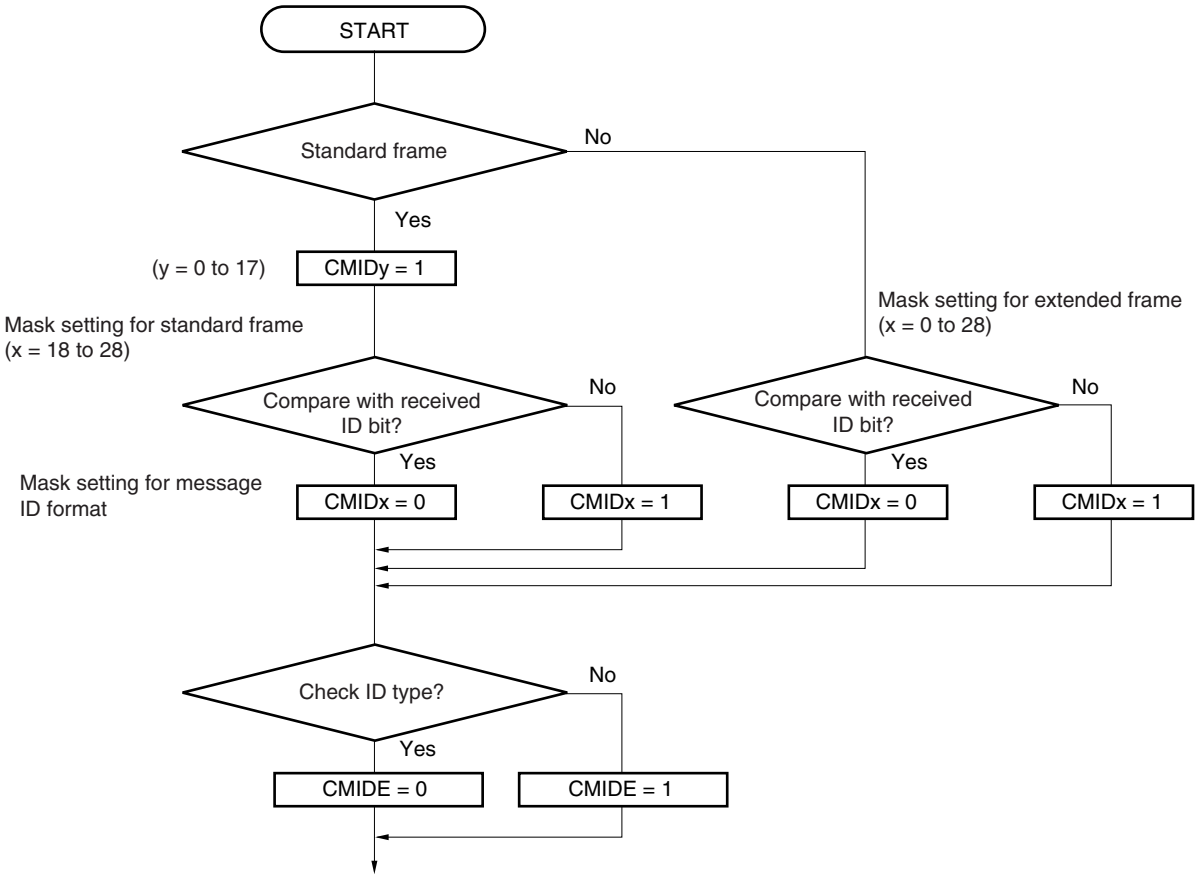
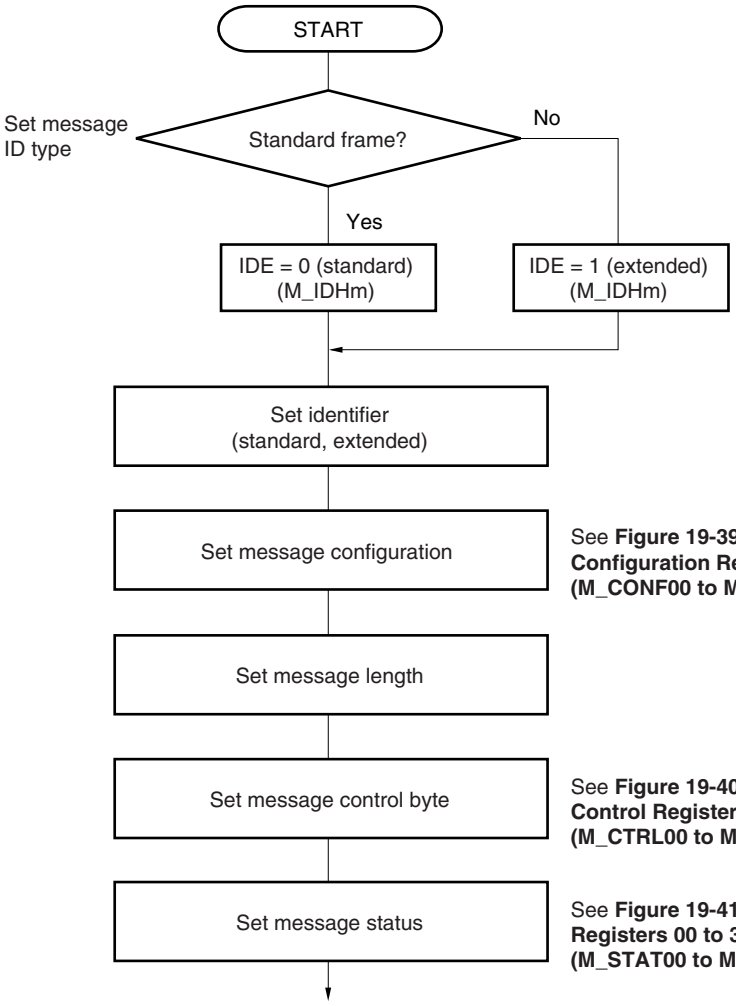


Figure 19-38. Message Buffer Setting



See **Figure 19-39 Setting of CAN Message Configuration Registers 00 to 31 (M_CONF00 to M_CONF31)**

See **Figure 19-40 Setting of CAN Message Control Registers 00 to 31 (M_CTRL00 to M_CTRL31)**

★ See **Figure 19-41 Setting of Message Status Registers 00 to 31 (M_STAT00 to M_STAT31)**

Remark m = 00 to 31

Figure 19-39. Setting of CAN Message Configuration Registers 00 to 31 (M_CONF00 to M_CONF31)

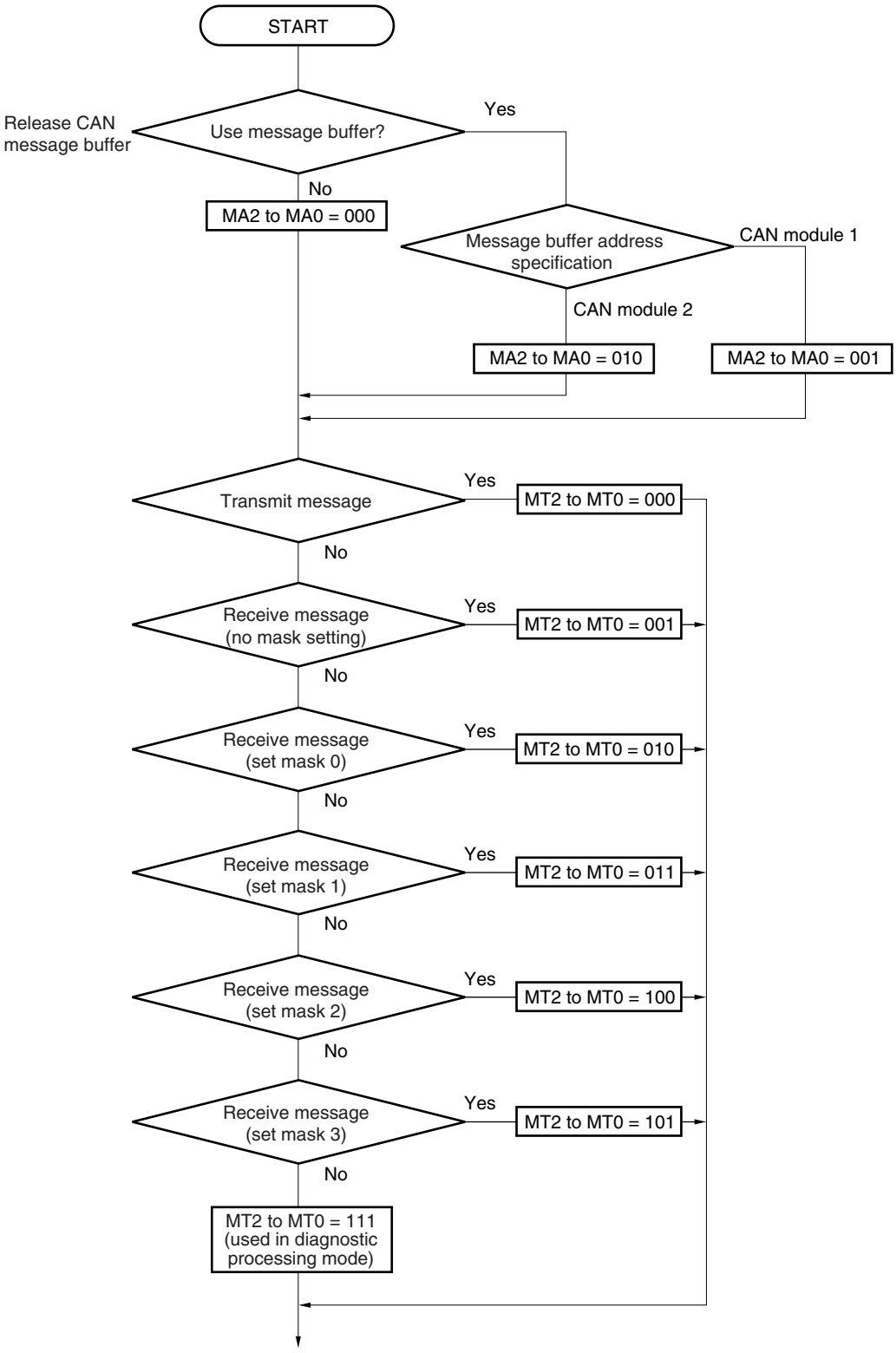
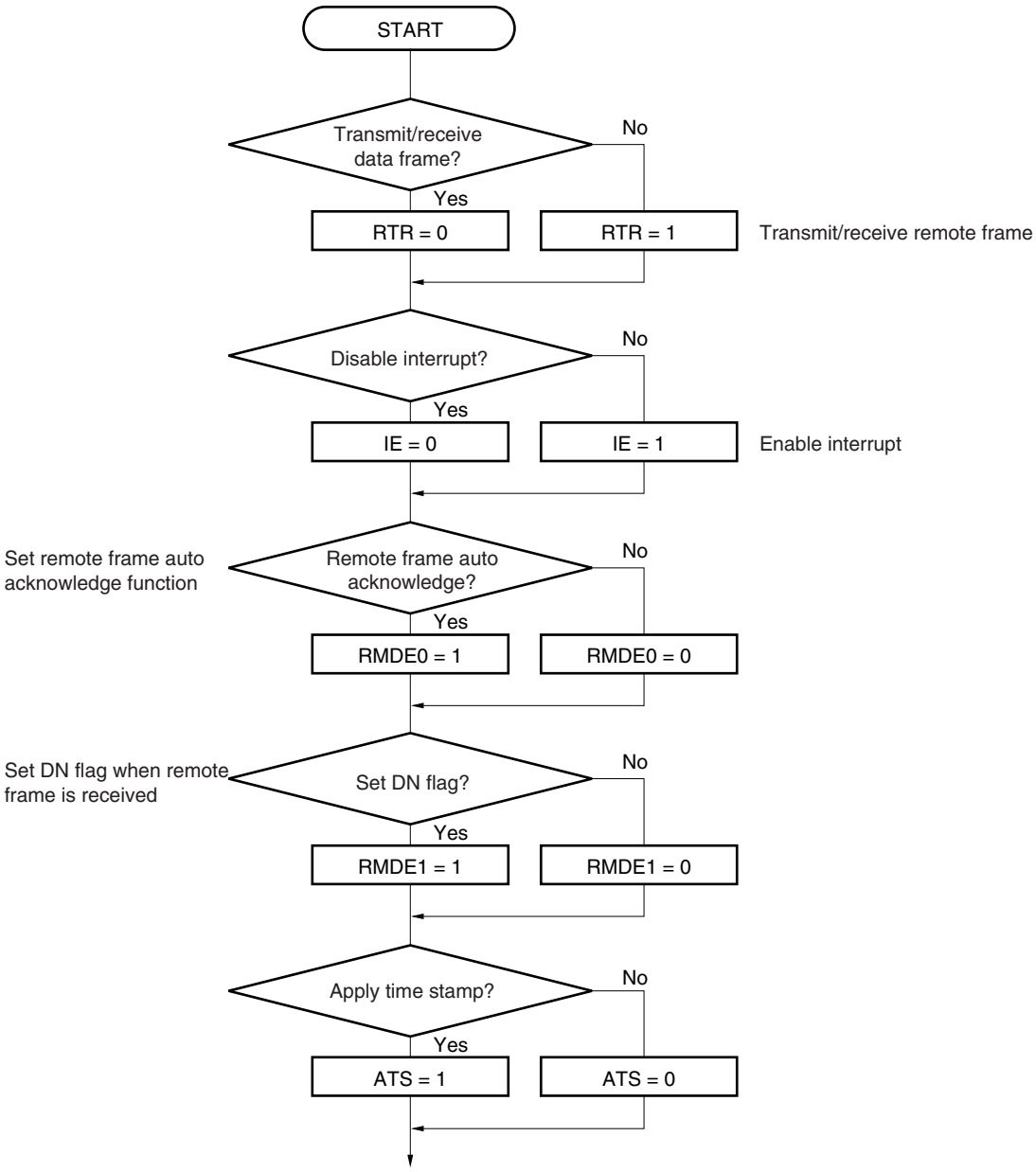
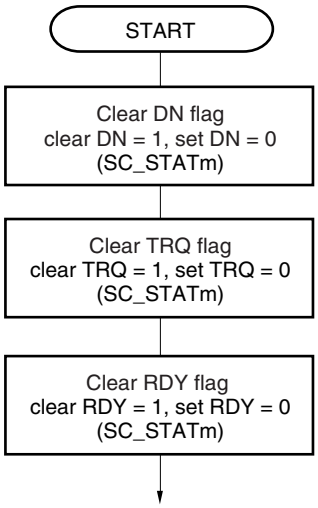


Figure 19-40. Setting of CAN Message Control Registers 00 to 31 (M_CTRL00 to M_CTRL31)



★ **Figure 19-41. Setting of CAN Message Status Registers 00 to 31 (M_STAT00 to M_STAT31)**

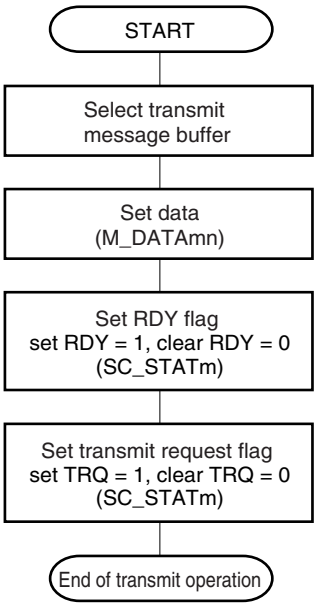


Remark m = 00 to 31

19.12.2 Transmit setting

Transmit messages are output from the target message buffer.

