To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

Preliminary User's Manual

V850/SV1™

32-/16-Bit Single-Chip Microcontroller

Hardware

μPD703039
μPD703039Y
μPD703040
μPD703040Y
μPD703041
μPD703041Y
μPD70F3040
μPD70F3040Y

Document No. U14462EJ1V0UMJ1 (1st edition)
Date Published August 2000 J CP(K)

[MEMO]

SUMMARY OF CONTENTS

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.

Caution This product contains an I^2C bus interface circuit.
When using the I^2C bus interface, notify its use to NEC when ordering custom code. NEC can guarantee the following only when the customer informs NEC of the use of the interface:
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 this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product 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:	μPD70F3040, 70F3040Y
The customer must judge the need for license:	μPD703039, 703039Y, 703040,703040Y, 703041, 703041Y

- **The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.**
- **Not all devices/types available in every country. Please check with local NEC representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
- NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation 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 the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
- While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
- NEC devices are classified into the following three quality grades:
  "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device 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 or medical equipment for life support, etc.

  The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

M5D 98. 12

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 Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Madrid Office
Madrid, Spain
Tel: 91-504-2787
Fax: 91-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

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 Singapore Pte. Ltd.
United Square, Singapore
Tel: 65-253-8311
Fax: 65-250-3583

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

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP Brasil
Tel: 55-11-6462-6810
Fax: 55-11-6462-6829

J00.7

INTRODUCTION

Readers This manual is intended for users who understand the functions of the V850/SV1 (μPD703039, 703039Y, 703040, 703040Y, 703041, 703041Y, 70F3040, and 70F3040Y) and design application system using the V850/SV1.

Purpose This manual is intended for users to understand the hardware functions described in the Organization below.

Organization The V850/SV1 User's Manual is divided into two parts: hardware (this manual) and architecture (V850 Family™ User's Manual Architecture).

Hardware	Architecture
• Pin function	• Data type
• CPU function	• Register set
• Internal peripheral function	• Instruction format and instruction set
• Flash memory programming	• Interrupt and exception
	• Pipeline operation

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 find out the details of a function, etc., whose name is known:

→ Refer to **APPENDIX C INDEX**.

To understand the details of a instruction function:

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

How to read register formats:

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

To understand the overall functions of the V850/SV1:

→ 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: High order at high stage and low order at low stage

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}=1,024

M (mega) : 2^{20}=$1,024^2$

G (giga) : 2^{30}=$1,024^3$

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

Related documents for V850/SV1

Document Name	Document No.
V850 Family User's Manual Architecture	U10243E
μPD703039, 703039Y, 703040, 703040Y, 703041, 703041Y Data Sheet	To be prepared
μPD70F3040, 70F3040Y Data Sheet	To be prepared
V850/SV1 User's Manual Hardware	This manual

Related documents for development tool (user's manual)

Document Name		Document No.
IE-703002-MC (In-circuit emulator)		U11595E
IE-703040-MC-EM1 (In-circuit emulator option board for the V850/SV1)		U14337E
V800 Series Development Tool (Tutorial Guide)		U14218E
CA850 (C compiler package)	Operation	U13998E
	C Language	U13997E
	Project Manager	U13996E
	Assembly Language	U13828E
ID850 (Ver. 1.31) (Integrated Debugger)	Operation (Windows™ based)	U13716E
ID850 (Ver. 2.00 or more) (Integrated Debugger)	Operation (Windows based)	U14217E
SM850 (Ver. 2.00 or more) (System Simulator)	Operation (Windows based)	U13759E
RX850 (Real Time OS)	Fundamental	U13430E
	Installation	U13410E
RX850 Pro (Real Time OS)	Fundamental	U13773E
	Installation	U13774E
RD850 (Task Debugger)[Note]		U11158E
RD850 (Ver. 3.0) (Task Debugger)		U13737E
RD850 Pro (Ver. 3.0) (Task Debugger)		U13916E
AZ850 (System Performance Analyzer)		U14410E
PG-FP3 (Flash Memory Programmer)		U13502E

Note ID850 (Ver. 1.31) supported

[MEMO]

CONTENTS

LIST OF FIGURES (1/10)

LIST OF FIGURES (2/10)

LIST OF FIGURES (3/10)

LIST OF FIGURES (4/10)

LIST OF FIGURES (5/10)

LIST OF FIGURES (6/10)

LIST OF FIGURES (7/10)

LIST OF FIGURES (8/10)

LIST OF FIGURES (9/10)

LIST OF FIGURES (10/10)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

[MEMO]

CHAPTER 1 INTRODUCTION

The V850/SV1 is a product in NEC's V850 Family of single-chip microcontrollers designed for low power operation.

1.1 General

The V850/SV1 is a 32-/16-bit single-chip microcontroller that incorporates the V850 Family CPU core, as well as peripherals such as large-size ROM/RAM, timers/counters, serial interfaces, an A/D converter, and a DMA controller.

Compared with the V850/SA1™ and V850/SB1™, the V850/SV1 offers more peripherals and pins.

In addition to having high-speed real-time response characteristics and employing single-clock basic instructions, the V850/SV1 offers multiply, saturation operation, and bit manipulation instructions, realized with a hardware multiplier, to provide digital servo control. Moreover, the V850/SV1 incorporates peripherals such as a 24-bit multifunction timer, PWM output, and a Vsync/Hsync separator that are well suited to VCR software control. The device offers extremely high cost-performance for applications such as camcorders that demand multiple functions but low power consumption.

Table 1-1 outlines the V850/SV1 product lineup.

Table 1-1. V850/SV1 Product Lineup

Product Name		ROM		RAM Size	I^2C
Popular Name	Part Number	Type	Size		
V850/SV1	μPD703039	Mask ROM	256 Kbytes	8 Kbytes	None
	μPD703040			16 Kbytes	
	μPD703041		192 Kbytes	8 Kbytes	
	μPD70F3040	Flash memory	256 Kbytes	16 Kbytes	
	μPD703039Y	Mask ROM		8 Kbytes	I^2C equipped
	μPD703040Y			16 Kbytes	
	μPD703041Y		192 Kbytes	8 Kbytes	
	μPD70F3040Y	Flash memory	256 Kbytes	16 Kbytes	

1.2 Features

- Number of instructions: 74
- Minimum instruction execution time
  - 50 ns (at 20 MHz: μPD703039, 703040, 703041, and 70F3040)
  - 59 ns (at 17 MHz: μPD703039Y, 703040Y, 703041Y, and 70F3040Y)
  - 30.5 μs (at 32.768 kHz with subsystem clock)
- General registers: 32 bits × 32 registers
- Instruction set
  - Signed multiplication (16 × 16 → 32): 100 ns (at 20 MHz)
  - (able to execute subsequent instructions in parallel without incurring any register conflicts).
  - Saturation operations (including overflow and underflow detection functions)
  - 32-bit shift instruction: 1 clock
  - Bit manipulation instructions
  - Load/store instructions in long/short format
- Memory space
  - 16 Mbytes of linear address space (for programs and data)
  - Memory block allocation function: 2 Mbytes per block
  - Programmable wait function
  - Idle state insertion function
- External bus interface
  - 16-bit multiplexed bus
  - Bus hold function
  - External wait function
- Internal memory
  - μPD703039 and 703039Y (mask ROM: 256 Kbytes/RAM: 8 Kbytes)
  - μPD703040 and 703040Y (mask ROM: 256 Kbytes/RAM: 16 Kbytes)
  - μPD703041 and 703041Y (mask ROM: 192 Kbytes/RAM: 8 Kbytes)
  - μPD70F3040 and 70F3040Y (flash memory: 256 Kbytes/RAM: 16 Kbytes)
- Interrupts and exceptions
  - Non-maskable interrupts: 2 sources
  - Maskable interrupts: 51 sources (μPD703039, 703040, 703041, and 70F3040)
    52 sources (μPD703039Y, 703040Y, 703041Y, and 70F3040Y)
  - Software exceptions: 32 sources
  - Exception trap: 1 source
- I/O lines: Total: 151 (16 input ports and 135 I/O ports)
- Timer/counter
  - 24-bit timer: 2 channels (one PWM output)
  - 16-bit timer: 2 channels (PWM outputs)
  - 8-bit timer: 8 channels (cascade connection and PWM output enabled)
- Watch timer
  - When operating under subsystem or main system clock: 1 channel
  - Operation using the subsystem or main system clock is also possible in IDLE mode.
- Watchdog timer: 1 channel

○ Serial interface (SIO)	Asynchronous serial interface (UART) Clock-synchronized serial interface (CSI) 3-wire variable length serial interface (CSI4) I^2C bus interface (I^2C) (μPD703039Y, 703040Y, 703041Y, and 70F3040Y) CSI/UART: 2 channels CSI/I^2C: 2 channels CSI (8 to 16 bits, valiable): 1 channel Dedicated baud rate generator: 3 channels (UART: 2 channels, variable length CSI: 1 channel)
○ A/D converter	10-bit resolution: 16 channels Supports select and scan modes (1-buffer/4-buffer mode selectable in select mode) ADCR buffer 10 bits × 8
○ DMA controller	Internal RAM ←→ on-chip peripheral I/O: 6 channels
○ Real-time output port	8 bits × 2 channels (Each channel can be used as one 8-bit channel or two 4-bit sub-channels.)
○ ROM correction	Modifiable 4 points
○ Key return function	4 to 8 selection enabled, falling edge fixed
○ Clock generator	During main system clock or subsystem clock operation 5-level CPU clock (including slew rate and sub operations) CPU clock output functions (CLKOUT) System clock output functions (CLO)
○ PWM output	4 channels
○ Vsync/Hsync separator	Supports both PAL and NTSC formats Field evaluation and Hsync automatic correction functions
○ Power-saving functions	HALT/IDLE/STOP modes
○ Package	176-pin plastic LQFP (fine pitch) (24 × 24 mm, 0.5 mm pitch, 1.40 mm thickness)
○ CMOS structure	Full static circuits

1.3 Application Fields

- Camcorders system/servo/camera control
- Compact cameras such as digital still cameras
- Cellular phones

1.4 Ordering Information

Part Number	Package	Internal ROM
μPD703039GM-×××-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Mask ROM
μPD703040GM-×××-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Mask ROM
μPD703041GM-×××-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Mask ROM
μPD703039YGM-×××-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Mask ROM
μPD703040YGM-×××-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Mask ROM
μPD703041YGM-×××-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Mask ROM
μPD70F3040GM-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Flash memory
μPD70F3040YGM-UEU[Note]	176-pin Plastic LQFP (fine-pitch) (24 × 24 mm)	Flash memory