Figure 19-42. Transmit Setting

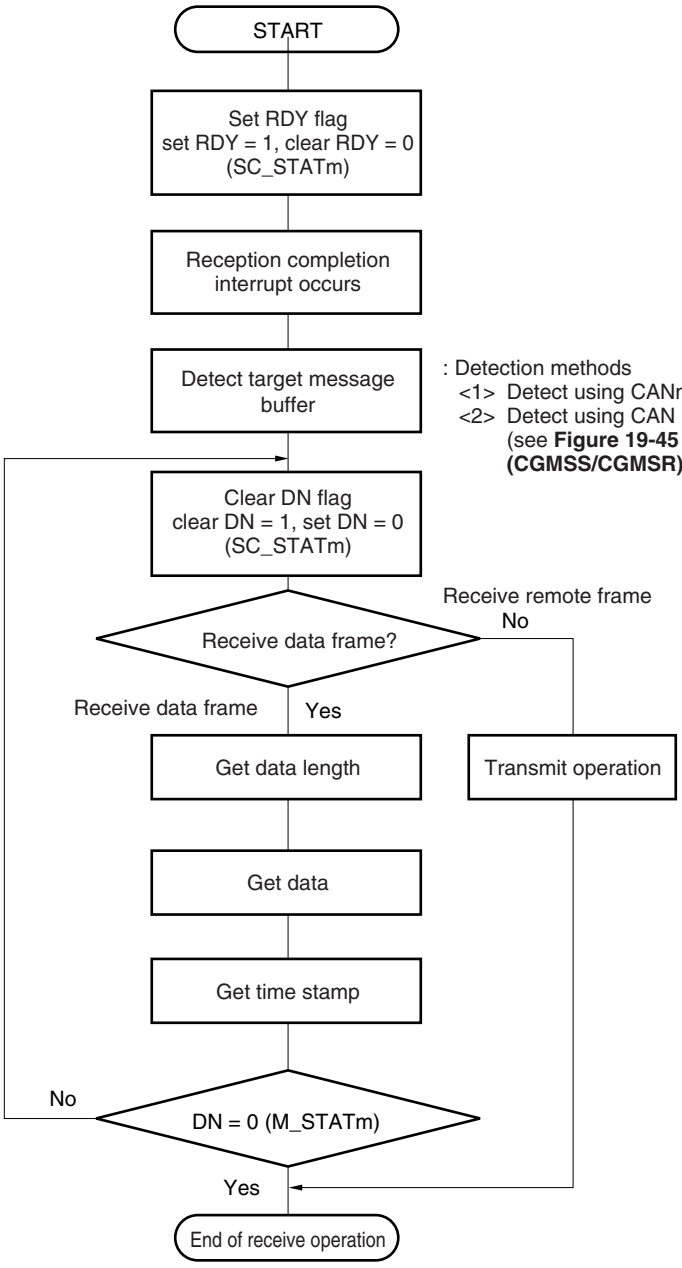


Remark n = 0 to 7
m = 00 to 31

19.12.3 Receive setting

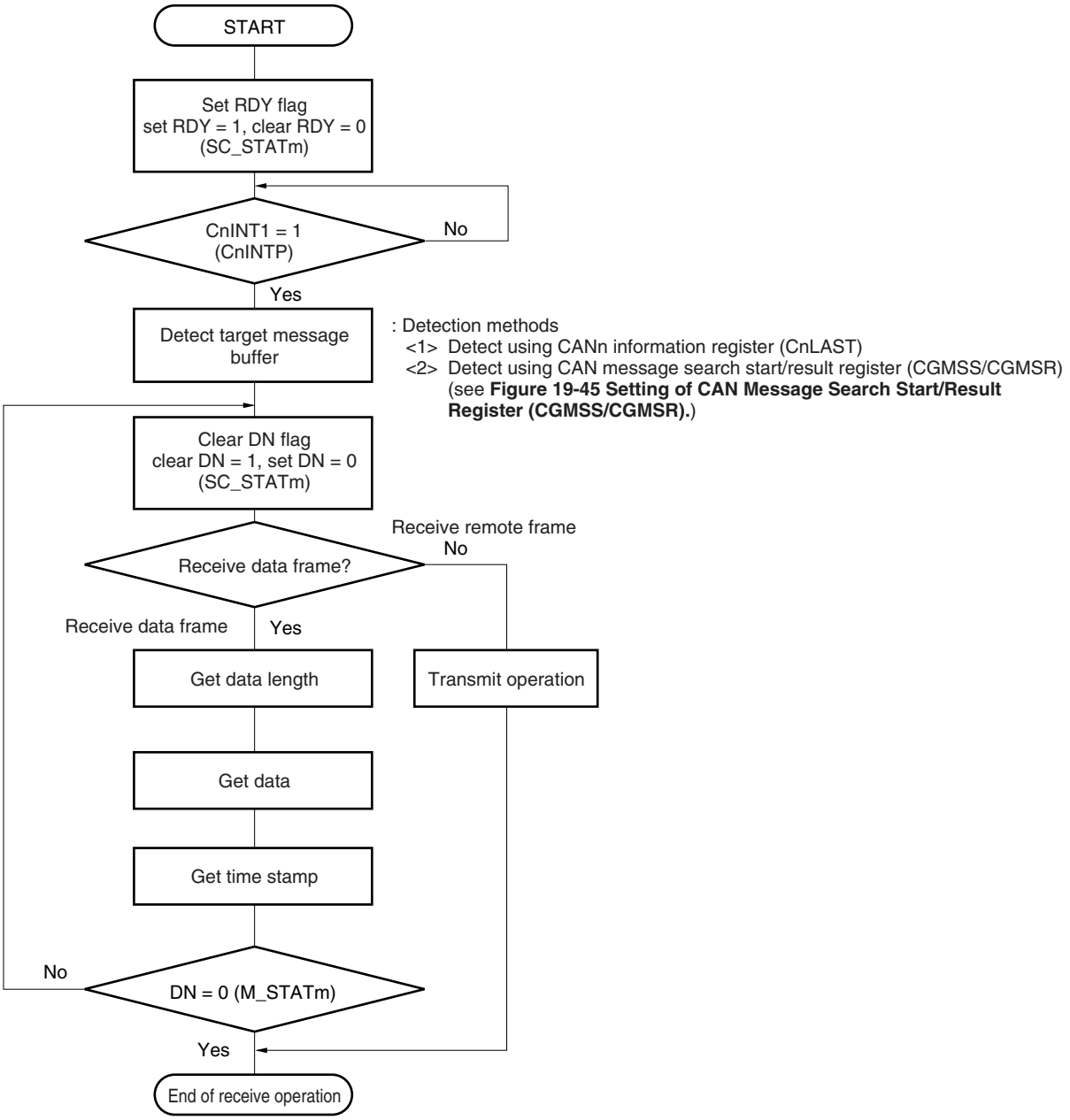
Receive messages are retrieved from the target message buffer.

Figure 19-43. Setting of Receive Operation Using Reception Completion Interrupt



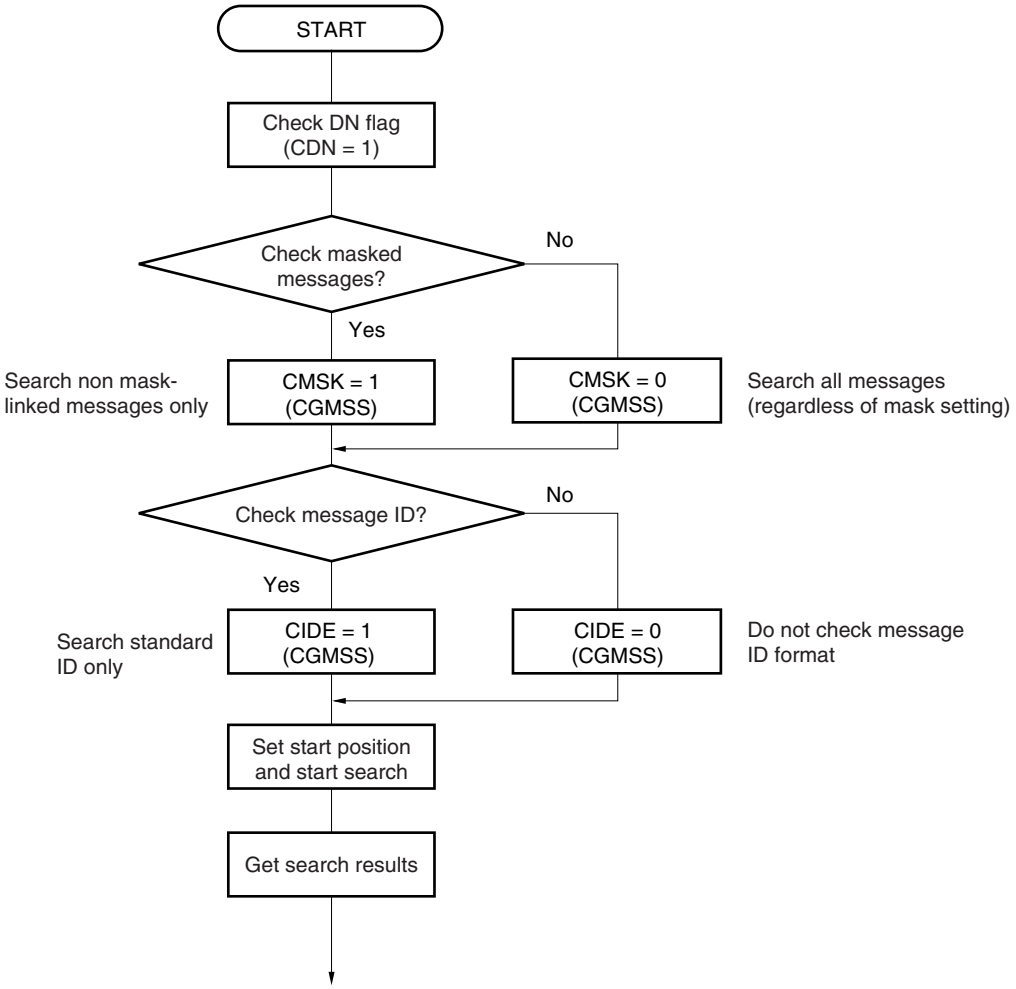
Remark m = 00 to 31

★ **Figure 19-44. Setting Receive Operation Using Reception Polling**



Remark m = 00 to 31

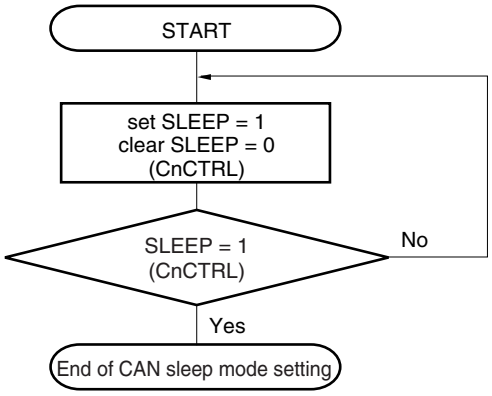
★ **Figure 19-45. Setting of CAN Message Search Start/Result Register (CGMSS/CGMSR)**



19.12.4 CAN sleep mode

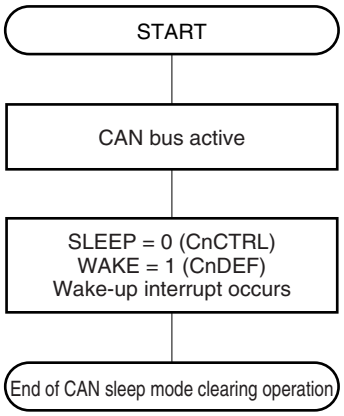
In CAN sleep mode, the FCAN controller can be set to standby mode. A wakeup occurs when there is a bus operation.

Figure 19-46. CAN Sleep Mode Setting



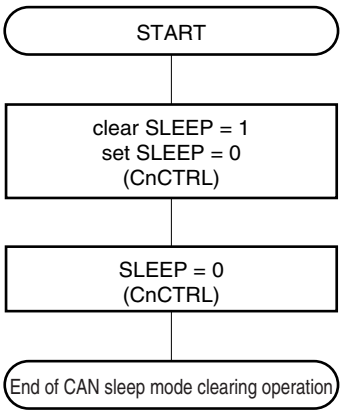
Remark n = 1, 2

Figure 19-47. Clearing CAN Sleep Mode by CAN Bus Active Status



Remark n = 1, 2

Figure 19-48. Clearing CAN Sleep Mode by CPU

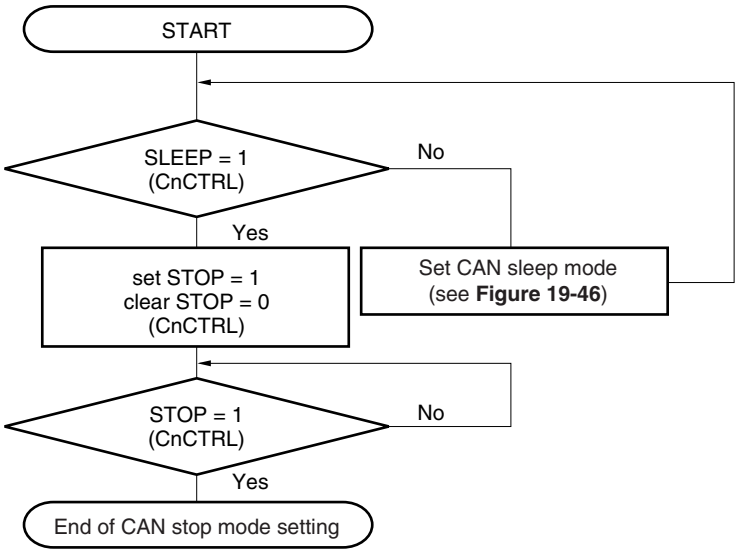


Remark n = 1, 2

19.12.5 CAN stop mode

In CAN stop mode, the FCAN controller can be set to standby mode. No wakeup occurs when there is a bus operation (stop mode is controlled by CPU access only).

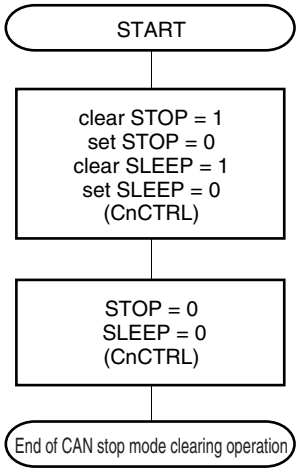
Figure 19-49. CAN Stop Mode Setting



★

Remark n = 1, 2

Figure 19-50. Clearing CAN Stop Mode



★

★ 19.13 Rules for Correct Setting of Baud Rate

The CAN protocol limit values for ensuring correct operation of FCAN are described below. If these limit values are exceeded, a CAN protocol violation may occur, which can result in operation faults. Always make sure that settings are within the range of limit values.

(a) $5 \times BTL \leq SPT$ (sampling point) $\leq 17 \times BTL$ [$4 \leq$ set values of SPTR4 to SPTR0 ≤ 16]
(b) $8 \times BTL \leq DBT$ (data bit time) $\leq 25 \times BTL$ [$7 \leq$ set values of DBTR4 to DBTR0 ≤ 24
(c) SJW (synchronization jump width) $\leq DBT - SPT$
(d) $2 \leq (DBT - SPT) \leq 8$

Remark $BTL = 1/f_{BTL}$ (f_{BTL}: CAN protocol layer basic system clock)
SPTR4 to SPTR0 (bits 9 to 5 of the CANn synchronization control register (CnSYNC))
DBTR4 to DBTR0 (bits 4 to 0 of the CANn synchronization control register (CnSYNC))

(1) Example of FCAN baud rate setting (when CnBRP register's TLM bit = 0)

The following is an example of how correct settings for the CnBRP register and CnSYNC register can be calculated.

Conditions from CAN bus:

<1> CAN base clock frequency (f_{MEM}): 16 MHz
<2> CAN bus baud rate: 83 Kbps
<3> Sample point: 80% or more
<4> Synchronization jump width: 3 BTL

First, calculate the ratio between the CAN base clock frequency and the CAN bus baud rate frequency as shown below.

f_{MEM}/CAN bus band rate = 16 MHz/83 kHz $\neq$ 192.77 $\neq 2^6 \times 3$

Set an even number between 2 and 128 to the CnBRP register's bits BRP5 to BRP0 as the setting for the prescaler (CAN protocol layer basic system clock: f_{BTL}), then set a value between 8 and 25 to the CnSYNC register's bits DTBR4 to DBTR0 as the data bit time.
Since it is assumed that the SJW (synchronization jump width) value is 3, the maximum setting for SPT (sample point) is the "data bit time setting minus 3" or less and 17 or less.

(SPT $\leq$ DBT – 3 and SPT $\leq$ 17)

Given the above limit values, the following four settings are possible.

Prescaler	DBT	SPT (MAX.)	Calculated SPT
24	8	5	5/8 = 62.5%
16	12	9	9/12 = 75%
12	16	13	13/16 = 81%
8	24	17	17/24 = 71%

16 MHz/83 Kbps ≅ 192
= 64 × 3 <1>
= 48 × 4 <2>
= 32 × 6 <3>
= 24 × 8 <4>
= 16 × 12 <5>
= 12 × 16 <6>
= 8 × 24 <7>
= 6 × 32 <8>
= 4 × 48 <9>
= 3 × 64 <10>

The settings that can actually be made for the V850/SC3 are in the range from <4> to <7> above (the section enclosed in broken lines).
Among these options in the range from <4> to <7> above, option <6> is the ideal setting when actually setting the register.

(i) Prescaler (CAN protocol layer basic system clock: f_{BTL}) setting

f_{BTL} is calculated as shown below.

- $f_{BTL} = f_{MEM}/\{(a + 1) \times 2\} : [0 \leq a \leq 63]$

Value a is set using bits 5 to 0 (BRP5 to BRP0) of the CnBRP register.

f_{BTL} = 16 MHz/12
= 16 MHz/{(5 + 1) × 2}
thus a = 5

Therefore, CnBRP register = 0005H

(ii) DBT (data bit time) setting

DBT is calculated as shown below.

• DBT = BTL × (a + 1) : [7 ≤ a ≤ 24]

Value a is set using bits 4 to 0 (DBTR4 to DBTR0) of the CnSYNC register.

DBT = BTL × 16
= BTL × (a + 1)
thus a = 15

Therefore, the CnBRP register's bits DBTR4 to DBTR0 = 01111B

Note that 1/DBT = f_{BTL}/16
≅ 1333 kHz/16
≅ 83 Kbps (nearly equal to the CAN bus baud rate)

(iii) SPT (sample point) setting

Given SJW = 3:

SJW ≤ DBT – SPT
3 ≤ 16 – SPT
SPT ≤ 13

Therefore, SPT is set as 13 (max.)

SPT is calculated as shown below.

• SPT = BTL × (a + 1) : [4 ≤ a ≤ 16]

Value a is set using bits 9 to 5 (SPTR4 to SPTR0) of the CnSYNC register.

SPT = BTL × 13
= BTL × (12 + 1)
thus a = 12

Therefore, the CnSYNC register's bits SPTR4 to SPTR0 = 01100B

(iv) SJW (Synchronization Jump Width) setting

SJW is calculated as shown below.

• SJW = BTL × (a + 1) : [0 ≤ a ≤ 3]

Value a is set using bits11 and 10 (SJWR1, SJWR0) of the CnSYNC register.

The CnSYNC register's bits SJWR1 and SJWR0 = BTL × 3
= BTL × (2 + 1)
thus a = 2

Therefore, the CnSYNC register's bits SJWR1 and SJWR0 = 10B

The CnSYNC register settings based on these results are shown in Figure 19-51 below.

Figure 19-51. CnSYNC Register Settings

	15	14	13	12	11	10	9	8
CnSYNC	0	0	0	SAMP	SJWR1	SJWR0	SPTR4	SPTR3
Setting	0	0	0	0	1	0	0	1

	7	6	5	4	3	2	1	0
	SPTR2	SPTR1	SPTR0	DBTR4	DBTR3	DBTR2	DBTR1	DBTR0
Setting	1	0	0	0	1	1	1	1

19.14 Ensuring Data Consistency

When the CPU reads data from CAN message buffers, it is essential for the read data to be consistent.

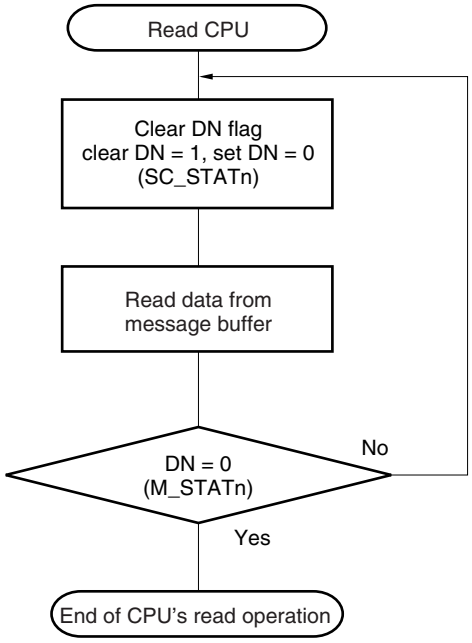
Two methods are used to ensure data consistency: sequential data read and burst read mode.

19.14.1 Sequential data read

When the CPU performs sequential access of a message buffer, data is read from the buffer in the order shown in Figure 19-74 below.

Only the FCAN internal operation can set the M_STATn register's DN bit (1) and only the CPU can clear it (0), so during the read operation the CPU must be able to check whether or not any new data has been stored in the message buffer.

Figure 19-52. Sequential Data Read



Remark n = 00 to 31

19.14.2 Burst read mode

Burst read mode is implemented in the FCAN to enable faster access to complete messages and secure the synchronization of data.

Burst read mode starts up automatically each time the CPU reads the M_DLCn register and data is then copied from the message buffer area to a temporary read buffer.

Data continues to be read from the temporary buffer as long as the CPU keeps directly incrementing (+1) the read address (in other words, when data is read in the following order: M_DLCn register → M_CTRLn register → M_TIMEn register → M_DATAn0 to M_DATAn7 registers → M_IDLn, M_IDHn register), and reads more data.