Note Under development

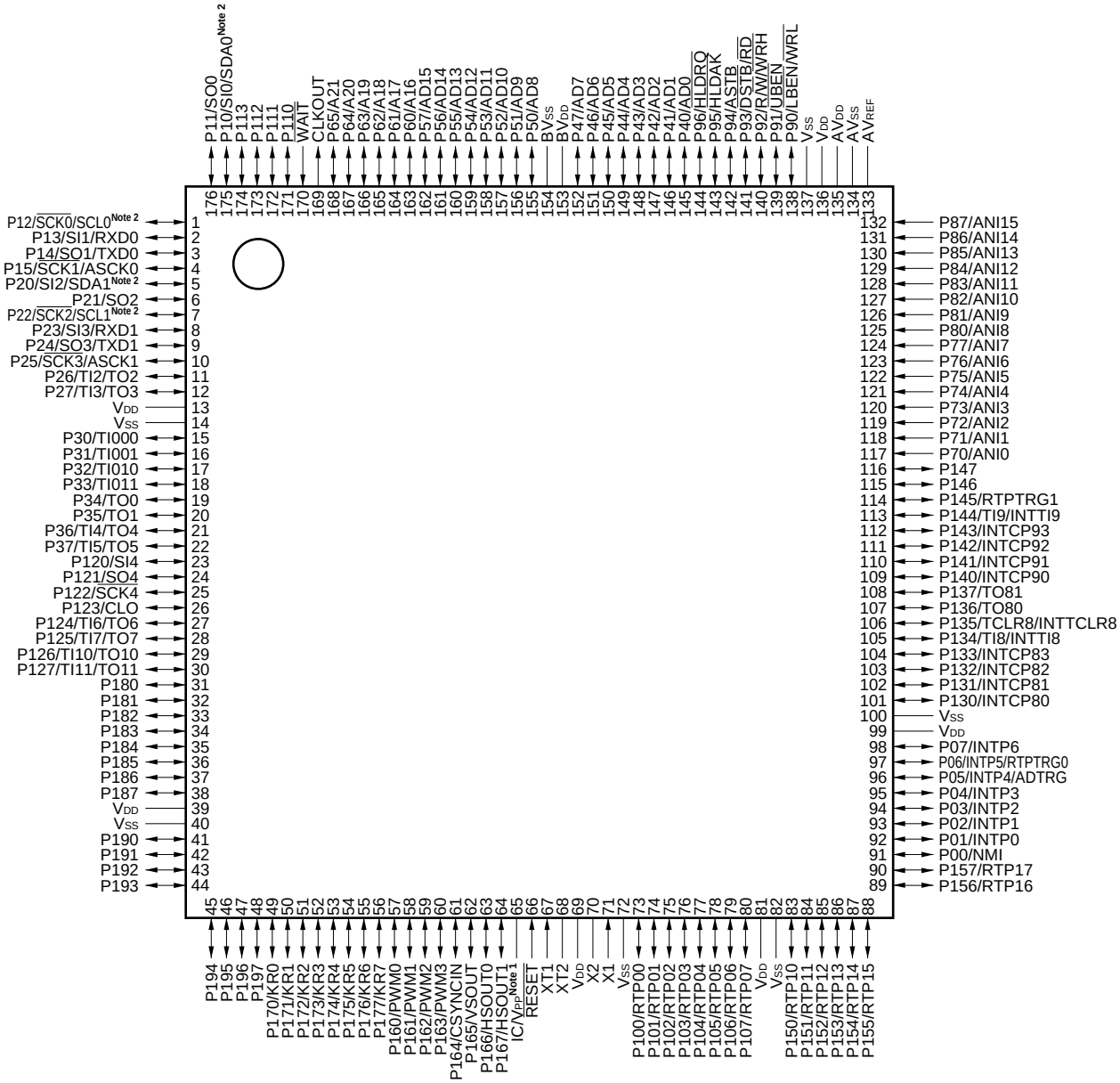
Remarks 1. ××× indicates ROM code suffix.

2. ROM-less devices are not provided.

1.5 Pin Configuration (Top View)

176-pin plastic LQFP (fine-pitch) (24 × 24 mm)

- μPD703039GM-×××-UEU
- μPD703039YGM-×××-UEU
- μPD703040GM-×××-UEU
- μPD703040YGM-×××-UEU
- μPD703041GM-×××-UEU
- μPD703041YGM-×××-UEU
- μPD70F3040GM-UEU
- μPD70F3040YGM-UEU



Notes **1.** μPD703039, 703039Y, 703040, 703040Y, 703041, 703041Y: IC (connect directly to VSS)
μPD70F3040, 70F3040Y: VPP (connect to VSS in normal operation mode)

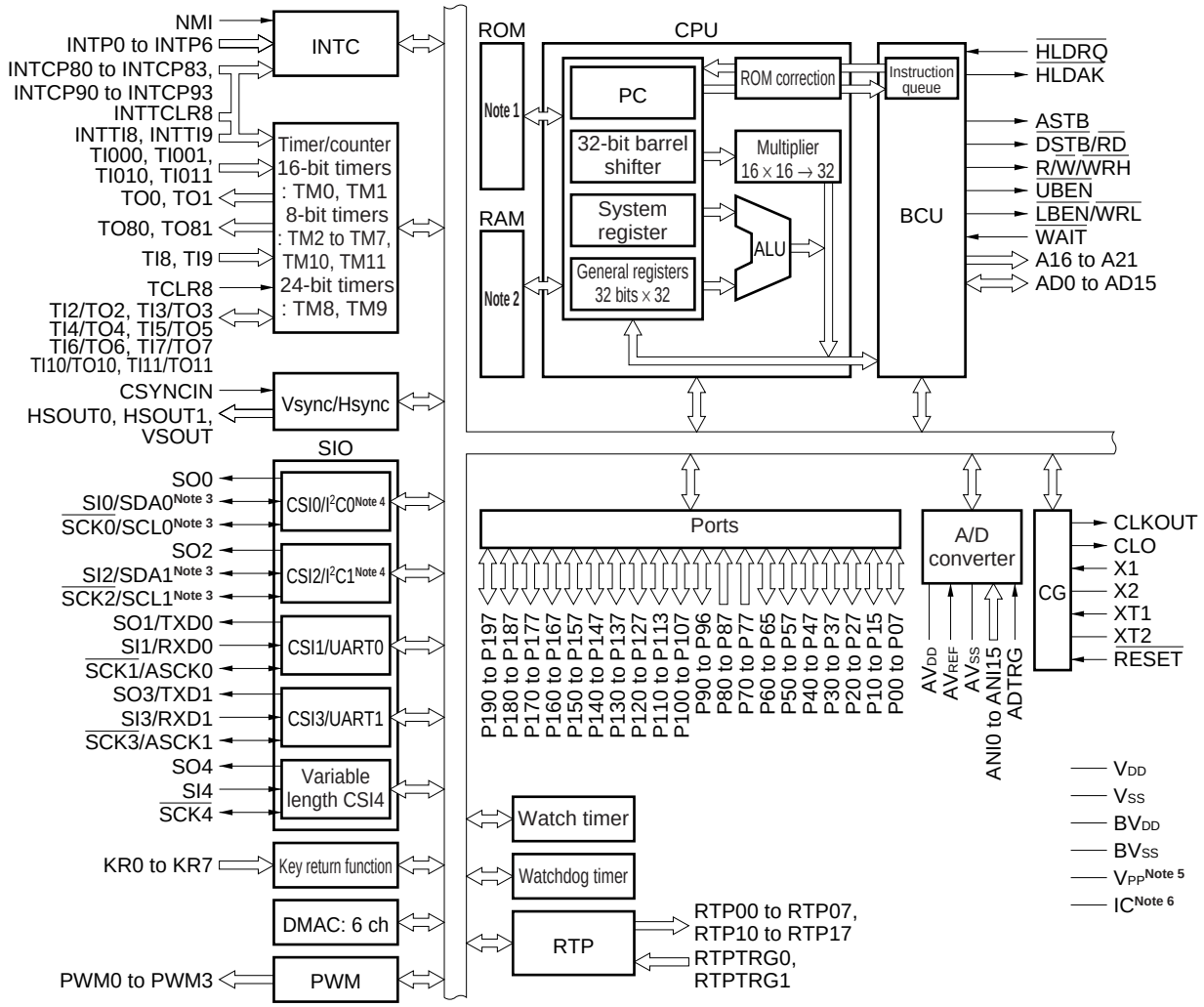
2. SCL0, SCL1, SDA0, and SDA1 are valid for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y only.