If these linear access rules are not followed or if access is attempted to an address that is lower than the MIDHn register's address (such as the M_CONFn register or M_STATn register), burst read mode becomes invalid.

Cautions 1. 16-bit read access is required for the entire message buffer area when using the burst read mode. If 8-bit access (byte read operation) is attempted, burst read mode does not start up even if the address is linearly incremented (+1) as described above.

★ **2. Be sure to read out the value of FCAN control registers other than the M_DLCn register before starting the burst read mode.**

Remark n = 00 to 31

19.15 Interrupt Conditions

19.15.1 Interrupts that occur for FCAN controller

When interrupts are enabled (condition <1>: the M_CTRLm register's IE bit = 1, conditions other than <1>: C_IE register's interrupt flags = 1), interrupts will occur under the following conditions (m = 00 to 31).

<1> Message-related operation has succeeded
- When a message has been received in the receive message buffer
- When a remote frame has been received in the transmit message buffer
  (only when auto acknowledge mode has not been set, i.e., when the M_CTRLm register's RMDE0 bit = 0)
- When a message has been transmitted from the transmit message buffer

<2> When a CAN bus error has been detected
- Bit error
- Bit stuff error
- Form error
- CRC error
- ACK error

<3> When the CAN bus mode has been changed
- Error passive status elapsed while FCAN was transmitting
- Bus off status was set while FCAN was transmitting
- Error passive status elapsed while FCAN was receiving

<4> Internal error
- Overrun error

19.15.2 Interrupts that occur for global CAN interface

Interrupts occur for the global CAN interface under the following conditions.

<1> Interrupt source generated at GINT1 (GINTP register)
- Access to unused area in the CAN module
- When clearing (0) the GOM bit is attempted with the EFSD bit of the CGST register = 0, when there is even one CAN module not initialized (INIT bit of CnCTRL register = 0)

<2> Interrupt source generated at GINT2 (CGINTP register)
- Write access to the CAN module register (register with a name starting with "Cn" (n = 1, 2)), when the GOM bit of the CGST register = 0
- Write access to a temporary buffer area when the GOM bit of the CGST register = 1

19.16 How to Shutdown FCAN Controller

The following procedure should be used to stop CAN bus operations in order to stop the clock supply to the CAN interface (to set low power mode).

<1> Set FCAN controller initialization mode
- Set initialization mode (INIT bit = 1 in CnCTRL register (set INIT bit = 1, clear INIT bit = 0)) (n = 1, 2)

<2> Stop time stamp counter
- Set TSM bit = 0 in CGST register (set TSM = 0, clear TSM = 1)

<3> Stop CAN interface
- Set GOM bit = 0 in CGST register (set GOM = 0, clear GOM = 1)
- Stop CAN clock

★ **Caution If the above procedure is not performed correctly, the CAN interface (in active status) can cause operation faults.**

19.17 Cautions on Use

<1> Bit manipulation is prohibited for all FCAN controller registers.

<2> Be sure to properly clear (0) all interrupt request flags[Note] in the interrupt routine. If these flags are not cleared (0), subsequent interrupt requests may not be generated. Note also that if an interrupt is generated at the same time as a CPU clear operation, that interrupt request flag will not be cleared (0). It is therefore important to confirm that interrupt request flags have been properly cleared (0).

Note See **19.5.9 CAN interrupt pending register (CCINTP)**, **19.5.10 CAN global interrupt pending register (CGINTP)**, and **19.5.11 CANn local interrupt pending register (CnINTP)**.

<3> When the CSTP bit of the CSTOP register is set (1), wakeup from the CAN sleep mode (SLEEP bit of CANn control register (CnCTRL) = 1) can be performed in accordance with a change on the CAN bus.

★ <4> If the OS (OSEK/COM) is not used, be sure to execute the following processing.

[When CAN communication is performed using an interrupt routine]
- Clear (0) the following interrupt pending bits at the start of the corresponding interrupt routine.
  - CnINTm bit of CnINTP register (n = 1, 2, m = 0 to 6)
  - CINTm bit of CGINTP register (n = 1, 2, m = 1 to 3)
- Clear (0) the following enable bits during the corresponding interrupt routine.
  - E_INTm bit of CnIE register (n = 1, 2, m = 0 to 6)
  - G_IEn bit of CGIE register (n = 1, 2)

[When CAN communication is performed by polling of bits, not using interrupt routines]

- The following interrupt mask flags and interrupt enable bits are used when set (1) (do not clear (0) them).
  - CANMKn bit of CANICn register (n= 1 to 7)
  - E_INTm bit of CnIE register (n = 1, 2, m = 0 to 6)
  - G_IEn bit of CGIE register (n = 1, 2)
  - IE bit of M_CTRLn register (n = 00 to 31)
- Clear (0) the following interrupt pending bits in accordance with procedures (i) to (iii) below.
  - CnINTm bit of CnINTP register (n = 1, 2, m = 0 to 6)
  - GINTn bit of CGINTP register (n = 1 to 3)

(i) Poll the corresponding interrupt request flag.
(ii) If the value of the bit in procedure (i) is 1, clear (0) the corresponding interrupt pending bit.
(iii) After executing procedure (ii), clear (0) the interrupt request flag.

Example CAN reception

(i) Poll until the CANIFm bit of the CANICm register becomes 1 (m = 2, 5).
(ii) Clear (0) the CnINT1 bit of the CnINTP register (n = 1, 2).
(iii) Clear (0) the CANIFm bit of the CANICm register.

★ <5> In the V850/SC3, the time stamp function by SOF detection during message transmission/reception cannot be used.

Only the time stamp function by EOF detection during message reception can be used for the V850/SC3. However, only the value captured by the M_TIME register is valid when the TSM bit of the CGST register is set to 1 and the TMR bit of the CnCTRL register is set to 1.

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	V_{DD0} pin = V_{DD1} pin	–0.3 to +6.0	V
	$ADCV_{DD}$	$ADCV_{DD}$ pin = V_{DD}	–0.3 to +6.0	V
	$PORTV_{DD}$	$PORTV_{DD0}$ pin ≤ $PORTV_{DD1}$ pin ≤ $PORTV_{DD2}$ pin	–0.3 to V_{DD}	V
Input voltage	V_{I0}	$PORTV_{DD}$ system pins	–0.3 to $PORTV_{DD}$ + 0.3[Note 1]	V
	V_{I1}	P170 to P176, $\overline{\text{RESET}}$ pins	–0.3 to V_{DD} + 0.3[Note 1]	V
	V_{I2}	V_{PP} pin[Note 2]/MODE pin[Note 3]	–0.3 to +8.5	V
Analog input voltage	V_{AN}	P70 to P77, P80 to P83	–0.3 to $ADCV_{DD}$ + 0.3[Note 1]	V
Output voltage	V_{O0}	$PORTV_{DD}$ system pins	–0.3 to $PORTV_{DD}$ + 0.3[Note 1]	V
	V_{O1}	P170 to P176	–0.3 to V_{DD} + 0.3[Note 1]	V
Output current, low	I_{OL}	Per pin	8.0	mA
		Total for all pins	40	mA
Output current, high	I_{OH}	Per pin	–8.0	mA
		P40 to P47, P50 to P57, P60 to P65, P90 to P96, CLKOUT	–25	mA
		P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117	–25	mA
		P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157	–25	mA
		P170 to P176	–15	mA
Operating ambient temperature	T_A	Normal operation mode	–40 to +85	°C
		Flash memory programming mode[Note 2]	–20 to +85	°C
Storage temperature	T_{stg}	**Note 3**	–65 to +150	°C
		Note 2	–40 to +125	°C

Notes 1. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

2. μPD70F3089Y

3. μPD703068Y, 703069Y, 703088Y, 703089Y

Cautions 1. Avoid direct connections among the IC device output (or I/O) pins and between V_{DD} or V_{CC} and GND. However, direct connections among open-drain and open-collector pins are possible, as are direct connections to external circuits that have timing designed to prevent output conflict with pins that become high-impedance.

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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Operating Conditions

(1) Operating voltage ($PORTV_{DD0} \leq PORTV_{DD1} \leq PORTV_{DD2} \leq V_{DD0} \leq V_{DD1} \leq ADCV_{DD}$)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD}, $ADCV_{DD}$	When all functions are operating		4.5		5.5	V
		When all functions are operating (except the A/D converter)	**Note 1**	3.5		5.5	V
			Note 2	4.0		5.5	V
	$PORTV_{DD}$	$0.5 \leq f_{CPU} \leq 17$ MHz, $f_{XT} = 32.768$ kHz, **Note 3**		3.0		5.5	V
		$0.5 \leq f_{CPU} \leq 20$ MHz, **Note 3**		4.0		5.5	V

Notes 1. μPD703068Y, 703069Y, 703088Y, 703089Y

2. μPD70F3089Y

3. When using the FCAN controller: $PORTV_{DD1} \leq PORTV_{DD2}$ (due to the supply voltage conditions of the in-circuit emulator)

Remark I/O buffer power supply of each pin is shown below.

Power Supply	Corresponding Pins
$PORTV_{DD0}$	P40 to P47, P50 to P57, P60 to P65, P90 to P96. CLKOUT
$PORTV_{DD1}$	P00 to P03, P10 to P17, P30 to P37, P100 to P107, P110 to P117
$PORTV_{DD2}$	P04 to P07, P20 to P27, P120 to P127, P130 to P133, P140 to P147, P150 to P157
V_{DD0}	$\overline{\text{RESET}}$
V_{DD1}	P170 to P176
$ADCV_{DD}$	P70 to P77, P80 to P83

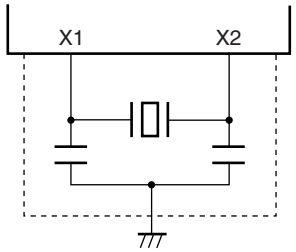
(2) CPU operating frequency

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU operating frequency	f_{CPU}	Main clock operation	0.5		20	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	f_{XX}		4		20	MHz
Oscillation stabilization time	–	When reset is released		$2^{18}/f_{XX}$		s
	–	When STOP mode is released		**Note**		s

Note The TYP. value differs depending on the setting of the oscillation stabilization time select register (OSTS).

Cautions 1. Main clock oscillator operates on the output voltage of the on-chip regulator. External clock input is prohibited.

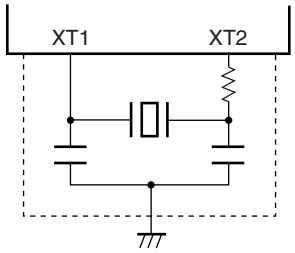
2. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- **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. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

(2) Subclock oscillator (T_A = –40 to +85°C)

(a) Connection of crystal resonator



Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{XT}			32.768		kHz
Oscillation stabilization time	–	When reset is released		10		s

Cautions 1. Subclock oscillator operates on the output voltage of the on-chip 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. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (T_A = –40 to +85°C, $PORTV_{DD0}$ = $PORTV_{DD1}$ = $PORTV_{DD2}$ = 3.0 to 5.5 V, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD0} = V_{DD1} = $ADCV_{DD}$ = 3.5 to 5.5 V, μPD70F3089Y: V_{DD0} = V_{DD1} = $ADCV_{DD}$ = 4.0 to 5.5 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**		0.7$PORTV_{DD}$		$PORTV_{DD}$	V
	V_{IH2}	**Note 2**		0.8$PORTV_{DD}$		$PORTV_{DD}$	V
	V_{IH6}	P70 to P77, P80 to P83		0.7$ADCV_{DD}$		$ADCV_{DD}$	V
	V_{IH7}	**Note 3**		0.7V_{DD}		V_{DD}	V
	V_{IH8}	$\overline{\text{RESET}}$ pin		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**		0		0.3$PORTV_{DD}$	V
	V_{IL3}	**Note 2**		0		0.2$PORTV_{DD}$	V
	V_{IL6}	P70 to P77, P80 to P83		0		0.3$ADCV_{DD}$	V
	V_{IL7}	**Note 3**		0		0.3V_{DD}	V
	V_{IL8}	$\overline{\text{RESET}}$ pin		0		0.2V_{DD}	V
Output voltage, high	V_{OH1}	**Note 3**	3.5 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –100 μA	V_{DD} – 0.5			V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –1 mA	V_{DD} – 1.0			V
	V_{OH2}	**Note 4**	3.0 V ≤ $PORTV_{DD}$ ≤ 4.0 V, I_{OH} = –100 μA	$PORTV_{DD}$ – 0.5			V
			4.0 V ≤ $PORTV_{DD}$ ≤ 5.5 V, I_{OH} = –1 mA	$PORTV_{DD}$ – 1.0			V
Output voltage, low	V_{OL1}	**Note 3**	3.5 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –100 μA			0.5	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –1 mA			0.5	V
	V_{OL2}	**Note 4**	3.0 V ≤ $PORTV_{DD}$ ≤ 4.0 V, I_{OH} = –100 μA			0.5	V
			4.0 V ≤ $PORTV_{DD}$ ≤ 5.5 V, I_{OH} = –1 mA			0.5	V

Notes 1. P11, P14, P16, P21, P25, P27, P33, P40 to P47, P50 to P57, P60 to P65, P90 to P96, P110 to P114, P116, P122, P125 to P127, P130 to P133, P141, P155 and their alternate-function pins

2. P00 to P03, P04 to P07, P10, P12, P13, P15, P17, P20, P22 to P24, P26, P30 to P32, P34 to P37, P100 to P107, P115, P117, P120, P121, P123, P124, P140, P142 to P147, P150 to P154, P156, P157, and their alternate-function pins

3. P170 to P176 and their alternate-function pins

4. All output pins other than P170 to P176 and their alternate-function pins

DC Characteristics (T_A = –40 to +85°C, $PORTV_{DD0}$ = $PORTV_{DD1}$ = $PORTV_{DD2}$ = 3.0 to 5.5 V, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD0} = V_{DD1} = $ADCV_{DD}$ = 3.5 to 5.5 V, μPD70F3089Y: V_{DD0} = V_{DD1} = $ADCV_{DD}$ = 4.0 to 5.5 V) (2/2)

Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high		I_{IH1}	**Note 1**	$V_{IN} = V_{DD}$			5.0	μA
			Note 2	$V_{IN} = PORTV_{DD}$				
			Note 3	$V_{IN} = ADCV_{DD}$				
Input leakage current, low		I_{IL1}	**Notes 1, 2**	V_{IN} = 0 V			–5.0	μA
Output off-leakage current		I_{L1}	**Note 4**	$V_{OH} = PORTV_{DD}$			5.0	μA
Pull-up resistor		R_{L1}	**Note 5**	V_{IN} = 0 V	10	30	100	kΩ
Supply current	μPD703068Y, μPD703069Y μPD703088Y, μPD703089Y	I_{DD1}	In normal operation mode[Note 6]			25	40	mA
		I_{DD2}	In HALT mode[Note 7]			10	20	mA
		I_{DD3}	In IDLE mode[Note 8]			1	4	mA
		I_{DD4}	In software STOP mode[Note 9]			8	100	μA
		I_{DD5}	In normal mode (subclock operation)[Note 10]			50	150	μA
		I_{DD6}	In IDLE mode (subclock operation)[Note 11]			13	120	μA
	μPD70F3089Y	I_{DD1}	In normal operation mode[Note 6]			42	60	mA
		I_{DD2}	In HALT mode[Note 7]			14	28	mA
		I_{DD3}	In IDLE mode[Note 8]			1	4	mA
		I_{DD4}	In software STOP mode[Note 9]			15	100	μA
		I_{DD5}	In normal mode (subclock operation)[Note 10]			300	600	μA
		I_{DD6}	In IDLE mode (subclock operation)[Note 11]			170	340	μA

Notes 1. P170 to P176, $\overline{\text{RESET}}$, and their alternate-function pins

2. All input pins other than P170 to P176, $\overline{\text{RESET}}$ and their alternate-function pins

3. P70 to P77, P80 to P83

4. P10, P12, P20, P22 (in N-ch open drain mode)

5. P100 to P107 (in key return mode)

6. $f_{CPU} = f_{XX}$ = 20 MHz, $V_{IN} = V_{CPUREG}$, peripheral functions operating (except FCAN controller)

7. $f_{CPU} = f_{XX}$ = 20 MHz, $V_{IN} = V_{CPUREG}$, CPU stopped, peripheral functions operating (except FCAN controller)

8. f_{XX} = 20 MHz, $V_{IN} = V_{CPUREG}$, all peripheral functions stopped (watch timer operating)

9. f_{XT} = 32.768 kHz, $V_{IN} = V_{CPUREG}$, main clock oscillator stopped, all peripheral functions stopped (watch timer operating)

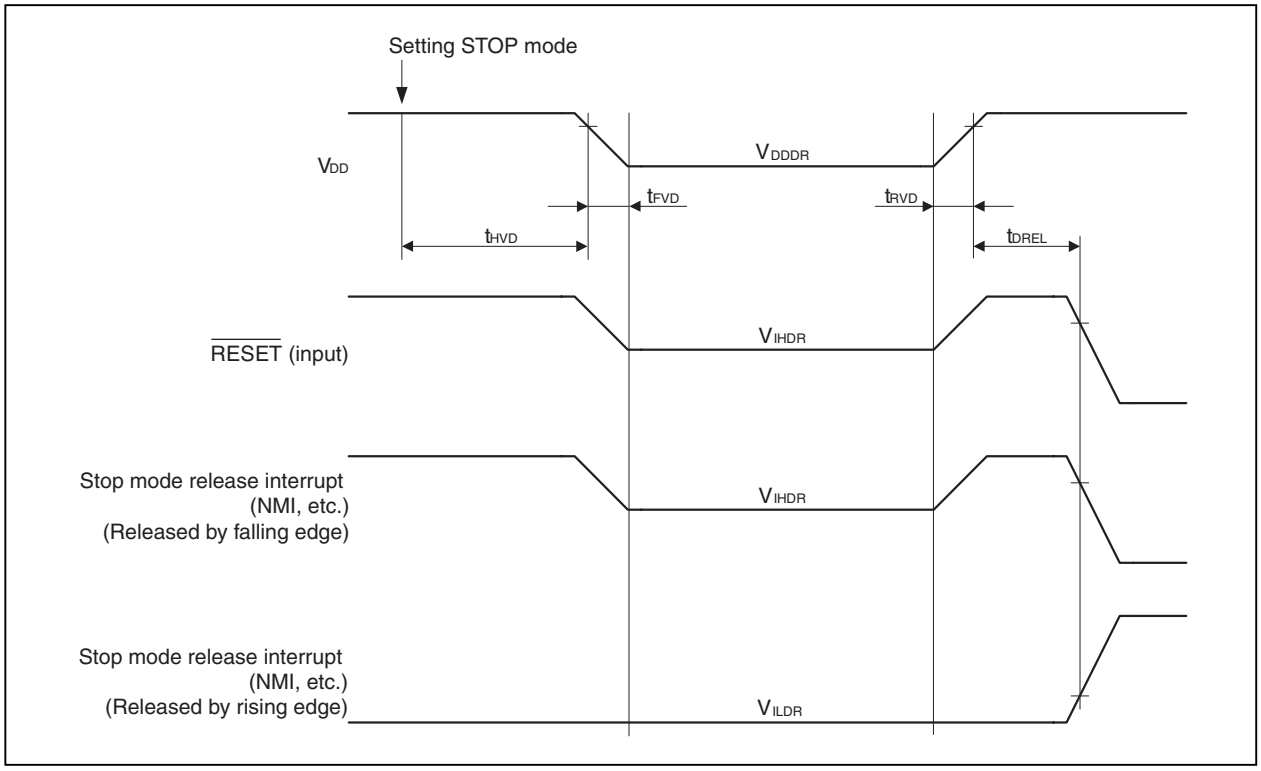
10. $f_{CPU} = f_{XT}$ = 32.768 kHz, $V_{IN} = V_{CPUREG}$, main clock oscillator stopped, all peripheral functions operating (except FCAN controller)

11. f_{XT} = 32.768 kHz, $V_{IN} = V_{CPUREG}$, main clock oscillator stopped, all peripheral functions stopped (watch timer operating)

Data Retention Characteristics (T_A = –40 to +85°C)

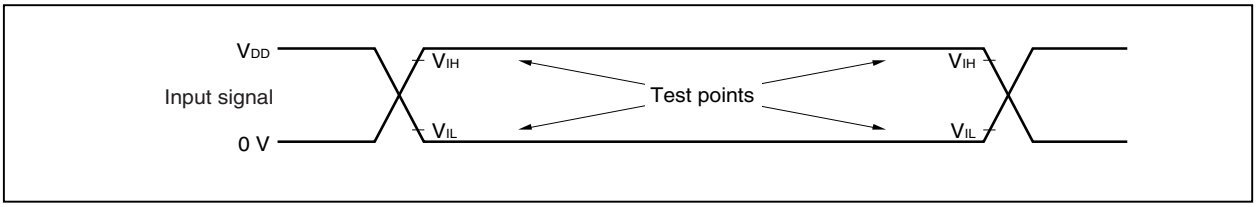
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode[Note] (no functions operating)	2.2		5.5	V
Data retention current	I_{DDDR}	STOP mode[Note] (no functions operating)		5	100	μA
Supply voltage rise time	t_{RVD}		200			μs
Supply voltage fall time	t_{FVD}		200			μs
Supply voltage hold time (from STOP mode setting)	t_{HVD}		0			ms
STOP release signal input time	t_{DREL}		0			ns
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

Note Subclock stopped

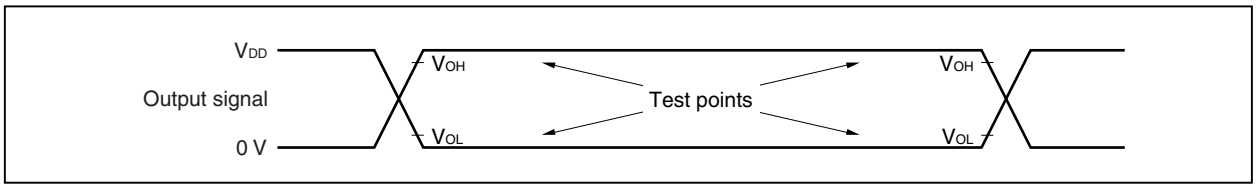


AC Characteristics

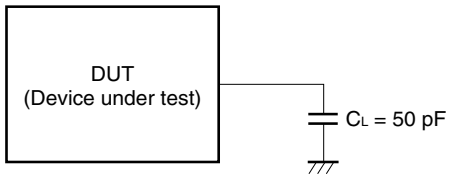
AC Test Input Test Points (V_{DD}: V_{DD}, PORTV_{DD})



AC Test Output Test Points (V_{DD}: V_{DD}, PORTV_{DD})



Load Conditions



Caution If the load capacitance exceeds 50 pF due to the circuit configuration, lower the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

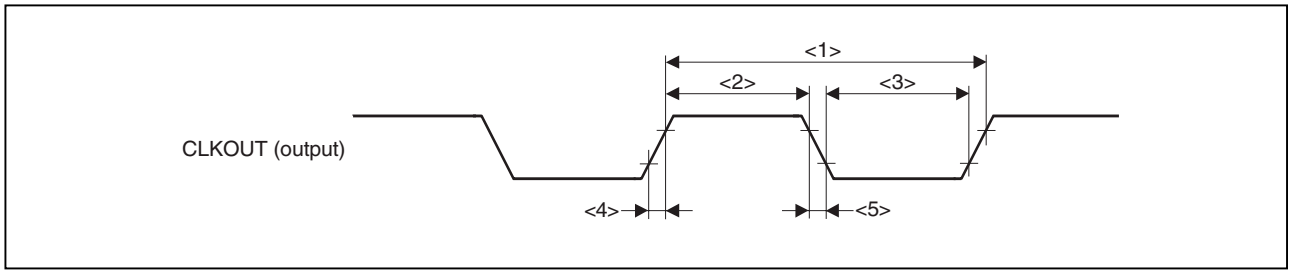
(1) Clock timing

(a) T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, $PORTV_{DD}$ = 4.0 to 5.5 V, PORTGND = 0 V

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
CLKOUT output cycle	<1>	t_{CYK}		50 ns	31 μs	
CLKOUT high-level width	<2>	t_{WKH}		$0.4(t_{CYK} - t_{KR} - t_{KF})$		ns
CLKOUT low-level width	<3>	t_{WKL}		$0.4(t_{CYK} - t_{KR} - t_{KF})$		ns
CLKOUT rise time	<4>	t_{KR}			12	ns
CLKOUT fall time	<5>	t_{KF}			12	ns

(b) T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 4.0 V, PORTGND = 0 V, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
CLKOUT output cycle	<1>	t_{CYK}		58.8 ns	31 μs	
CLKOUT high-level width	<2>	t_{WKH}		$0.4(t_{CYK} - t_{KR} - t_{KF})$		ns
CLKOUT low-level width	<3>	t_{WKL}		$0.4(t_{CYK} - t_{KR} - t_{KF})$		ns
CLKOUT rise time	<4>	t_{KR}			15	ns
CLKOUT fall time	<5>	t_{KF}			15	ns



(2) Output waveform (other than CLKOUT)

(a) T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, $PORTV_{DD}$ = 4.0 to 5.5 V, PORTGND = 0 V

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	<6>	t_{OR}			30	ns
Output fall time	<7>	t_{OF}			30	ns

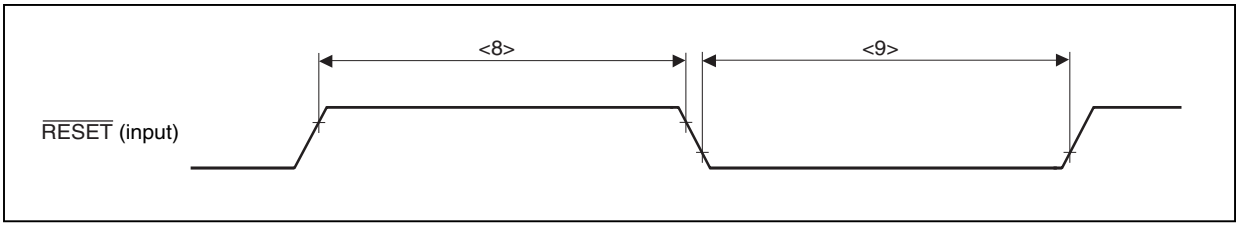
(b) T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 4.0 V, PORTGND = 0 V, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	<6>	t_{OR}			35	ns
Output fall time	<7>	t_{OF}			35	ns

(3) Reset timing

(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 5.0 V, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{RESET}$ pin high-level width	<8>	t_{WRSH}		500		ns
$\overline{RESET}$ pin low-level width	<9>	t_{WRSL}		500		ns



(4) Bus timing

(a) Clock asynchronous (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, $PORTV_{DD}$ = 4.0 to 5.5 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}		0.5T – 15		ns
Delay time from $\overline{DSTB}$↓ to address float	<12>	t_{FDA}			0	ns
Data input setup time from address	<13>	t_{SAID}			(2 + n)T – 40	ns
Data input setup time from $\overline{DSTB}$↓	<14>	t_{SDID}			(1 + n)T – 40	ns
Data input setup time from ASTB↓	<15>	t_{SASID}			(1.5 + n)T – 51	ns
Delay time from ASTB↓ to $\overline{DSTB}$↓	<16>	t_{DSTD}		0.5T – 15		ns
Data input hold time (from $\overline{DSTB}$↑)	<17>	t_{HDID}		0		ns
Address output time from $\overline{DSTB}$↑	<18>	t_{DDA}		(1 + i)T – 15		ns
Delay time from $\overline{DSTB}$↑ to ASTB↑	<19>	t_{DDST1}		0.5T – 15		ns
Delay time from $\overline{DSTB}$↑ to ASTB↓	<20>	t_{DDST2}		(1.5 + i)T – 15		ns
$\overline{DSTB}$ low-level width	<21>	t_{WDL}		(1 + n)T – 22		ns
ASTB high-level width	<22>	t_{WSTH}		T – 15		ns
Data output time from $\overline{DSTB}$↓	<23>	t_{DDOD}			10	ns
Data output setup time (to $\overline{DSTB}$↑)	<24>	t_{SODD}		(1 + n)T – 25		ns
Data output hold time (from $\overline{DSTB}$↑)	<25>	t_{HDOD}		T – 20		ns
$\overline{WAIT}$ setup time (to address)	<26>	t_{SAWT1}	n ≥ 1		1.5T – 40	ns
	<27>	t_{SAWT2}			(1.5 + n)T – 40	ns
$\overline{WAIT}$ hold time (from address)	<28>	t_{HAWT1}	n ≥ 1	(0.5 + n)T		ns
	<29>	t_{HAWT2}		(1.5 + n)T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	<30>	t_{SSTWT1}	n ≥ 1		T – 32	ns
	<31>	t_{SSTWT2}			(1 + n)T – 32	ns
$\overline{WAIT}$ hold time (from ASTB↓)	<32>	t_{HSTWT1}	n ≥ 1	nT		ns
	<33>	t_{HSTWT2}		(1 + n)T		ns
$\overline{HLDRQ}$ high-level width	<34>	t_{WHQH}		T + 10		ns
$\overline{HLDAK}$ low-level width	<35>	t_{WHAL}		T – 15		ns
Delay time from $\overline{HLDAK}$↑ to bus output	<36>	t_{DHAC}		– 11		ns
Delay time from $\overline{HLDRQ}$↓ to $\overline{HLDAK}$↓	<37>	t_{DHQHA1}		1.5T	(2n + 7.5)T + 25	ns
Delay time from $\overline{HLDRQ}$↑ to $\overline{HLDAK}$↑	<38>	t_{DHQHA2}		0.5T	1.5T + 25	ns

Remarks 1. T: $1/f_{CPU}$ (f_{CPU}: CPU 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 cycles inserted in the bus cycle.

4. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

(b) Clock asynchronous

(T_A = –40 to +85°C, PORTV_{DD} = 3.0 to 4.0 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	<10>	t_{SAST}		0.5T – 23		ns
Address hold time (from ASTB↓)	<11>	t_{HSTA}		0.5T – 22		ns
Delay time from $\overline{DSTB}$↓ to address float	<12>	t_{FDA}			0	ns
Data input setup time from address	<13>	t_{SAID}			(2 + n)T – 55	ns
Data input setup time from $\overline{DSTB}$↓	<14>	t_{SDID}			(1 + n)T – 50	ns
Data input setup time from ASTB↓	<15>	t_{SASID}			(1.5 + n)T – 65	ns
Delay time from ASTB↓ to $\overline{DSTB}$↓	<16>	t_{DSTD}		0.5T – 15		ns
Data input hold time (from $\overline{DSTB}$↑)	<17>	t_{HDID}		0		ns
Address output time from $\overline{DSTB}$↑	<18>	t_{DDA}		(1 + i)T – 15		ns
Delay time from $\overline{DSTB}$↑ to ASTB↑	<19>	t_{DDST1}		0.5T – 15		ns
Delay time from $\overline{DSTB}$↑ to ASTB↓	<20>	t_{DDST2}		(1.5 + i)T – 18		ns
$\overline{DSTB}$ low-level width	<21>	t_{WDL}		(1 + n)T – 35		ns
ASTB high-level width	<22>	t_{WSTH}		T – 18		ns
Data output time from $\overline{DSTB}$↓	<23>	t_{DDOD}			20	ns
Data output setup time (to $\overline{DSTB}$↑)	<24>	t_{SODD}		(1 + n)T – 35		ns
Data output hold time (from $\overline{DSTB}$↑)	<25>	t_{HDOD}		T – 30		ns
$\overline{WAIT}$ setup time (to address)	<26>	t_{SAWT1}	n ≥ 1		1.5T – 55	ns
	<27>	t_{SAWT2}			(1.5 + n)T – 55	ns
$\overline{WAIT}$ hold time (from address)	<28>	t_{HAWT1}	n ≥ 1	(0.5 + n)T		ns
	<29>	t_{HAWT2}		(1.5 + n)T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	<30>	t_{SSTWT1}	n ≥ 1		T – 45	ns
	<31>	t_{SSTWT2}			(1 + n)T – 45	ns
$\overline{WAIT}$ hold time (from ASTB↓)	<32>	t_{HSTWT1}	n ≥ 1	nT		ns
	<33>	t_{HSTWT2}		(1 + n)T		ns
$\overline{HLDRQ}$ high-level width	<34>	t_{WHQH}		T + 10		ns
$\overline{HLDAK}$ low-level width	<35>	t_{WHAL}		T – 25		ns
Delay time from $\overline{HLDAK}$↑ to bus output	<36>	t_{DHAC}		– 13		ns
Delay time from $\overline{HLDRQ}$↓ to $\overline{HLDAK}$↓	<37>	t_{DHQHA1}			(2n + 7.5)T + 25	ns
Delay time from $\overline{HLDRQ}$↑ to $\overline{HLDAK}$↑	<38>	t_{DHQHA2}		0.5T	1.5T + 25	ns

Remarks 1. T: 1/f_{CPU} (f_{CPU}: CPU 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 cycles inserted in the bus cycle.

4. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

(c) Clock synchronous (T_A = –40 to +85°C, V_{DD} = 4.0 to 5.5 V, $PORTV_{DD}$ = 4.0 to 5.5 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	<39>	t_{DKA}		0	25	ns
Delay time from CLKOUT↑ to address float	<40>	t_{FKA}		–12	10	ns
Delay time from CLKOUT↓ to ASTB	<41>	t_{DKST}		0	19	ns
Delay time from CLKOUT↑ to $\overline{DSTB}$	<42>	t_{DKD}		0	19	ns
Data input setup time (to CLKOUT↑)	<43>	t_{SIDK}		20		ns
Data input hold time (from CLKOUT↑)	<44>	t_{HKID}		5		ns
Delay time from CLKOUT↑ to data output	<45>	t_{DKOD}			19	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	<46>	t_{SWTK}		20		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	<47>	t_{HKWT}		5		ns
$\overline{HLDRQ}$ setup time (to CLKOUT↓)	<48>	t_{SHQK}		20		ns
$\overline{HLDRQ}$ hold time (from CLKOUT↓)	<49>	t_{HKHQ}		5		ns
Delay time from CLKOUT↑ to address float (during bus hold)	<50>	t_{DKF}			19	ns
Delay time from CLKOUT↑ to $\overline{HLDAK}$	<51>	t_{DKHA}			19	ns

Remark The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

(d) Clock synchronous

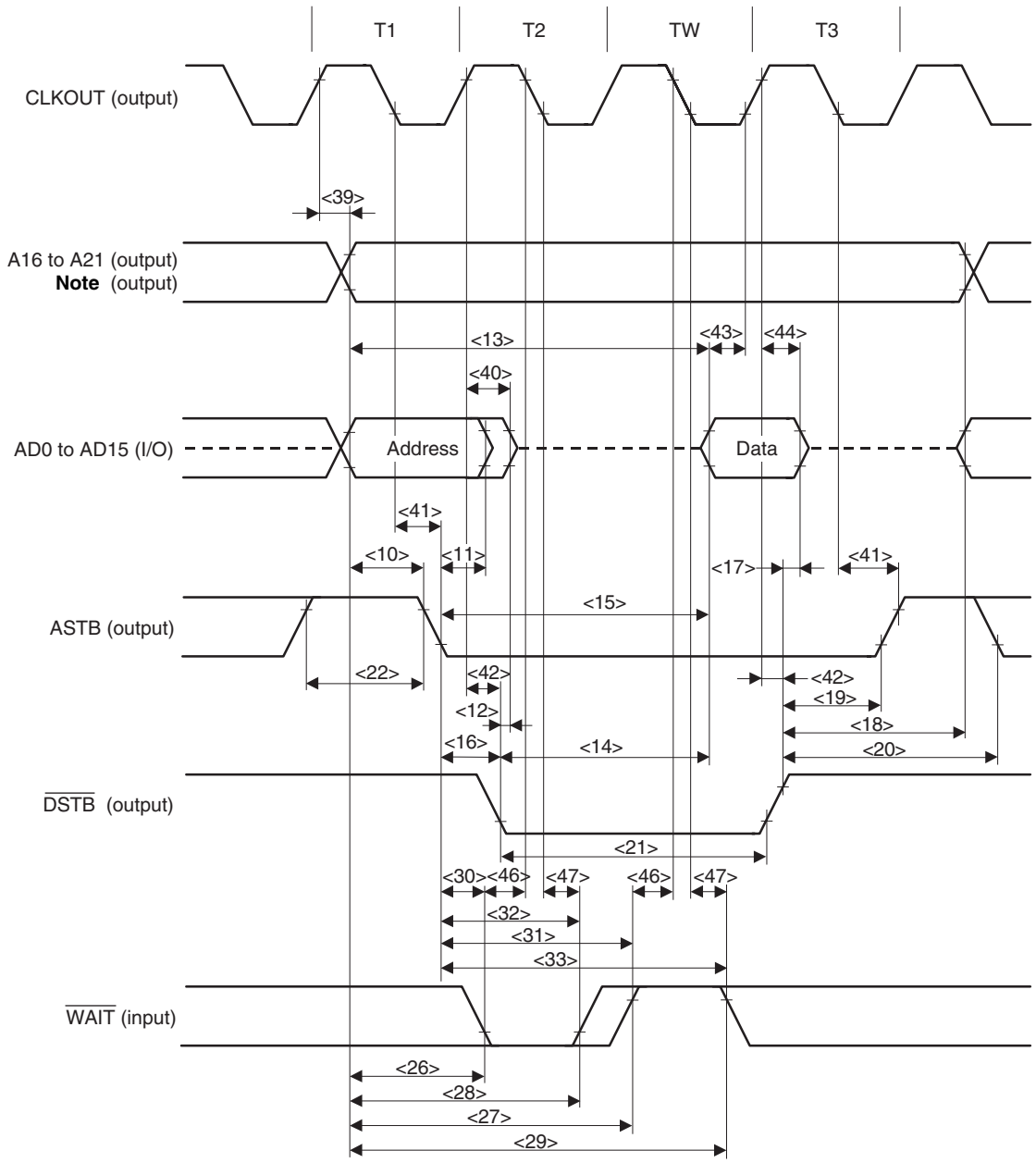
(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 4.0 V,

μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	<39>	t_{DKA}		0	40	ns
Delay time from CLKOUT↑ to address float	<40>	t_{FKA}		–16	10	ns
Delay time from CLKOUT↓ to ASTB	<41>	t_{DKST}		0	30	ns
Delay time from CLKOUT↑ to $\overline{DSTB}$	<42>	t_{DKD}		0	30	ns
Data input setup time (to CLKOUT↑)	<43>	t_{SIDK}		20		ns
Data input hold time (from CLKOUT↑)	<44>	t_{HKID}		5		ns
Delay time from CLKOUT↑ to data output	<45>	t_{DKOD}			40	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	<46>	t_{SWTK}		24		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	<47>	t_{HKWT}		5		ns
$\overline{HLDRQ}$ setup time (to CLKOUT↓)	<48>	t_{SHQK}		24		ns
$\overline{HLDRQ}$ hold time (from CLKOUT↓)	<49>	t_{HKHQ}		5		ns
Delay time from CLKOUT↑ to address float (during bus hold)	<50>	t_{DKF}			19	ns
Delay time from CLKOUT↑ to $\overline{HLDAK}$	<51>	t_{DKHA}			35	ns

Remark The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

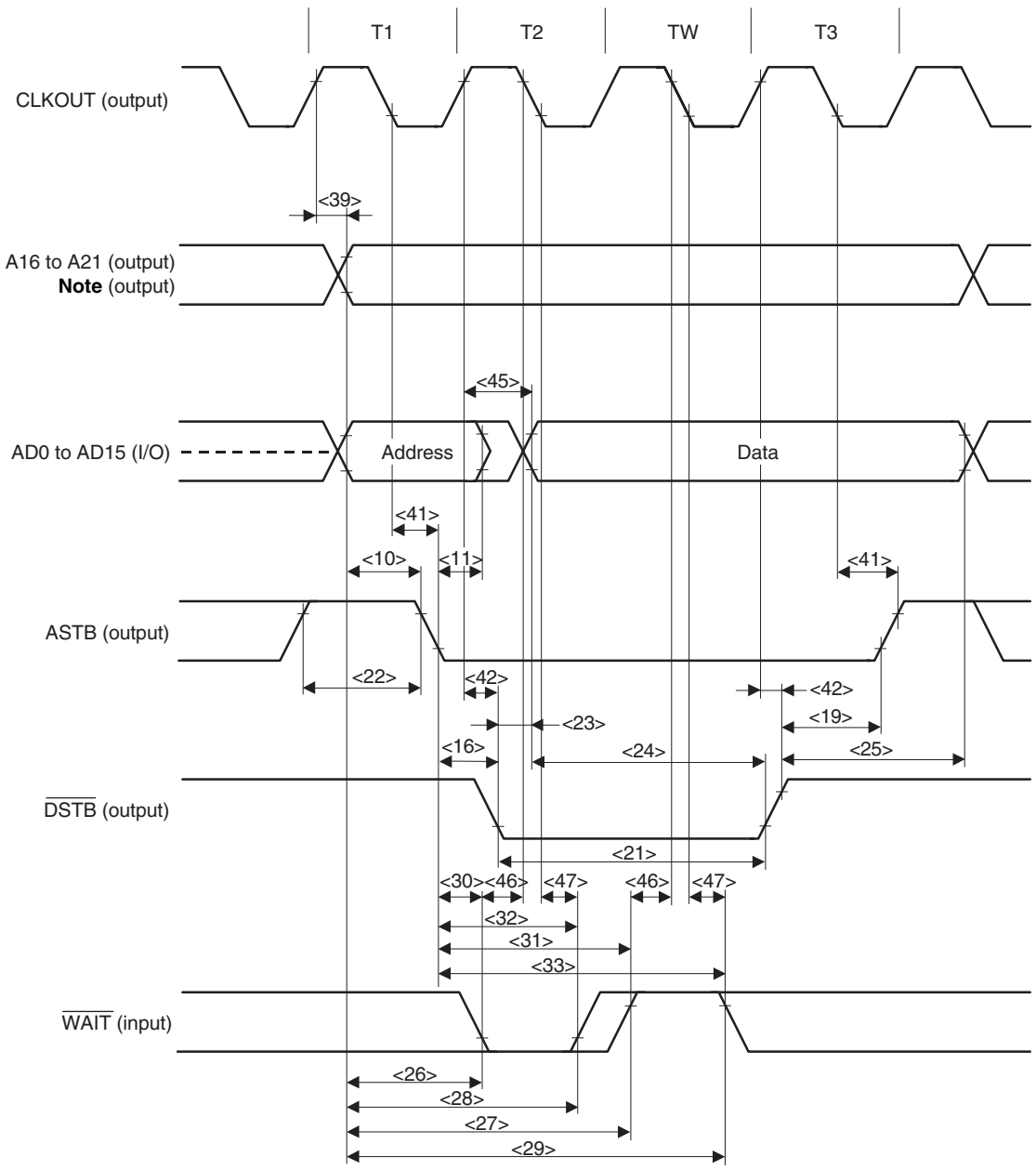
(e) Read cycle (CLKOUT synchronous/asynchronous, 1 wait)



Note R/$\overline{W}$, $\overline{UBEN}$, $\overline{LBEN}$

Remark Broken lines indicate high impedance.

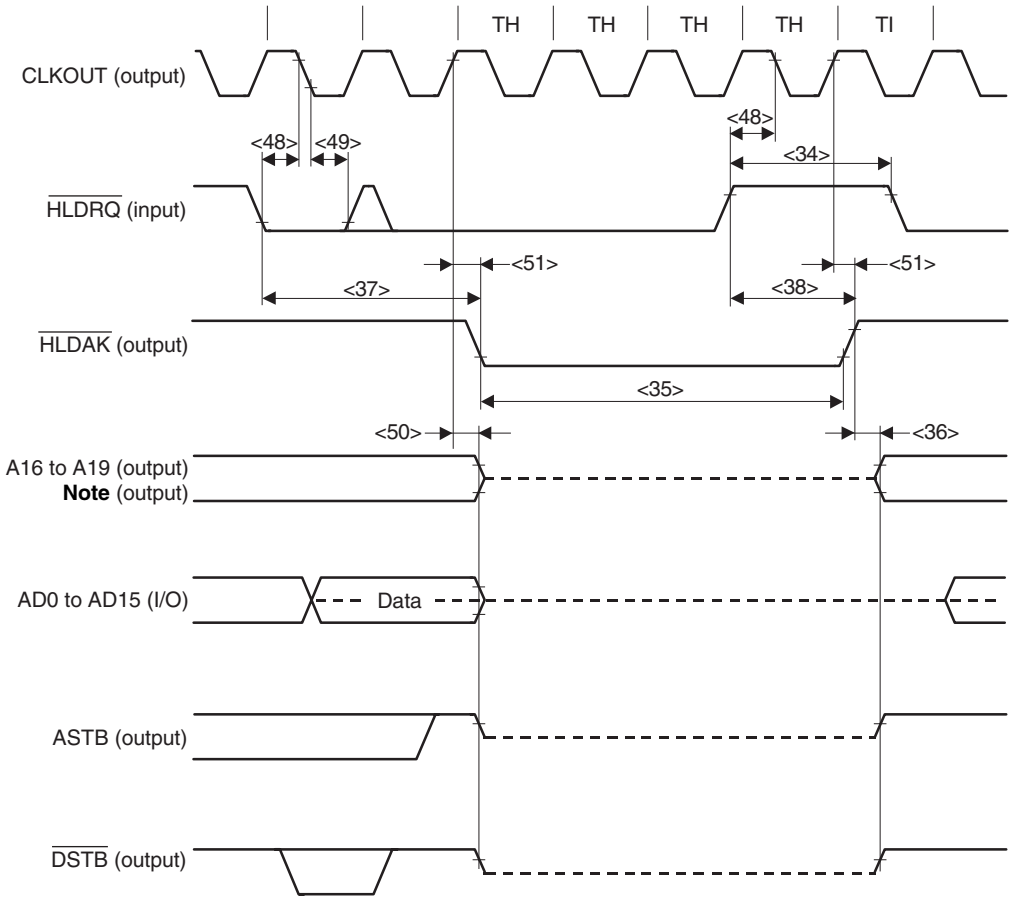
(f) Write cycle (CLKOUT synchronous/asynchronous, 1 wait)



Note R/$\overline{W}$, $\overline{UBEN}$, $\overline{LBEN}$

Remark Broken lines indicate high impedance.

(g) Bus hold timing



Note R/$\overline{W}$, $\overline{UBEN}$, $\overline{LBEN}$

Remark Broken lines indicate high impedance.

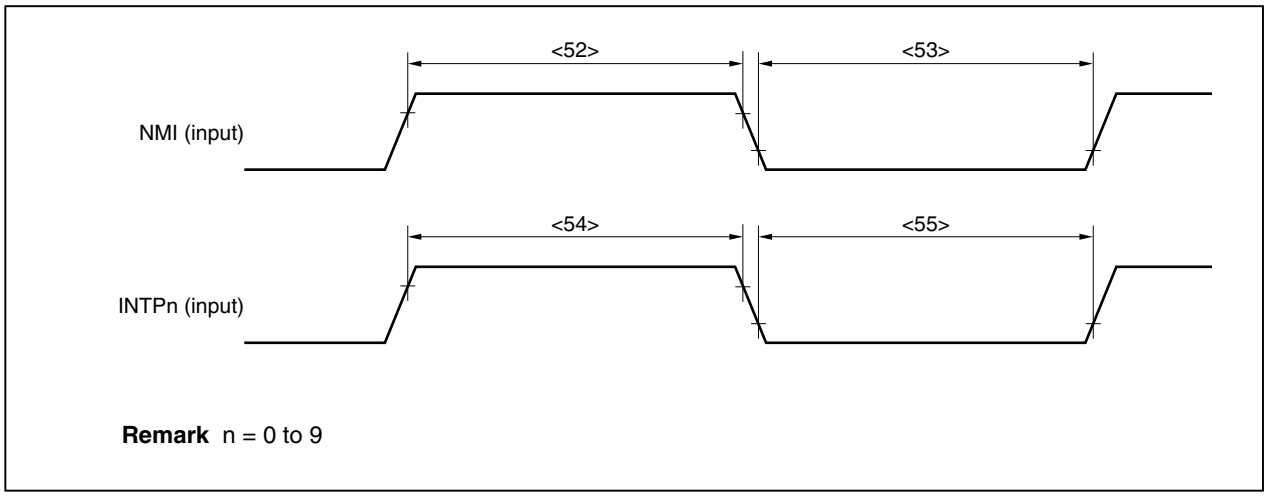
(5) Interrupt timing

(T_A = –40 to +85°C, PORTV_{DD} = 3.0 to 5.0 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	<52>	t_{WNIH}		500		ns
NMI low-level width	<53>	t_{WNIL}		500		ns
INTPn high-level width	<54>	t_{WITH}	n = 0 to 3, 7 to 9, 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	<55>	t_{WITL}	n = 0 to 3, 7 to 9, 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 9

(6) TIn timing

(T_A = –40 to +85°C, PORTV_{DD} = 3.0 to 5.5 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

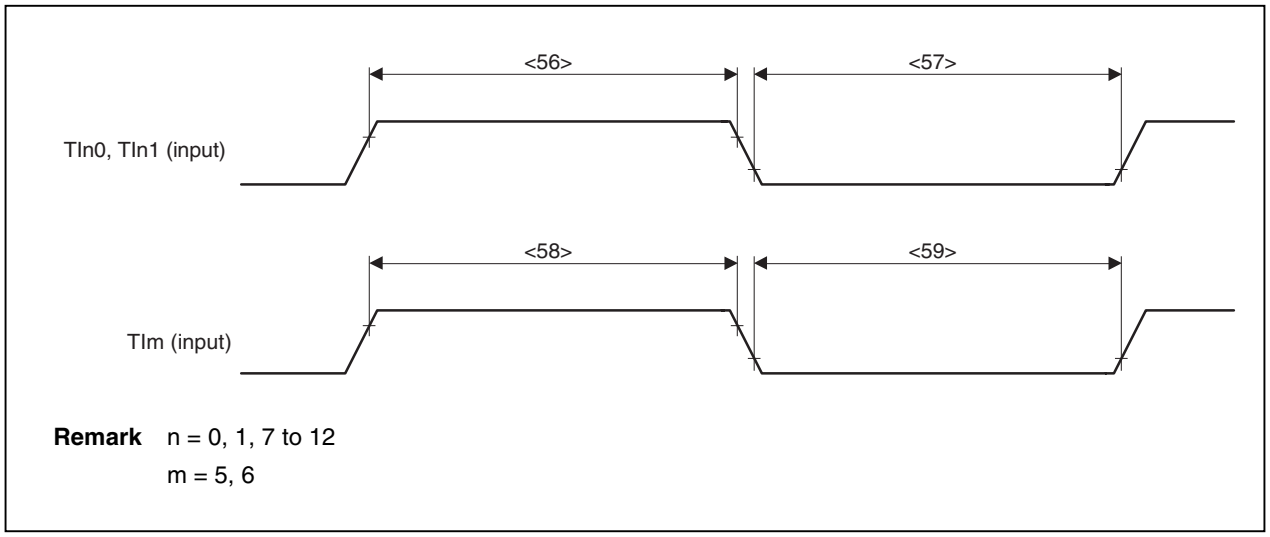
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn0, TIn1 high-level width	<56>	t_{TIHn}	n = 0, 1, 7 to 12	$2T_{sam}$ + 20[Note]		ns
TIn0, TIn1 low-level width	<57>	t_{TILn}	n = 0, 1, 7 to 12	$2T_{sam}$ + 20[Note]		ns
TIm high-level width	<58>	t_{TIHm}	m = 5, 6	3T + 20		ns
TIm low-level width	<59>	t_{TILm}	m = 5, 6	3T + 20		ns

Note The following cycles can be selected for T_{sam} (count clock cycle) by setting the PRMn2 to PRMn0 bits of prescaler mode registers n0 and n1 (PRMn0, PRMn1).

When n = 0 (TM0), T_{sam} = 2T, 4T, 16T, 64T, 256T, or 1/INTWTNI cycle
When n = 1, 7 (TM1, TM7), T_{sam} = 2T, 4T, 16T, 32T, 128T, or 256T cycle
When n = 8, 10, 12 (TM8, TM10, TM12), T_{sam} = 2T, 8T, 16T, 32T, 128T, or 256T cycle
When n = 9, 11, (TM9, TM11), T_{sam} = 4T, 8T, 32T, 64T, 128T or 512T cycle

However, when the TIn0 valid edge is selected as the count clock, T_{sam} = 4T.

Remark T: 1/fxx



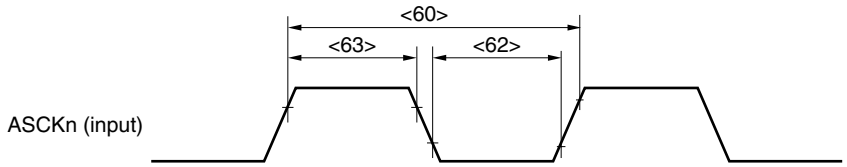
Remark n = 0, 1, 7 to 12
m = 5, 6

(7) Asynchronous serial interface (UART0 to UART3) timing

(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 5.5 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
ASCKn cycle time	<60>	t_{KCY13}		200		ns
ASCKn high-level width	<61>	t_{KH13}		80		ns
ASCKn low-level width	<62>	t_{KL13}		80		ns

Remark n = 0 to 3



Remark n = 0 to 3

(8) 3-wire serial interface (CSI0, CSI2, CSI3) timing

(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 5.5 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

(a) Master mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle	<63>	t_{KCY1}		400		ns
$\overline{SCKn}$ high-level width	<64>	t_{KH1}		140		ns
$\overline{SCKn}$ low-level width	<65>	t_{KL1}		140		ns
SIn setup time (to $\overline{SCKn}\uparrow$)	<66>	t_{SIK1}		50		ns
SIn hold time (from $\overline{SCKn}\uparrow$)	<67>	t_{KSI1}		50		ns
Delay time from $\overline{SCKn}\downarrow$ to SOn output	<68>	t_{KSO1}				ns
					60	ns

Remark n = 0, 2, 3

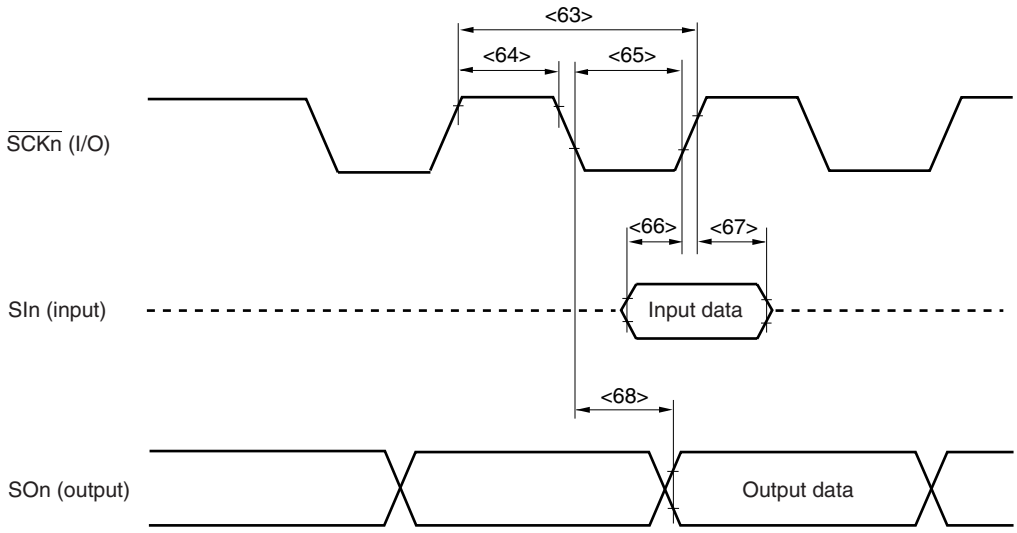
(b) Slave mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKn}$ cycle	<63>	t_{KCY2}		400		ns
$\overline{SCKn}$ high-level width	<64>	t_{KH2}		140		ns
$\overline{SCKn}$ low-level width	<65>	t_{KL2}		140		ns
SIn setup time (to $\overline{SCKn}\uparrow$)	<66>	t_{SIK2}		50		ns
SIn hold time (from $\overline{SCKn}\uparrow$)	<67>	t_{KSI2}		50		ns
Delay time from $\overline{SCKn}\downarrow$ to SOn output	<68>	t_{KSO2}	**Note 1**		80	ns
			Note 2		100	ns

Notes 1. $PORTV_{DD}$ = 4.0 to 5.5 V

2. $PORTV_{DD}$ = 3.0 to 4.0 V

Remark n = 0, 2, 3



Remarks 1. Broken lines indicate high impedance.