Pin identification

A16 to A21:	Address Bus	P110 to P113:	Port 11
AD0 to AD15:	Address/Data Bus	P120 to P127:	Port 12
ADTRG:	AD Trigger Input	P130 to P137:	Port 13
ANI0 to ANI15:	Analog Input	P140 to P147:	Port 14
ASCK0, ASCK1:	Asynchronous Serial Clock	P150 to P157:	Port 15
ASTB:	Address Strobe	P160 to P167:	Port 16
AV_{DD}:	Analog Power Supply	P170 to P177:	Port 17
AV_{REF}:	Analog Reference Voltage	P180 to P187:	Port 18
AV_{SS}:	Analog Ground	P190 to P197:	Port 19
BV_{DD}:	Bus Interface Power Supply	PWM0 to PWM3:	Pulse Width Modulation
BV_{SS}:	Bus Interface Ground	$\overline{RD}$:	Read
CLKOUT:	Clock Output	$\overline{RESET}$:	Reset
CLO:	Clock Output (divided)	RTP00 to RTP07,:	Real-time Output Port
CSYNCIN:	Csync Input	RTP10 to RTP17	
$\overline{DSTB}$:	Data Strobe	RTPTRG0, RTPTRG1:	RTP Trigger Input
$\overline{HLDAK}$:	Hold Acknowledge	R/$\overline{W}$:	Read/Write Status
$\overline{HLDRQ}$:	Hold Request	RXD0, RXD1:	Receive Data
HSOUT0, HSOUT1:	Hsync Output	$\overline{SCK0}$ to $\overline{SCK4}$:	Serial Clock
IC:	Internally Connected	SCL0, SCL1:	Serial Clock
INTCP80 to INTPC83,:	Interrupt Request from Peripherals	SDA0, SDA1:	Serial Data
INTCP90 to INTCP93,		SI0 to SI4:	Serial Input
INTP0 to INTP6,		SO0 to SO4:	Serial Output
INTTCLR8,		TCLR8:	Timer Clear
INTTI8, INTTI9		TI000, TI001, TI010,:	Timer Input
KR0 to KR7:	Key Return	TI011, TI2 to TI11	
$\overline{LBEN}$:	Lower Byte Enable	TO0 to TO7, TO80,:	Timer Output
NMI:	Non-Maskable Interrupt Request	TO81, TO10, TO11	
P00 to P07:	Port 0	TXD0,TXD1:	Transmit Data
P10 to P15:	Port 1	$\overline{UBEN}$:	Upper Byte Enable
P20 to P27:	Port 2	V_{DD}:	Power Supply
P30 to P37:	Port 3	V_{PP}:	Programming Power Supply
P40 to P47:	Port 4	VSOUT:	Vsync Output
P50 to P57:	Port 5	V_{SS}:	Ground
P60 to P65:	Port 6	$\overline{WAIT}$:	Wait
P70 to P77:	Port 7	$\overline{WRH}$:	Write Strobe High Level Data
P80 to P87:	Port 8	$\overline{WRL}$:	Write Strobe Low Level Data
P90 to P96:	Port 9	X1, X2:	Crystal for Main System Clock
P100 to P107:	Port 10	XT1, XT2:	Crystal for Subsystem Clock

1.6 Function Blocks

1.6.1 Internal block diagram



Notes 1. μPD703039, 703040, 703039Y, 703040Y: 256 Kbytes (Mask ROM)
μPD903041,703041Y: 192 Kbytes (Mask ROM)
μPD70F3040, 70F3040Y: 256 Kbytes (Flash memory)

2. μPD703039, 703039Y, 703041, 703041Y: 8 Kbytes
μPD703040, 703040Y, 70F3040, 70F3040Y: 16 Kbytes

3. SDA0, SDA1, SCL0, and SCL1 are valid for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y only.

4. The I^2C function is valid for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y only.

5. Applies to the μPD70F3040 and 70F3040Y.

6. Applies to the μPD703039, 703040, 703041, 703039Y, 703040Y, and 703041Y.

1.6.2 Internal units

(1) CPU

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

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

(2) Bus control unit (BCU)

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

(3) ROM

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

ROM can be accessed by the CPU in one clock cycle during instruction fetch. The table below shows the differences among the products.

Product Name	Internal ROM Capacity	Internal ROM Area
μPD703041, 703041Y	192 Kbytes (Mask ROM)	xx000000H to xx02FFFFH
μPD703039, 703039Y	256 Kbytes (Mask ROM)	xx000000H to xx03FFFFH
μPD703040, 703040Y		
μPD70F3040, 70F3040Y	256 Kbytes (Flash memory)	xx000000H to xx03FFFFH

(4) RAM

This consists of an 8-Kbyte or 16-Kbyte RAM. RAM can be accessed by the CPU in one clock cycle during data access. The table below shows the differences among the products.

Product Name	Internal ROM Capacity	Internal ROM Area
μPD703039, 703039Y	8 Kbytes	xxFFD000H to xxFFEFFH
μPD703041, 703041Y		
μPD703040, 703040Y	16 Kbytes	xxFFB000H to xxFFEFFH
μPD70F3040, 70F3040Y		

(5) Interrupt controller (INTC)

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

(6) Clock generator (CG)

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

(7) Timer/counter

A two-channel 24-bit multifunction timer/event counter, two-channel 16-bit timer/event counter, and an eight-channel 8-bit timer/event counter are equipped, enabling measurement of pulse intervals and frequency as well as programmable pulse output.

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

(8) Watch timer

This timer generates interrupts for the reference period (0.5 or 0.25 seconds) using the subsystem clock (32.768 kHz) or main system clock (16.777 MHz). The use of an internal timer enables the counting of clocks of up to 68 years.

(9) Watchdog timer

A watchdog timer is equipped to detect runaway programs, system abnormalities, etc.

It can also be used as an interval timer.

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

(10) Serial interface (SIO)

The V850/SV1 includes three kinds of serial interfaces: an asynchronous serial interface (UART0, UART1), a clock-synchronized serial interface (CSI0 to CSI3), and an 8-bit/16-bit variable serial interface (CSI4). These plus the I^2C bus interface (I^2C0, I^2C1) comprise five channels. Two of these channels are switchable between the UART and CSI and another two switchable between CSI and I^2C.

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

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

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

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

I^2C0 and I^2C1 are equipped only in the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

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

(11) A/D converter

This high-speed, high-resolution 10-bit A/D converter includes 16 analog input pins. Conversion uses the successive approximation method. Eight A/D conversion result registers (ADCR0 to ADCR7) are also included to support select mode (1-buffer/4-buffer mode) and scan mode.

(12) DMA controller

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

(13) Real-time output port (RTP)

The RTP consists of two channels of real-time output functions that transfer the previously set 8-bit data to an output latch when an external trigger signal occurs or when there is a coincidence signal in a timer compare register. Each channel can be used as one 8-bit channel or two 4-bit channels.

(14) Pulse width modulation (PWM) output circuit

The PWM output circuit has four PWM signal output channels that allow a resolution of 12 to 16 bits to be selected. When connected to a low-pass filter, the PWM output circuit can be used for D/A converter output. It is well suited to the control of an actuator such as for a motor.

(15) Ports

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

Port	I/O	Port Function	Control Function
P0	8-bit I/O	General port	NMI, external interrupt, A/D converter trigger, RTP trigger
P1	6-bit I/O		Serial interface
P2	8-bit I/O		Serial interface, timer output
P3	8-bit I/O		Timer I/O
P4	8-bit I/O		External address/data bus
P5	8-bit I/O		
P6	6-bit I/O		External address bus
P7	8-bit input		A/D converter analog input
P8	8-bit input		
P9	7-bit I/O		External bus interface control signal I/O
P10	8-bit I/O		Real-time output port
P11	4-bit I/O		–
P12	8-bit I/O		Serial interface, timer output
P13	8-bit I/O		External interrupt, timer I/O
P14	8-bit I/O		External interrupt, timer input, RTP trigger
P15	8-bit I/O		Real-time output port
P16	8-bit I/O		PWM output, Csync signal input, Vsync/Hsync signal output
P17	8-bit I/O		Key return input
P18	8-bit I/O		–
P19	8-bit I/O		–

(16) Vsync/Hsync separator

The Vsync/Hsync separator separates a composite synchronizing signal (Csync) used by the VCR into a vertical synchronizing signal (Vsync) and a horizontal synchronizing signal (Hsync).

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The names and functions of pins of V850/SV1 are listed below. These pins can be divided into port pins and non-port pins according to their functions.

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

Power Supply	Corresponded Pins	Usable Voltage Range
AV_{DD}	Port 7, port 8	2.7 V ≤ AV_{DD} ≤ 3.6 V
BV_{DD}	Port 4, port 5, port 6, port 9, CLKOUT, $\overline{WAIT}$	2.7 V ≤ BV_{DD} ≤ V_{DD}
V_{DD}	Port 0, port 1, port 2, port 3, port 10, port 11, port 12, port 13, port 14, port 15, port 16, port 17, port 18, port 19, $\overline{RESET}$	2.7 V ≤ V_{DD} ≤ 3.6 V

(1) Port pins

(1/5)

Pin Name	I/O	PULL	Function	Alternate Function
P00	I/O	Yes	Port 0 8-bit I/O port Input/output mode can be specified in 1-bit units.	NMI
P01				INTP0
P02				INTP1
P03				INTP2
P04				INTP3
P05				INTP4/ADTRG
P06				INTP5/RTPTRG0
P07				INTP6
P10	I/O	Yes	Port 1 6-bit I/O port Input/output mode can be specified in 1-bit units.	SI0/SDA0
P11				SO0
P12				$\overline{SCK0}$/SCL0
P13				SI1/RXD0
P14				SO1/TXD0
P15				$\overline{SCK1}$/ASCK0
P20	I/O	Yes	Port 2 8-bit I/O port Input/output mode can be specified in 1-bit units.	SI2/SDA1
P21				SO2
P22				$\overline{SCK2}$/SCL1
P23				SI3/RXD1
P24				SO3/TXD1
P25				$\overline{SCK3}$/ASCK1
P26				TI2/TO2
P27				TI3/TO3