2. n = 0, 2, 3

(9) 3-wire variable-length serial interface (CSI4) timing

(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 5.5 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

(a) Master mode

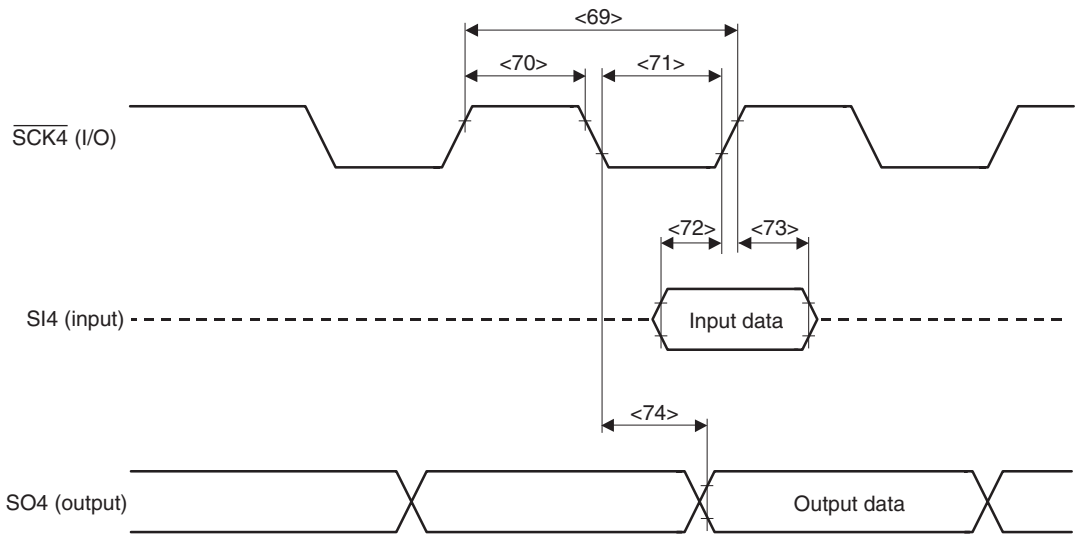
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCK4}$ cycle	<69>	t_{KCY1}	**Note 1**	200		ns
			Note 2	400		ns
$\overline{SCK4}$ high-level width	<70>	t_{KH1}	**Note 1**	60		ns
			Note 2	140		ns
$\overline{SCK4}$ low-level width	<71>	t_{KL1}	**Note 1**	60		ns
			Note 2	140		ns
SI4 setup time (to $\overline{SCK4}$↑)	<72>	t_{SIK1}	**Note 1**	25		ns
			Note 2	50		ns
SI4 hold time (from $\overline{SCK4}$↑)	<73>	t_{KSI1}		20		ns
Delay time from $\overline{SCK4}$↓ to SO4 output	<74>	t_{KSO1}			55	ns

Notes 1. $PORTV_{DD}$ = 4.0 to 5.5 V
2. $PORTV_{DD}$ = 3.0 to 4.0 V

(b) Slave mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCK4}$ cycle	<69>	t_{KCY2}	**Note 1**	200		ns
			Note 2	400		ns
$\overline{SCK4}$ high-level width	<70>	t_{KH2}	**Note 1**	60		ns
			Note 2	140		ns
$\overline{SCK4}$ low-level width	<71>	t_{KL2}	**Note 1**	60		ns
			Note 2	140		ns
SI4 setup time (to $\overline{SCK4}$↑)	<72>	t_{SIK2}	**Note 1**	25		ns
			Note 2	50		ns
SI4 hold time (from $\overline{SCK4}$↑)	<73>	t_{KSI2}		20		ns
Delay time from $\overline{SCK4}$↓ to SO4 output	<74>	t_{KSO2}	**Note 1**		70	ns
			Note 2		120	ns

Notes 1. $PORTV_{DD}$ = 4.0 to 5.5 V
2. $PORTV_{DD}$ = 3.0 to 4.0 V



Remark Broken lines indicate high impedance.

(10) 3-wire serial interface (CSI5, CSI6) timing

(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 5.5 V, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

(a) Master mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCK}$n cycle	<75>	t_{KCY1}	**Note 1**	200		ns
			Note 2	400		ns
$\overline{SCK}$n high-level width	<76>	t_{KH1}	**Note 1**	60		ns
			Note 2	140		ns
$\overline{SCK}$n low-level width	<77>	t_{KL1}	**Note 1**	60		ns
			Note 2	140		ns
SIn setup time (to $\overline{SCK}$n↑)	<78>	t_{SIK1}	**Note 1**	30		ns
			Note 2	60		ns
SIn hold time (from $\overline{SCK}$n↑)	<79>	t_{KSI1}		30		ns
Delay time from $\overline{SCK}$n↓ to SOn output	<80>	t_{KSO1}			55	ns

Notes 1. $PORTV_{DD}$ = 4.0 to 5.5 V

2. $PORTV_{DD}$ = 3.0 to 4.0 V

Remark n = 5, 6

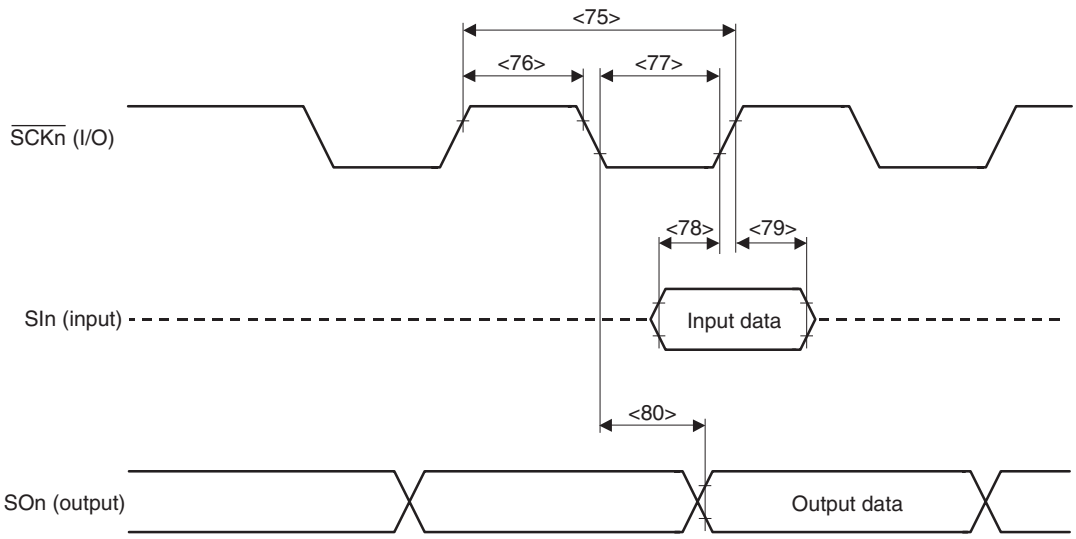
(b) Slave mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCK}$n cycle	<75>	t_{KCY2}	**Note 1**	200		ns
			Note 2	400		ns
$\overline{SCK}$n high-level width	<76>	t_{KH2}	**Note 1**	60		ns
			Note 2	140		ns
$\overline{SCK}$n low-level width	<77>	t_{KL2}	**Note 1**	60		ns
			Note 2	140		ns
SIn setup time (to $\overline{SCK}$n↑)	<78>	t_{SIK2}	**Note 1**	50		ns
			Note 2	100		ns
SIn hold time (from $\overline{SCK}$n↑)	<79>	t_{KSI2}		50		ns
Delay time from $\overline{SCK}$n↓ to SOn output	<80>	t_{KSO2}	**Note 1**		70	ns
			Note 2		120	ns

Notes 1. $PORTV_{DD}$ = 4.0 to 5.5 V

2. $PORTV_{DD}$ = 3.0 to 4.0 V

Remark n = 5, 6



Remarks 1. Broken lines indicate high impedance.

2. n = 5, 6

(11) I²C interface (I²C0, I²C1) timing

(T_A = –40 to +85°C, $PORTV_{DD}$ = 3.0 to 5.5 V,
μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 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)		<81>	t_{BUF}	4.7	–	1.3	–	μs
Hold time[Note 1]		<82>	$t_{HD:STA}$	4.0	–	0.6	–	μs
SCLn clock low-level width		<83>	t_{LOW}	4.7	–	1.3	–	μs
SCLn clock high-level width		<84>	t_{HIGH}	4.0	–	0.6	–	μs
Setup time for start/restart conditions		<85>	$t_{SU:STA}$	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	<86>	$t_{HD:DAT}$	5.0	–	–	–	μs
	I²C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		<87>	$t_{SU:DAT}$	250	–	100[Note 4]	–	ns
SDAn and SCLn signal rise time		<88>	t_R	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDAn and SCLn signal fall time		<89>	t_F	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		<90>	$t_{SU:STO}$	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		<91>	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.

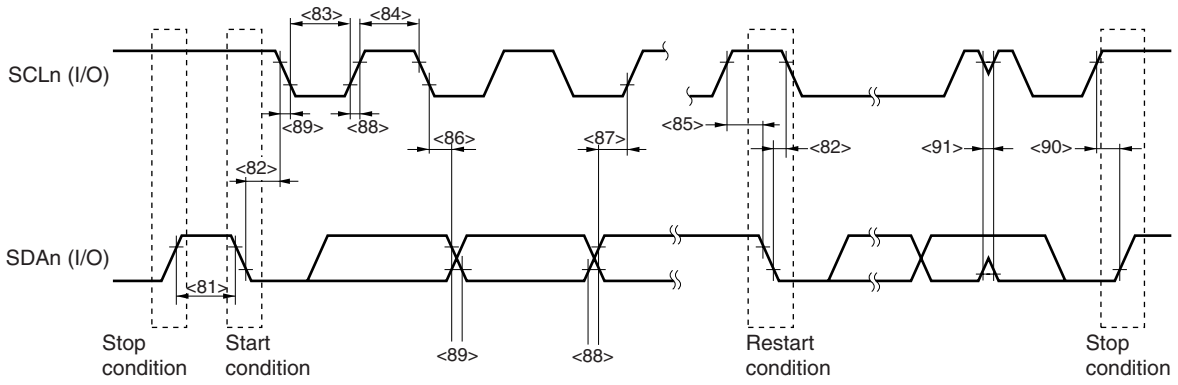
3. If the system does not extend the SCLn signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

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



Remark n = 0, 1

A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = $ADCV_{DD}$ = 4.5 to 5.5 V, GND = ADCGND = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	–		10	10	10	bit
Overall error[Note 1]	–				±1.0	%FSR
Conversion time	t_{CONV}		5		10	μs
Zero-scale error[Note 1]	AINL				±0.4	%FSR
Full-scale error[Note 1]	AINL				±0.6	%FSR
Integral linearity error[Note 2]	INL				±6.0	LSB
Differential linearity error[Note 2]	DNL				±6.0	LSB
Analog power supply voltage	AV_{DD}		4.5		5.5	V
Analog input voltage	V_{IAN}		0		$ADCV_{DD}$	V
$ADCV_{DD}$ current	AI_{DD}			4	8	mA

Notes 1. Excluding quantization error (±0.05%FSR)

2. Excluding quantization error (±0.5LSB)

Remark LSB: Least Significant Bit
FSR: Full Scale Range

Regulator, Power-on-Clear Circuit, 4.5 V Detection Flag Characteristics (T_A = –40 to +85°C, μPD703068Y, 703069Y, 703088Y, 703089Y: V_{DD} = 3.5 to 5.5 V, μPD70F3089Y: V_{DD} = 4.0 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPUREG output voltage	V_{REG}	1 μF capacitor connected	3.0	3.3	3.6	V
POC circuit detection voltage	V_{POCH}	CPU operation	2.7	3.0	3.3	V
	V_{POCL}	STOP mode	1.5	1.8	2.1	V
VM45 flag setting voltage	VM45		3.7	4.2	4.5	V

Flash Memory Programming Mode (μPD70F3089Y only)

Basic Characteristics (T_A = –20 to +85°)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply voltage	V_{DD}		4.5		5.5	V
V_{PP} supply voltage	V_{PP0}	Normal operation	0		0.6	V
	V_{PP1}	Flash memory programming	7.5	7.8	8.1	V
V_{PP} write supply current	V_{PPW}	$V_{PP} = V_{PP1}$			50	mA
V_{PP} erase supply current	I_{PPE}	$V_{PP} = V_{PP1}$			100	mA
Step erase time	t_{ER}		0.2	0.2	0.2	s
Overall erase time per area	t_{ERA}	**Note 1**			20	s/area
Number of rewrites per area	C_{ERWR}	1 erase + 1 write after erase = 1 rewrite, **Note 2**	100			Count/ area

Notes 1. The prewrite time prior to erasure and the erase verify time (write-back time) are not included.

2. When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)

Shipped product ——— P → E → P → E → P: 3 rewrites

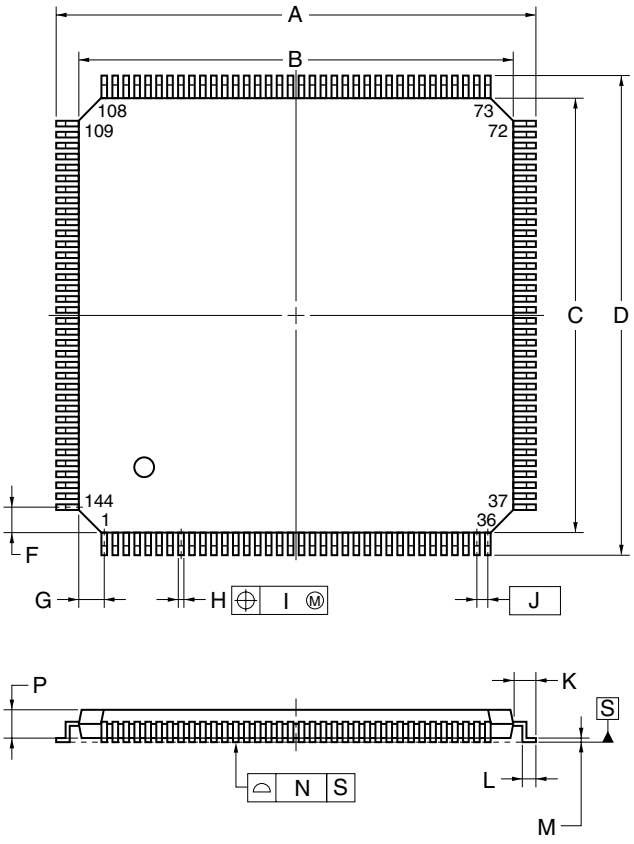
Shipped product → E → P → E → P → E → P: 3 rewrites

Remarks 1. The operating clock range during programming flash memory is the same as normal operation.

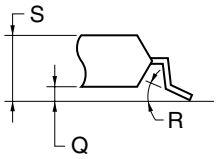
2. When the PG-FP3 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings otherwise specified.

3. Area 0 = 00000H to 1FFFFH, area 1 = 20000H to 3FFFFH, area 2 = 40000H to 5FFFFH, area 3= 60000H to 7FFFFH

144-PIN PLASTIC LQFP (FINE PITCH) (20x20)



detail of lead end



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	22.0±0.2
B	20.0±0.2
C	20.0±0.2
D	22.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5±0.2
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.4
Q	0.10±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.5±0.1

S144GJ-50-UEN

★ CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS

TBD

APPENDIX A REGISTER INDEX

(1/11)

Symbol	Name	Unit	Page
ADCR	A/D conversion result register	ADC	456
ADCRH	A/D conversion result register H (higher 8 bits)	ADC	456
ADIC	Interrupt control register	ADC	233
ADM1	A/D converter mode register 1	ADC	458
ADM2	A/D converter mode register 2	ADC	460
ADS	Analog input channel specification register	ADC	460
ASIM0	Asynchronous serial interface mode register 0	UART	437
ASIM1	Asynchronous serial interface mode register 1	UART	437
ASIM2	Asynchronous serial interface mode register 2	UART	437
ASIM3	Asynchronous serial interface mode register 3	UART	437
ASIS0	Asynchronous serial interface status register 0	UART	438
ASIS1	Asynchronous serial interface status register 1	UART	438
ASIS2	Asynchronous serial interface status register 2	UART	438
ASIS3	Asynchronous serial interface status register 3	UART	438
BCC	Bus cycle control register	BCU	206
BCR	IEBus control register	IEBus	525
BRGC0	Baud rate generator control register 0	BRG	439
BRGC1	Baud rate generator control register 1	BRG	439
BRGC2	Baud rate generator control register 2	BRG	439
BRGC3	Baud rate generator control register 3	BRG	439
BRGCK4	Baud rate generator output clock select register 4	BRG	337
BRGCN4	Baud rate generator source clock select register 4	BRG	336
BRGMC00	Baud rate generator mode control register 00	BRG	440
BRGMC01	Baud rate generator mode control register 01	BRG	440
BRGMC10	Baud rate generator mode control register 10	BRG	440
BRGMC11	Baud rate generator mode control register 11	BRG	440
BRGMC20	Baud rate generator mode control register 20	BRG	440
BRGMC21	Baud rate generator mode control register 21	BRG	440
BRGMC30	Baud rate generator mode control register 30	BRG	440
BRGMC31	Baud rate generator mode control register 31	BRG	440
C1BA	CAN1 bus active register	FCAN	621
C1BRP	CAN1 bit rate prescaler register	FCAN	622
C1CTRL	CAN1 control register	FCAN	608
C1DEF	CAN1 definition register	FCAN	612
C1DINF	CAN1 bus diagnostic information register	FCAN	625
C1ERC	CAN1 error count register	FCAN	617

(2/11)