Remark PULL: on-chip pull-up resistor

(2/5)

Pin Name	I/O	PULL	Function	Alternate Function
P30	I/O	Yes	Port 3 8-bit I/O port Input/output mode can be specified in 1-bit units.	TI000
P31				TI001
P32				TI010
P33				TI011
P34				TO0
P35				TO1
P36				TI4/TO4
P37				TI5/TO5
P40	I/O	No	Port 4 8-bit I/O port Input/output mode can be specified in 1-bit units.	AD0
P41				AD1
P42				AD2
P43				AD3
P44				AD4
P45				AD5
P46				AD6
P47				AD7
P50	I/O	No	Port 5 8-bit I/O port Input/output mode can be specified in 1-bit units.	AD8
P51				AD9
P52				AD10
P53				AD11
P54				AD12
P55				AD13
P56				AD14
P57				AD15
P60	I/O	No	Port 6 6-bit I/O port Input/output mode can be specified in 1-bit units.	A16
P61				A17
P62				A18
P63				A19
P64				A20
P65				A21

Remark PULL: on-chip pull-up resistor

(3/5)

Pin Name	I/O	PULL	Function	Alternate Function
P70	Input	No	Port 7 8-bit input port	ANI0
P71				ANI1
P72				ANI2
P73				ANI3
P74				ANI4
P75				ANI5
P76				ANI6
P77				ANI7
P80	Input	No	Port 8 8-bit input port	ANI8
P81				ANI9
P82				ANI10
P83				ANI11
P84				ANI12
P85				ANI13
P86				ANI14
P87				ANI15
P90	I/O	No	Port 9 7-bit I/O port Input/output mode can be specified in 1-bit units.	$\overline{\text{LBEN}}$/$\overline{\text{WRL}}$
P91				$\overline{\text{UBEN}}$
P92				R/$\overline{\text{W}}$/$\overline{\text{WRH}}$
P93				$\overline{\text{DSTB}}$/$\overline{\text{RD}}$
P94				ASTB
P95				$\overline{\text{HLDAK}}$
P96				$\overline{\text{HLDRQ}}$
P100	I/O	Yes	Port 10 8-bit I/O port Input/output mode can be specified in 1-bit units.	RTP00
P101				RTP01
P102				RTP02
P103				RTP03
P104				RTP04
P105				RTP05
P106				RTP06
P107				RTP07
P110	I/O	No	Port 11 4-bit I/O port Input/output mode can be specified in 1-bit units.	–
P111				–
P112				–
P113				–

Remark PULL: on-chip pull-up resistor

(4/5)

Pin Name	I/O	PULL	Function	Alternate Function
P120	I/O	No	Port 12 8-bit I/O port Input/output mode can be specified in 1-bit units.	SI4
P121				SO4
P122				$\overline{\text{SCK4}}$
P123				CLO
P124				TI6/TO6
P125				TI7/TO7
P126				TI10/TO10
P127				TI11/TO11
P130	I/O	No	Port 13 8-bit I/O port Input/output mode can be specified in 1-bit units.	INTCP80
P131				INTCP81
P132				INTCP82
P133				INTCP83
P134				TI8
P135				TCLR8
P136				TO80
P137				TO81
P140	I/O	No	Port 14 8-bit I/O port Input/output mode can be specified in 1-bit units.	INTCP90
P141				INTCP91
P142				INTCP92
P143				INTCP93
P144				TI9
P145				RTPTRG1
P146				–
P147				–
P150	I/O	No	Port 15 8-bit I/O port Input/output mode can be specified in 1-bit units.	RTP10
P151				RTP11
P152				RTP12
P153				RTP13
P154				RTP14
P155				RTP15
P156				RTP16
P157				RTP17

Remark PULL: on-chip pull-up resistor

(5/5)

Pin Name	I/O	PULL	Function	Alternate Function
P160	I/O	No	Port 16 8-bit I/O port Input/output mode can be specified in 1-bit units.	PWM0
P161				PWM1
P162				PWM2
P163				PWM3
P164				CSYNCIN
P165				VSOUT
P166				HSOUT0
P167				HSOUT1
P170	I/O	Yes	Port 17 8-bit I/O port Input/output mode can be specified in 1-bit units.	KR0
P171				KR1
P172				KR2
P173				KR3
P174				KR4
P175				KR5
P176				KR6
P177				KR7
P180	I/O	No	Port 18 8-bit I/O port Input/output mode can be specified in 1-bit units.	–
P181				–
P182				–
P183				–
P184				–
P185				–
P186				–
P187				–
P190	I/O	No	Port 19 8-bit I/O port Input/output mode can be specified in 1-bit units.	–
P191				–
P192				–
P193				–
P194				–
P195				–
P196				–
P197				–

Remark PULL: on-chip pull-up resistor

(2) Non-port pins

(1/4)

Pin Name	I/O	PULL	Function	Alternate Function
A16 to A21	Output	No	Address bus 16 to 21	P60 to P65
AD0 to AD7	I/O	No	Address/data multiplexed bus 0 to 15	P40 to P47
AD8 to AD15				P50 to P57
ADTRG	Input	Yes	A/D converter external trigger input	P05/INTP4
ANI0 to ANI7	Input	No	Analog input to A/D converter	P70 to P77
ANI8 to ANI15	Input	No		P80 to P87
ASCK0	Input	Yes	Baud rate clock input for UART0 and UART1	P15/$\overline{\text{SCK1}}$
ASCK1				P25/$\overline{\text{SCK3}}$
ASTB	Output	No	External address strobe signal output	P94
AV_{DD}	–	–	Positive power supply for A/D converter and ports used for alternate functions	–
AV_{REF}	Input	–	Reference voltage input for A/D converter	–
AV_{SS}	–	–	Ground potential for A/D converter and ports used for alternate functions	–
BV_{DD}	–	–	Positive power supply for bus interface and ports used for alternate functions	–
BV_{SS}	–	–	Ground potential for bus interface and ports used for alternate functions	–
CLKOUT	Output	–	Internal system clock output	–
CLO	Output	No	CLO output signal	P123
CSYNCIN	Input	No	Csync signal input	P164
$\overline{\text{DSTB}}$	Output	No	External data strobe signal output	P93/$\overline{\text{RD}}$
$\overline{\text{HLDAK}}$	Output	No	Bus hold acknowledge output	P95
$\overline{\text{HLDRQ}}$	Input	No	Bus hold request input	P96
HSOUT0	Output	No	Hsync signal output before compensation	P166
HSOUT1			Hsync signal output after compensation	P167
IC	–	–	Internal connection (connect directly to V_{SS}) (μPD703039, 703039Y, 703040, 703040Y, 703041, and 703041Y)	–
INTCP80 to INTCP83	Input	No	External capture input for CC80 to CC83	P130 to P133
INTCP90 to INTCP93	Input	No	External capture input for CP90 to CP93	P140 to P143
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/RTPTRG0
INTP6			External interrupt request input (digital noise elimination supporting remote controller)	P07

Remark PULL: on-chip pull-up resistor

(2/4)

Pin Name	I/O	PULL	Function	Alternate Function
INTTCLR8	Input	No	External interrupt request input (digital noise elimination)	P135/TCLR8
INTTI8	Input	No		P134/TI8
INTTI9				P144/TI9
KR0 to KR7	Input	Yes	Key return input	P170 to P177
$\overline{\text{LBEN}}$	Output	No	Low-order byte enable signal output for external data bus	P90/$\overline{\text{WRL}}$
NMI	Input	Yes	Non-maskable interrupt request input	P00
PWM0 to PWM3	Output	No	Output of PWM channels 0 to 3	P160 to P163
$\overline{\text{RD}}$	Output	No	Bus read strobe signal output	P93/$\overline{\text{DSTB}}$
$\overline{\text{RESET}}$	Input	–	System reset input	–
RTP00 to RTP07	Output	Yes	Real-time output port	P100 to P107
RTP10 to RTP17				P150 to P157
RTPTRG0	Input	Yes	RTP external trigger input	P06
RTPTRG1		No		P146
R/$\overline{\text{W}}$	Output	No	External read/write status output	P92/$\overline{\text{WRH}}$
RXD0	Input	Yes	Serial receive data input for UART0 and UART1	P13/SI1
RXD1				P23/SI3
$\overline{\text{SCK0}}$	I/O	Yes	Serial clock I/O for CSI0 to CSI3 (3-wire mode)	P12/SCL0
$\overline{\text{SCK1}}$				P15/ASCK0
$\overline{\text{SCK2}}$				P22/SCL1
$\overline{\text{SCK3}}$				P25/ASCK1
$\overline{\text{SCK4}}$		No	Variable length CSI4 serial clock I/O	P122
SCL0	I/O	Yes	Serial clock I/O for I^2C0 and I^2C1 (μPD703039Y, 703040Y, 703041Y, and 70F3040Y)	P12/$\overline{\text{SCK0}}$
SCL1				P22/$\overline{\text{SCK2}}$
SDA0	I/O	Yes	Serial transmit/receive data I/O for I^2C0 and I^2C1 (μPD703039Y, 703040Y, 703041Y, and 70F3040Y)	P10/SI0
SDA1				P20/SI2
SI0	Input	Yes	Serial receive data input for CSI0 to CSI3 (3-wire mode)	P10/SDA0
SI1				P13/RXD0
SI2				P20/SDA1
SI3				P23/RXD1
SI4		No	Variable length CSI4 serial receive data input (3-wire mode)	P120
SO0	Output	Yes	Serial transmit data output for CSI0 to CSI3	P11
SO1				P14/TXD0
SO2				P21
SO3				P24/TXD1
SO4		No	Variable length CSI4 serial transmit data output	P121
TCLR8	Input	No	External clear input for TM8	P135/INTTCLR8

Remark PULL: on-chip pull-up resistor

(3/4)

Pin Name	I/O	PULL	Function	Alternate Function
TI000	Input	Yes	External count clock input/external capture trigger input for TM0	P30
TI001			External capture trigger input for TM0	P31
TI010			External count clock input/external capture trigger input for TM1	P32
TI011			External capture trigger input for TM1	P33
TI2			External count clock input for TM2	P26/TO2
TI3			External count clock input for TM3	P27/TO3
TI4			External count clock input for TM4	P36/TO4/A15
TI5			External count clock input for TM5	P37/TO5
TI6		No	External count clock input for TM6	P124/TO6
TI7			External count clock input for TM7	P125/TO7
TI8			External count clock input for TM8	P134/INTTI8
TI9			External count clock input for TM9	P144/INTTI9
TI10			External count clock input for TM10	P126/TO10
TI11			External count clock input for TM11	P127/TO11
TO0	Output	Yes	Pulse signal output for TM0	P34
TO1			Pulse signal output for TM1	P35
TO2			Pulse signal output for TM2	P26/TI2
TO3			Pulse signal output for TM3	P27/TI3
TO4			Pulse signal output for TM4	P36/TI4
TO5			Pulse signal output for TM5	P37/TI5
TO6		No	Pulse signal output for TM6	P124/TI6
TO7			Pulse signal output for TM7	P125/TI7
TO80			Pulse signal output 0 for TM8	P136
TO81			Pulse signal output 1 for TM8	P137
TO10			Pulse signal output for TM10	P126/TI10
TO11			Pulse signal output for TM11	P127/TI11
TXD0	Output	Yes	Serial transmit data output for UART0 and UART1	P14/SO1
TXD1				P24/SO3
$\overline{\text{UBEN}}$	Output	No	High-order byte enable signal output for external data bus	P91
V_{DD}	–	–	Positive power supply pin	–
V_{PP}	–	–	High voltage application pin for program write/verify (μPD70F3040 and 70F3040Y)	–
VSOUT	Output	No	Vsync signal output	P165
V_{SS}	–	–	Ground potential	–
$\overline{\text{WAIT}}$	Input	No	External $\overline{\text{WAIT}}$ signal input	–
$\overline{\text{WRH}}$	Output	No	High-order byte write strobe signal output for external data bus	P92/R/$\overline{\text{W}}$
$\overline{\text{WRL}}$			Low-order byte write strobe signal output for external data bus	P90/$\overline{\text{LBEN}}$