Symbol	Name	Unit	Page
C1IE	CAN1 interrupt enable register	FCAN	618
C1INTP	CAN1 interrupt pending register	FCAN	594
C1LAST	CAN1 information register	FCAN	616
C1MASKH0	CAN1 address mask 0 register H	FCAN	606
C1MASKH1	CAN1 address mask 1 register H	FCAN	606
C1MASKH2	CAN1 address mask 2 register H	FCAN	606
C1MASKH3	CAN1 address mask 3 register H	FCAN	606
C1MASKL0	CAN1 address mask 0 register L	FCAN	606
C1MASKL1	CAN1 address mask 1 register L	FCAN	606
C1MASKL2	CAN1 address mask 2 register L	FCAN	606
C1MASKL3	CAN1 address mask 3 register L	FCAN	606
C1SYNC	CAN1 synchronization control register	FCAN	626
C2BA	CAN2 bus active register	FCAN	621
C2BRP	CAN2 bit rate prescaler register	FCAN	622
C2CTRL	CAN2 control register	FCAN	608
C2DEF	CAN2 definition register	FCAN	612
C2DINF	CAN2 bus diagnostic information register	FCAN	625
C2ERC	CAN2 error count register	FCAN	617
C2IE	CAN2 interrupt enable register	FCAN	618
C2INTP	CAN2 interrupt pending register	FCAN	594
C2LAST	CAN2 information register	FCAN	616
C2MASKH0	CAN2 address mask 0 register H	FCAN	606
C2MASKH1	CAN2 address mask 1 register H	FCAN	606
C2MASKH2	CAN2 address mask 2 register H	FCAN	606
C2MASKH3	CAN2 address mask 3 register H	FCAN	606
C2MASKL0	CAN2 address mask 0 register L	FCAN	606
C2MASKL1	CAN2 address mask 1 register L	FCAN	606
C2MASKL2	CAN2 address mask 2 register L	FCAN	606
C2MASKL3	CAN2 address mask 3 register L	FCAN	606
C2SYNC	CAN2 synchronization control register	FCAN	626
CANIC1	Interrupt control register	INTC	233
CANIC2	Interrupt control register	INTC	233
CANIC3	Interrupt control register	INTC	233
CANIC4	Interrupt control register	INTC	233
CANIC5	Interrupt control register	INTC	233
CANIC6	Interrupt control register	INTC	233
CANIC7	Interrupt control register	INTC	233
CCINTP	CAN interrupt pending register	FCAN	591
CCR	IEBus transmit counter	IEBus	543

(3/11)

Symbol	Name	Unit	Page
CDR	IEBus control data register	IEBus	528
CGCS	CAN main clock select register	FCAN	601
CGIE	CAN global interrupt enable register	FCAN	600
CGINTP	CAN global interrupt pending register	FCAN	593
CGMSR	CAN message search result register	FCAN	604
CGMSS	CAN message search start register	FCAN	604
CGST	CAN global status register	FCAN	597
CGTSC	CAN time stamp count register	FCAN	603
CORAD0	Correction address register 0	CPU	491
CORAD1	Correction address register 1	CPU	491
CORAD2	Correction address register 2	CPU	491
CORAD3	Correction address register 3	CPU	491
CORCN	Correction control register	CPU	490
CORRQ	Correction request register	CPU	490
CR00	16-bit capture/compare register 00	RPU	258
CR01	16-bit capture/compare register 01	RPU	259
CR10	16-bit capture/compare register 10	RPU	258
CR100	16-bit capture/compare register 100	RPU	258
CR101	16-bit capture/compare register 101	RPU	259
CR11	16-bit capture/compare register 11	RPU	259
CR110	16-bit capture/compare register 110	RPU	258
CR111	16-bit capture/compare register 111	RPU	259
CR120	16-bit capture/compare register 120	RPU	258
CR121	16-bit capture/compare register 121	RPU	259
CR5	16-bit compare register 5	RPU	297
CR6	16-bit compare register 6	RPU	297
CR70	16-bit capture/compare register 70	RPU	258
CR71	16-bit capture/compare register 71	RPU	259
CR80	16-bit capture/compare register 80	RPU	258
CR81	16-bit capture/compare register 81	RPU	259
CR90	16-bit capture/compare register 90	RPU	258
CR91	16-bit capture/compare register 91	RPU	259
CRC0	Capture/compare control register 0	RPU	262
CRC1	Capture/compare control register 1	RPU	262
CRC10	Capture/compare control register 10	RPU	262
CRC11	Capture/compare control register 11	RPU	262
CRC12	Capture/compare control register 12	RPU	262
CRC7	Capture/compare control register 7	RPU	262
CRC8	Capture/compare control register 8	RPU	262

(4/11)

Symbol	Name	Unit	Page
CRC9	Capture/compare control register 9	RPU	262
CSIB4	Variable-length serial setting register 4	CSI	335
CSIC0	Interrupt control register	INTC	233
CSIC2	Interrupt control register	INTC	233
CSIC3	Interrupt control register	INTC	233
CSIC4	Interrupt control register	INTC	233
CSIC5	Interrupt control register	INTC	233
CSIC6	Interrupt control register	INTC	233
CSICK5	Clocked serial interface clock select register 5	CSI	348
CSICK6	Clocked serial interface clock select register 6	CSI	348
CSIM0	Serial operation mode register 0	CSI	326
CSIM2	Serial operation mode register 2	CSI	326
CSIM3	Serial operation mode register 3	CSI	326
CSIM4	Variable-length serial control register 4	CSI	334
CSIM5	Clocked serial interface mode register 5	CSI	346
CSIM6	Clocked serial interface mode register 6	CSI	346
CSIS0	Serial clock select register 0	CSI	326
CSIS2	Serial clock select register 2	CSI	326
CSIS3	Serial clock select register 3	CSI	326
CSTOP	CAN stop register	FCAN	596
DBC0	DMA byte counter register 0	DMAC	475
DBC1	DMA byte counter register 1	DMAC	475
DBC2	DMA byte counter register 2	DMAC	475
DBC3	DMA byte counter register 3	DMAC	475
DBC4	DMA byte counter register 4	DMAC	475
DBC5	DMA byte counter register 5	DMAC	475
DCHC0	DMA channel control register 0	DMAC	476
DCHC1	DMA channel control register 1	DMAC	476
DCHC2	DMA channel control register 2	DMAC	476
DCHC3	DMA channel control register 3	DMAC	476
DCHC4	DMA channel control register 4	DMAC	476
DCHC5	DMA channel control register 5	DMAC	476
DIOA0	DMA peripheral I/O address register 0	DMAC	473
DIOA1	DMA peripheral I/O address register 1	DMAC	473
DIOA2	DMA peripheral I/O address register 2	DMAC	473
DIOA3	DMA peripheral I/O address register 3	DMAC	473
DIOA4	DMA peripheral I/O address register 4	DMAC	473
DIOA5	DMA peripheral I/O address register 5	DMAC	473

(5/11)

Symbol	Name	Unit	Page
DLR	IEBus telegraph length register	IEBus	532
DMAIC0	Interrupt control register	INTC	233
DMAIC1	Interrupt control register	INTC	233
DMAIC2	Interrupt control register	INTC	233
DMAIC3	Interrupt control register	INTC	233
DMAIC4	Interrupt control register	INTC	233
DMAIC5	Interrupt control register	INTC	233
DMAS	DMA start factor expansion register	DMAC	475
DR	IEBus data register	IEBus	533
DRA0	DMA internal RAM address register 0	DMAC	473
DRA1	DMA internal RAM address register 1	DMAC	473
DRA2	DMA internal RAM address register 2	DMAC	473
DRA3	DMA internal RAM address register 3	DMAC	473
DRA4	DMA internal RAM address register 4	DMAC	473
DRA5	DMA internal RAM address register 5	DMAC	473
DWC	Data wait control register	BCU	204
ECR	Interrupt source register	CPU	89
EGN0	Falling edge specification register 0	INTC	143, 226
EGN1	Falling edge specification register 1	INTC	155, 240
EGP0	Rising edge specification register 0	INTC	142, 226
EGP1	Rising edge specification register 1	INTC	155, 240
IEBIC1	Interrupt control register	INTC	233
IEBIC2	Interrupt control register	INTC	233
IECLK	IEBus clock select register	IEBus	543
IIC0	IIC shift register 0	I^2C	373, 387
IIC1	IIC shift register 1	I^2C	373, 387
IICC0	IIC control register 0	I^2C	375
IICC1	IIC control register 1	I^2C	375
IICCE0	IIC clock expansion register 0	I^2C	385
IICCE1	IIC clock expansion register 1	I^2C	385
IICCL0	IIC clock select register 0	I^2C	385
IICCL1	IIC clock select register 1	I^2C	385
IICF0	IIC flag register 0	I^2C	383
IICF1	IIC flag register 1	I^2C	383
IICS0	IIC status register 0	I^2C	380
IICS1	IIC status register 1	I^2C	380
IICX0	IIC function expansion register 0	I^2C	385
IICX1	IIC function expansion register 1	I^2C	385
ISPR	In-service priority register	INTC	237

(6/11)

Symbol	Name	Unit	Page
ISR	IEBus interrupt status register	IEBus	536
KRIC	Interrupt control register	INTC	233
KRM	Key return mode register	KR	253
MAM	Memory address output mode register	Port	105
MM	Memory expansion mode register	Port	104
M_CONF00 to M_CONF31	CAN message configuration registers 00 to 31	FCAN	585
M_CTRL00 to M_CTRL31	CAN message control registers 00 to 31	FCAN	577
M_DATA000 to M_DATA317	CAN message data registers 000 to 317	FCAN	581
M_DLC00 to M_DLC31	CAN message data length registers 00 to 31	FCAN	576
M_IDH00 to M_IDH31	CAN message ID registers H00 to H31	FCAN	583
M_IDL00 to M_IDL31	CAN message ID registers L00 to L31	FCAN	583
M_STAT00 to M_STAT31	CAN message status registers 00 to 31	FCAN	587
M_TIME00 to M_TIME31	CAN message time stamp registers 00 to 31	FCAN	579
NCC	Noise elimination control register	INTC	239
OSTS	Oscillation stabilization time select register	WDT	124, 318, 323
P0	Port 0	Port	140
P1	Port 1	Port	144
P10	Port 10	Port	168
P11	Port 11	Port	171
P12	Port 12	Port	176
P13	Port 13	Port	180
P14	Port 14	Port	182
P15	Port 15	Port	185
P17	Port 17	Port	188
P2	Port 2	Port	149
P3	Port 3	Port	153
P4	Port 4	Port	157

(7/11)

Symbol	Name	Unit	Page
P5	Port 5	Port	157
P6	Port 6	Port	160
P7	Port 7	Port	163
P8	Port 8	Port	163
P9	Port 9	Port	165
PAC	Port alternate-function control register	Port	173
PAC2	Port alternate-function control register 2	Port	177
PAR	IEBus partner address register	IEBus	528
PCC	Processor clock control register	CG	122
PF1	Port 1 function register	Port	145
PF2	Port 2 function register	Port	150
PIC0	Interrupt control register	INTC	233
PIC1	Interrupt control register	INTC	233
PIC2	Interrupt control register	INTC	233
PIC3	Interrupt control register	INTC	233
PIC4	Interrupt control register	INYC	233
PIC5	Interrupt control register	INTC	233
PIC6	Interrupt control register	INTC	233
PIC7	Interrupt control register	INTC	233
PM0	Port 0 mode register	Port	142
PM1	Port 1 mode register	Port	145
PM10	Port 10 mode register	Port	169
PM11	Port 11 mode register	Port	172
PM12	Port 12 mode register	Port	177
PM13	Port 13 mode register	Port	180
PM14	Port 14 mode register	Port	183
PM15	Port 15 mode register	Port	186
PM17	Port 17 mode register	Port	189
PM2	Port 2 mode register	Port	150
PM3	Port 3 mode register	Port	154
PM4	Port 4 mode register	Port	158
PM5	Port 5 mode register	Port	158
PM6	Port 6 mode register	Port	161
PM9	Port 9 mode register	Port	166
POCC	POC control register	Reset	487
POCS	POC status register	Reset	486
PRCMD	Command register	CG	119
PRM00	Prescaler mode register 00	RPU	264
PRM01	Prescaler mode register 01	RPU	264

(8/11)

Symbol	Name	Unit	Page
PRM10	Prescaler mode register 10	RPU	266
PRM100	Prescaler mode register 100	RPU	268
PRM101	Prescaler mode register 101	RPU	268
PRM11	Prescaler mode register 11	RPU	266
PRM110	Prescaler mode register 110	RPU	270
PRM111	Prescaler mode register 111	RPU	270
PRM120	Prescaler mode register 120	RPU	268
PRM121	Prescaler mode register 121	RPU	268
PRM70	Prescaler mode register 70	RPU	266
PRM71	Prescaler mode register 71	RPU	266
PRM80	Prescaler mode register 80	RPU	268
PRM81	Prescaler mode register 81	RPU	268
PRM90	Prescaler mode register 90	RPU	270
PRM91	Prescaler mode register 91	RPU	270
PSC	Power save control register	CG	123
PSW	Program status word	CPU	90
PU10	Pull-up resistor option register 10	Port	169
RXB0	Receive buffer register 0	UART	435
RXB1	Receive buffer register 1	UART	435
RXB2	Receive buffer register 2	UART	435
RXB3	Receive buffer register 3	UART	435
SAR	IEBus slave address register	IEBus	528
SCR	IEBus communication success register	IEBus	542
SC_STAT00 to SC_STAT31	CAN status set/clear registers 00 to 31	FCAN	589
SIO0	Serial I/O shift register 0	CSI	325
SIO2	Serial I/O shift register 2	CSI	325
SIO3	Serial I/O shift register 3	CSI	325
SIO4	Variable-length serial I/O shift register 4	CSI	332
SIO5	Serial I/O shift register 5	CSI	353
SIO6	Serial I/O shift register 6	CSI	353
SIOL5	Serial I/O shift register L5	CSI	353
SIOL6	Serial I/O shift register L6	CSI	353
SIRB5	Clocked serial interface receive buffer register 5	CSI	349
SIRB6	Clocked serial interface receive buffer register 6	CSI	349
SIRBE5	Clocked serial interface read-only receive buffer register 5	CSI	350
SIRBE6	Clocked serial interface read-only receive buffer register 6	CSI	350
SIRBEL5	Clocked serial interface read-only receive buffer register L5	CSI	350

(9/11)

Symbol	Name	Unit	Page
SIRBEL6	Clocked serial interface read-only receive buffer register L6	CSI	350
SIRBL5	Clocked serial interface receive buffer register L5	CSI	349
SIRBL6	Clocked serial interface receive buffer register L6	CSI	349
SOTB5	Clocked serial interface transmit buffer register 5	CSI	351
SOTB6	Clocked serial interface transmit buffer register 6	CSI	351
SOTBF5	Clocked serial interface initial transmit buffer register 5	CSI	352
SOTBF6	Clocked serial interface initial transmit buffer register 6	CSI	352
SOTBFL5	Clocked serial interface initial transmit buffer register L5	CSI	352
SOTBFL6	Clocked serial interface initial transmit buffer register L6	CSI	352
SOTBL5	Clocked serial interface transmit buffer register L5	CSI	351
SOTBL6	Clocked serial interface transmit buffer register L6	CSI	351
SRIC2	Interrupt control register	INTC	233
SRIC3	Interrupt control register	INTC	233
SSR	IEBus slave status register	IEBus	541
STIC0	Interrupt control register	INTC	233
STIC1	Interrupt control register	INTC	233
STIC2	Interrupt control register	INTC	233
STIC3	Interrupt control register	INTC	233
SVA0	Slave address register 0	I^2C	373, 387
SVA1	Slave address register 1	I^2C	373, 387
SYC	System control register	Port	201
SYS	System status register	CG	119
TCL50	Timer clock select register 50	RPU	298
TCL51	Timer clock select register 51	RPU	298
TCL60	Timer clock select register 60	RPU	298
TCL61	Timer clock select register 61	RPU	298
TM0	16-bit timer register 0	RPU	257
TM1	16-bit timer register 1	RPU	257
TM10	16-bit timer register 10	RPU	257
TM11	16-bit timer register 11	RPU	257
TM12	16-bit timer register 12	RPU	257
TM5	16-bit counter 5	RPU	297
TM6	16-bit counter 6	RPU	297
TM7	16-bit timer register 7	RPU	257
TM8	16-bit timer register 8	RPU	257
TM9	16-bit timer register 9	RPU	257
TMC0	16-bit timer mode control register 0	RPU	260
TMC1	16-bit timer mode control register 1	RPU	260
TMC10	16-bit timer mode control register 10	RPU	260

(10/11)

Symbol	Name	Unit	Page
TMC11	16-bit timer mode control register 11	RPU	260
TMC12	16-bit timer mode control register 12	RPU	260
TMC50	Timer mode control register 50	RPU	300
TMC60	Timer mode control register 60	RPU	300
TMC7	16-bit timer mode control register 7	RPU	260
TMC8	16-bit timer mode control register 8	RPU	260
TMC9	16-bit timer mode control register 9	RPU	260
TMIC00	Interrupt control register	INTC	233
TMIC01	Interrupt control register	INTC	233
TMIC10	Interrupt control register	INTC	233
TMIC100	Interrupt control register	INTC	233
TMIC101	Interrupt control register	INTC	233
TMIC11	Interrupt control register	INTC	233
TMIC110	Interrupt control register	INTC	233
TMIC111	Interrupt control register	INTC	233
TMIC120	Interrupt control register	INTC	233
TMIC121	Interrupt control register	INTC	233
TMIC5	Interrupt control register	INTC	233
TMIC6	Interrupt control register	INTC	233
TMIC70	Interrupt control register	INTC	233
TMIC71	Interrupt control register	INTC	233
TMIC80	Interrupt control register	INTC	233
TMIC81	Interrupt control register	INTC	233
TMIC90	Interrupt control register	INTC	233
TMIC91	Interrupt control register	INTC	233
TOC0	Timer output control register 0	RPU	262
TOC1	Timer output control register 1	RPU	262
TOC10	Timer output control register 10	RPU	262
TOC11	Timer output control register 11	RPU	262
TOC12	Timer output control register 12	RPU	262
TOC7	Timer output control register 7	RPU	262
TOC8	Timer output control register 8	RPU	262
TOC9	Timer output control register 9	RPU	262
TXS0	Transmit shift register 0	UART	435
TXS1	Transmit shift register 1	UART	435
TXS2	Transmit shift register 2	UART	435
TXS3	Transmit shift register 3	UART	435
UAR	IEBus unit address register	IEBus	528
USR	IEBus unit status register	IEBus	534

(11/11)

Symbol	Name	Unit	Page
VM45C	VM45 control register	Reset	487
WDCS	Watchdog timer clock select register	WDT	319
WDTIC	Interrupt control register	INTC	233
WDTM	Watchdog timer mode register	WDT	238, 320
WTNCS	Watch timer clock select register	WT	312
WTNHC	Watch timer high-speed clock select register	WT	312
WTNIC	Interrupt control register	INTC	233
WTNIIC	Interrupt control register	INTC	233
WTNM	Watch timer mode control register	WT	311

APPENDIX B LIST OF INSTRUCTION SETS

- How to read instruction set list

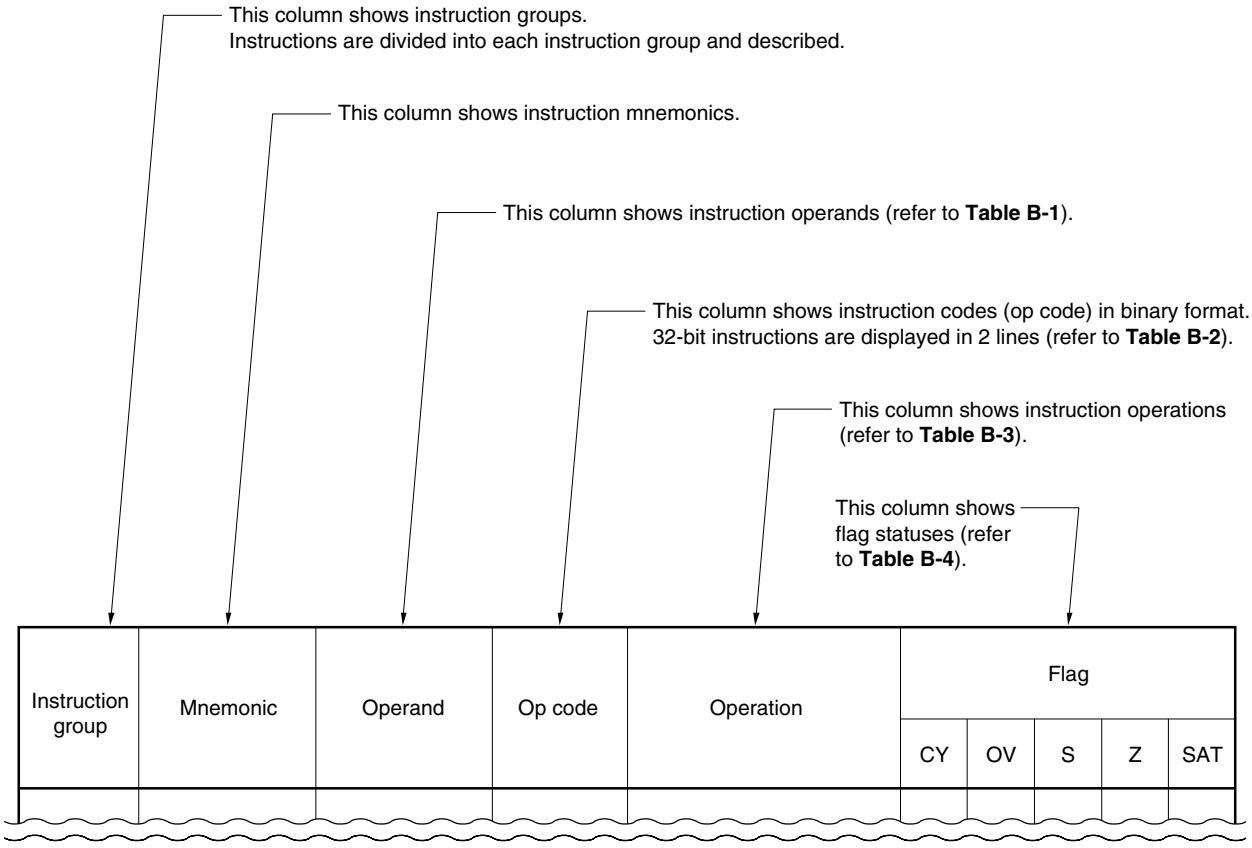


Table B-1. Symbols in Operand Description

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 B-2. Symbols Used for Op Code

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 B-3. Symbols Used for Operation Description

Symbol	Description
←	Assignment
GR[ ]	General-purpose register
SR[ ]	System register
zero-extend (n)	Zero-extends n to word length.
sign-extend (n)	Sign-extends n to word length.
load-memory (a, b)	Reads data of size b from address a.
store-memory (a, b, c)	Writes data b of size c to address a.
load-memory-bit (a, b)	Reads bit b from address a.
store-memory-bit (a, b, c)	Writes c to bit b of address a
saturated (n)	Performs saturated processing of n. (n is 2's 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 B-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 B-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	Op Code	Operation	Flag				
					CY	OV	S	Z	SAT
Load/store	SLD.B	disp7 [ep], reg2	rrrrr0110ddddddd	adr ← ep + zero-extend (disp7) GR [reg2] ← sign-extend (Load-memory (adr, Byte))					
	SLD.H	disp8 [ep], reg2	rrrrr1000ddddddd (**Note 1**)	adr ← ep + zero-extend (disp8) GR [reg2] ← sign-extend (Load-memory (adr, Halfword))					
	SLD.W	disp8 [ep], reg2	rrrrr1010ddddddd0 (**Note 2**)	adr ← ep + zero-extend (disp8) GR [reg2] ← Load-memory (adr, Word)					
	LD.B	disp16 [reg1], reg2	rrrrr111000RRRRR 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]	rrrrr1010ddddddd1 (**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	Op Code	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	Op Code	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	0000011110ddddddd dddddddddddddddd0 (**Note 1**)	PC ← PC + sign-extend (disp22)					
	JARL	disp22, reg2	rrrrr11110ddddddd dddddddddddddddd0 (**Note 1**)	GR [reg2] ← PC + 4 PC ← PC + sign-extend (disp22)					
	Bcond	disp9	ddddd1011dddcccc (**Note 2**)	if conditions are satisfied then PC ← PC + sign-extend (disp9)					
Bit manipulate	SET1	bit#3, disp16 [reg1]	00bbb111110RRRRR dddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3) Store memory-bit (adr, bit#3, 1)				×	
	CLR1	bit#3, disp16 [reg1]	10bbb111110RRRRR dddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3)) Store memory-bit (adr, bit#3, 0)				×	
	NOT1	bit#3, disp16 [reg1]	01bbb111110RRRRR dddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3)) Store-memory-bit (adr, bit#3, Z flag)				×	
	TST1	bit#3, disp16 [reg1]	11bbb111110RRRRR dddddddddddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3))				×	

Notes 1. ddddddddddddddddddddddd is the higher 21 bits of dip22.

2. dddddddd is the higher 8 bits of disp9.

Instruction Set List (4/4)

Instruction Group	Mnemonic	Operand	Op Code	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 op code of the LDSR instruction uses the field of reg1 even though the source register is shown as reg2 in the above table. Therefore, the meaning of the register specification for the mnemonic description and op code differs to that of the other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

★ APPENDIX C REVISION HISTORY

The following table shows the revision history up to the previous editions. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/5)

Edition	Major Revision from Previous Edition	Applied to:
3rd	• Deletion of indication “under development” for the following products (developed) μPD703068YGJ-×××-UEN, 703069YGJ-×××-UEN	Throughout
	• Addition of watch timer high-speed clock select register (WTNHC), IIC flag registers 0 and 1 (IICF0, IICF1)	
	Change of minimum instruction execution time in **1.4.1 Features (V850/SC3)**	**CHAPTER 1 INTRODUCTION**
	Modification of description in **Table 2-1 Pin I/O Buffer Power Supplies**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of description in **Table 2-3 Pin Operation States in Various Operating Modes**	
	Modification of **3.4.8 Peripheral I/O registers**	**CHAPTER 3 CPU FUNCTIONS**
	Addition of **Remarks** in **3.4.9 (2) System status register (SYS)**	
	Change of frequency of the V850/SC3 in **4.1 (1) Main clock oscillator**	**CHAPTER 4 CLOCK GENERATION FUNCTION**
	Addition of **Note** and **Caution** in **4.3.1 (1) Processor clock control register (PCC)**	
	Modification of description for setting DCLK1 and DCLK0 bits = 01B and addition to **Notes** in **4.3.1 (2) Power save control register (PSC)**	
	Modification of description on operation status of A16 to A21 pins in **Table 4-1 Operating Statuses in HALT Mode**	
	Modification of description on operation of UART0 to UART3 in **Table 4-2 Operating Statuses in IDLE Mode**	
	Addition of description in **4.4.4 (1) Settings and operating states**	
	Modification of description on operation status of UART0 to UART3 in **Table 4-3 Operating Statuses in Software STOP Mode**	
	Addition of **4.6 (1) When executing an instruction on internal ROM**	
	Addition of **Caution** in **4.6 (2) When executing an instruction on external ROM**	
	Modification of description in **Table 5-1 Pin I/O Buffer Power Supplies**	**CHAPTER 5 PORT FUNCTIONS**
	Addition of **Caution** in **5.2.8 (1) Function of P9 pins**	
	Addition and modification of description in **Table 5-16 Setting When Port Pin Is Used for Alternate Function**	
	Addition of **5.4 Operation of Port Function**	
	Addition of **Note** and **Caution** in **6.2.2 (1) System control register (SYC) (V850/SC1, V850/SC2)**	**CHAPTER 6 BUS CONTROL FUNCTION**
	Modification of description in **Figure 7-2 Acknowledging Non-Maskable Interrupt Requests**	**CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION**
	Addition of **7.8.1 Interrupt request valid timing following EI instruction**	
	Addition of **7.9 Bit Manipulation Instruction of Interrupt Control Register on DMA Transfer**	

(2/5)

Edition	Major Revision from Previous Edition	Applied to:
3rd	Addition and modification of description in **8.1.3 (2) Capture/compare register n0 (CR00, CR10, CR70 to CR120)**	**CHAPTER 8 TIMER/COUNTER FUNCTION**
	Addition and modification of description in **8.1.3 (3) Capture/compare register n1 (CR01, CR11, CR71 to CR121)**	
	Addition to **Cautions** in **8.1.4 (1) 16-bit timer mode control registers 0, 1, 7 to 12 (TMC0, TMC1, TMC7 to TMC12)**	
	Addition to **Cautions** in **8.1.4 (2) Capture/compare control registers 0, 1, 7 to 12 (CRC0, CRC1, CRC7 to CRC12)**	
	Addition of **Figure 8-6 Configuration of PPG Output** and **Figure 8-7 PPG Output Operation Timing**	
	Change of description of **Caution** in **8.2.6 (2) One-shot pulse output via external trigger**	
	Addition of **Caution** in **10.3 (2) Watchdog timer clock select register (WDCS)**	**CHAPTER 10 WATCHDOG TIMER FUNCTION**
	Addition of description in **11.2 (2) 3-wire serial I/O mode (fixed as MSB first)**	**CHAPTER 11 SERIAL INTERFACE FUNCTION**
	Addition to **Cautions** in **11.2.2 (1) Serial clock select register n (CSISn) and serial operation mode register n (CSIMn)**	
	Modification of description on manipulatable bits in **11.4.3 (6) Clocked serial interface read-only receive buffer registers L5, L6 (SIRBEL5, SIRBEL6)**	
	Modification of description on manipulatable bits in **11.4.3 (8) Clocked serial interface transmit buffer registers L5, L6 (SOTBL5, SOTBL6)**	
	Modification of description on manipulatable bits in **11.4.3 (10) Clocked serial interface initial transmit buffer registers L5, L6 (SOTBFL5, SOTBFL6)**	
	Modification of description on manipulatable bits in **11.4.3 (12) Serial I/O shift registers L5, L6 (SIOL5, SIOL6)**	
	Modification of description and addition to **Note** in **11.5.2 (1) IIC control register 0, 1 (IICC0, IICC1)**	
	Addition of **Caution** in **11.5.2 (4) IIC clock expansion registers 0, 1 (IICCE0, IICCE1), IIC function expansion registers 0, 1 (IICX0, IICX1), IIC clock select registers 0, 1 (IICCL0, IICCL1)**	
	Addition of **11.5.12 (2) When communication reservation function is disabled (IICRSVn of IICFn register = 1)**	
	Change of description in **11.5.13 Cautions**	
	Change of description in **11.5.14 (1) Master operations (1)**	
	Addition of **11.5.14 (2) Master operations (2)**	
	Addition of description in **Figure 11-39 Slave Operation Flowchart**	
	Addition to **Cautions** in **11.6.2 (1) Asynchronous serial interface mode registers 0 to 3 (ASIM0 to ASIM3)**	
	Addition to **Cautions** in **11.6.2 (4) Baud rate generator mode control registers n0, n1 (BRGMCn0, BRGMCn1)**	
	Addition to **Cautions** in **Figure 11-43 ASIMn Setting (Operation Stopped Mode)**	
	Addition to **Cautions** in **Figure 11-44 ASIMn Setting (Asynchronous Serial Interface Mode)**	

(3/5)

Edition	Major Revision from Previous Edition	Applied to:
3rd	Addition to **Cautions** in **Figure 11-47 BRGMCn0 and BRGMCn1 Settings (Asynchronous Serial Interface Mode)**	**CHAPTER 11 SERIAL INTERFACE FUNCTION**
	Addition of description in **11.6.3 (3) (d) Reception**	
	Deletion of description in **11.6.3 (3) (e) Receive error**	
	Modification of **Note** in **Figure 11-52 Receive Error Timing**	
	Modification of **Caution** in **12.2 (2) A/D conversion result register (ADCR), A/D conversion result register H (ADCRH)**	**CHAPTER 12 A/D CONVERTER**
	Addition of **Caution** in **12.3 (2) Analog input channel specification register (ADS)**	
	Modification of description in **12.6 (3) <3> Conflict between writing of ADCR and writing A/D converter mode register 1 (ADM1) or analog input channel specification register (ADS)**	
	Modification of description in **12.6 (8) Reading out A/D converter result register (ADCR)**	
	Addition of **13.3 Configuration**	**CHAPTER 13 DMA FUNCTIONS**
	Addition to **Cautions** in **13.4 (6) Start factor settings**	
	Addition of **13.5 Operation**	
	Addition of **13.6 Cautions**	
	Modification of description in **14.1 (3) Internal reset by power-on-clear (POC)**	**CHAPTER 14 RESET FUNCTION**
	Modification of description in **14.3 (3) POC control register (POCC)**	
	Addition of **Figure 17-1 Example of Wiring of Adapter for Flash Programming (FA-144GJ-UEN)**	**CHAPTER 17 FLASH MEMORY (μPD70F3089Y)**
	Addition of **Table 17-1 Table for Wiring of Adapter for μPD70F3089Y Flash Programming (FA-144GJ-UEN)**	
	Addition of description in **Table 18-5 Control Field Acknowledge Signal Output Conditions**	**CHAPTER 18 IEBus CONTROLLER (V850/SC2)**
	Addition of **19.1 Features**	**CHAPTER 19 FCAN CONTROLLER (V850/SC3)**
	Modification of description in **Table 19-1 Overview of Functions**	
	Change of manipulatable bits and reset values in **19.4.2 List of FCAN registers**	
	Modification of description in **19.5.1 CAN message data length registers 00 to 31 (M_DLC00 to M_DLC31)**	
	Modification of description in **19.5.2 CAN message control registers 00 to 31 (M_CTRL00 to M_CTRL31)**	
	Addition of description in **19.5.6 CAN message configuration registers 00 to 31 (M_CONF00 to M_CONF31)**	
	Modification of description in **19.5.7 CAN message status registers 00 to 31 (M_STAT00 to M_STAT31)**	
	Modification of description on manipulatable bits and modification of register format and bit description in **19.5.10 CAN global interrupt pending register (CGINTP)**	
	Modification of description on manipulatable bits and modification of register format in **19.5.11 CANn interrupt pending register (CnINTP)**	
	Addition to **Cautions** in **19.5.12 CAN stop register (CSTOP)**	
	Modification of description on manipulatable bits and modification of bit description in **19.5.13 CAN global status register (CGST)**	

(4/5)

Edition	Major Revision from Previous Edition	Applied to:
3rd	Modification of description on manipulatable bits and modification of bit description in **19.5.14 CAN global interrupt enable register (CGIE)**	**CHAPTER 19 FCAN CONTROLLER (V850/SC3)**
	Addition of description in **19.5.15 CAN main clock select register (CGCS)**	
	Deletion of **Caution** in **Figure 19-2 FCAN Clocks**	
	Addition of **Cautions** and bit name, and modification of bit description in **19.5.17 CAN message search start/result register (CGMSS/CGMSR)**	
	Addition of description in **19.5.18 CANn address mask a registers L and H (CnMASKLa and CnMASKHa)**	
	Addition of description in **19.5.19 CANn control register (CnCTRL)**	
	Modification of description on manipulatable bits and modification of bit description in **19.5.20 CANn definition register (CnDEF)**	
	Modification of description on manipulatable bits and addition of bit description in **19.5.23 CANn interrupt enable register (CnIE)**	
	Modification of description in **Cautions** and addition of bit description in **19.5.27 CANn synchronization control register (CnSYNC)**	
	Addition of **Caution** in **19.7 Time Stamp Function**	
	Modification of description in **19.8 Message Processing**	
	Change of **Figure 19-10 Composition of Layers**	
	Addition of **Caution** in **19.11.7 (2) Nominal bit time (8 to 25 time quanta)**	
	Addition to **Note** in **Figure 19-25 Nominal Bit Time**	
	Addition of description in **Figure 19-28 Initialization Processing**	
	Addition of **Note** in **Figure 19-33 Setting of CANn Synchronization Control Register (CnSYNC)**	
	Addition of description in **Figure 19-38 Message Buffer Setting**	
	Addition of **Figure 19-41 Setting of CAN Message Status Registers 00 to 31 (M_STAT00 to M_STAT31)**	
	Addition of **Figure 19-44 Setting Receive Operation Using Reception Polling**	
	Addition of **Figure 19-45 Setting of CAN Message Search Start/Result Register (CGMSS/CGMSR)**	
	Addition of description in **Figure 19-49 CAN Stop Mode Setting**	
	Addition of description in **Figure 19-50 Clearing CAN Stop Mode**	
	Modification of description in **19.13 Rules for Correct Setting of Baud Rate**	
	Addition to **Cautions** in **19.14.2 Burst read mode**	
	Deletion of **Caution 2** in **19.16 How to Shutdown FCAN Controller**	
	Addition of <4> and <5> in **19.17 Cautions on Use**	
	Addition of **CHAPTER 20 ELECTRICAL SPECIFICATIONS**	**CHAPTER 20 ELECTRICAL SPECIFICATIONS**
	Addition of **CHAPTER 21 PACKAGE DRAWING**	**CHAPTER 21 PACKAGE DRAWING**

(5/5)

Edition	Major Revision from Previous Edition	Applied to:
3rd	Addition of **CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**	**CHAPTER 22 RECOMMENDED SOLDERING CONDITIONS**
	Addition of **APPENDIX C REVISION HISTORY**	**APPENDIX C REVISION HISTORY**

NEC

Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel. FAX

Address

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