Remark PULL: on-chip pull-up resistor

(4/4)

Pin Name	I/O	PULL	Function	Alternate Function
X1	Input	No	Resonator connection for main system clock	–
X2	–			–
XT1	Input	No	Resonator connection for subsystem clock	–
XT2	–			–

Remark PULL: on-chip pull-up resistor

2.2 Pin States

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

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

Note “L” when in clock output inhibit mode

Remark Hi-Z: High impedance
Held: State is held during previously set external bus cycle
L: Low-level output
H: High-level output
–: Input without sampling

2.3 Description of Pin Functions

(1) P00 to P07 (Port 0) ⋯ 3-state I/O

Port 0 is an 8-bit I/O port that can be set in 1-bit units for input or output.

P00 to P07 can function as I/O port pins and can also function as NMI inputs, external interrupt request inputs, external triggers for the A/D converter, and external triggers for the real-time output port. Port/control mode can be selected for each bit, and the pin's valid edge is specified by the EGP0 and EGN0 registers.

(a) Port mode

P00 to P07 can be set bit-wise as input or output according to the contents of port 0 mode register (PM0).

(b) Control mode

(i) NMI (Non-maskable Interrupt Request) ⋯ input

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

(ii) INTP0 to INTP6 (Interrupt Request from Peripherals) ⋯ input

These are external interrupt request input pins.

(iii) ADTRG (AD Trigger Input) ⋯ input

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

(iv) RTPTRG0 (Real-Time Port Trigger Input) ⋯ input

This is the real-time output port's external trigger input pin. This pin is controlled with real-time output port control register 0 (RTPC0).

(2) P10 to P15 (Port 1) ⋯ 3-state I/O

Port 1 is a 6-bit I/O port in which input and output pins can be specified in 1-bit units.

P10 to P15 can function as I/O port pins and can also operate as input or output pins for the serial interface.

Port/control mode can be selected for each bit.

P10 to P12, P14, and P15 can select normal output and N-ch open-drain output.

(a) Port mode

P10 to P15 can be set in 1-bit units as input or output pins according to the contents of port 1 mode register (PM1).

(b) Control mode

(i) SI0, SI1 (Serial Input 0, 1) ⋯ input

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

(ii) SO0, SO1 (Serial Output 0, 1) ⋯ output

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

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

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

(iv) SDA0 (Serial Data 0) ⋯ I/O

This is the serial transmit/receive data I/O pin for I^2C0 (μPD703039Y, 703040Y, 703041Y, and 70F3040Y only).

(v) SCL0 (Serial Clock 0) ⋯ I/O

This is the serial clock I/O pin for I^2C0 (μPD703039Y, 703040Y, 703041Y, and 70F3040Y only).

(vi) RXD0 (Receive Data 0) ⋯ input

This is the serial receive data input pin of UART0.

(vii) TXD0 (Transmit Data 0) ⋯ output

This is the serial transmit data output pin of UART0.

(viii) ASCK0 (Asynchronous Serial Clock 0) ⋯ input

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

(3) P20 to P27 (Port 2) ⋯ 3-state I/O

Port 2 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

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

Port/control mode can be selected for each bit.

P20 to P22, P24 and P25 can select normal output and N-ch open-drain output.

(a) Port mode

P20 to P27 can be set in 1-bit units as input or output pins according to the contents of port 2 mode register (PM2).

(b) Control mode

(i) SI2, SI3 (Serial Input 2, 3) ⋯ input

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

(ii) SO2, SO3 (Serial Output 2, 3) ⋯ output

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

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

This is the serial clock I/O pin of CSI2 and CSI3.

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

This is the serial transmit/receive data I/O pin for I^2C1 (μPD703039Y, 703040Y, 703041Y, and 70F3040Y only).

(v) SCL1 (Serial Clock 1) ... I/O

This is the serial clock I/O pin for I^2C1 (μPD703039Y, 703040Y, 703041Y, and 70F3040Y only).

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

This is the serial receive data input pin of UART1.

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

This is the serial transmit data output pin of UART1.

(viii) ASCK1 (Asynchronous Serial Clock 1) ... input

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

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

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

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

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

(4) P30 to P37 (Port 3) ⋯ 3-state I/O

Port 3 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.
P30 to P37 can function as I/O port pins and input or output pins for the timer/counter.
Port/control mode can be selected for each bit.
P31 and P32 can select normal output and N-ch open-drain output.

(a) Port mode

P30 to P37 can be set in 1-bit units as input or output pins according to the contents of the port 3 mode register (PM3).

(b) Control mode

(i) TI000, TI001, TI010, TI011, TI4, TI5 (Timer Input 000, 001, 010, 011, 4, 5) ⋯ input

These pins accept external count clock input from timer 0, timer 1, timer 4, and timer 5.

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

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

(5) P40 to P47 (Port 4) ⋯ 3-state I/O

Port 4 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.
P40 to P47 can function as I/O port pins and as a time division address/data buses (AD0 to AD7) when memory is expanded externally.
Port/control mode can be selected for each bit.
The I/O signal level uses the bus interface power supply pins BV_{DD} and BV_{SS} as a reference.

(a) Port mode

P40 to P47 can be set in 1-bit units as input or output pins according to the contents of the port 4 mode register (PM4).

(b) Control mode (external expansion mode)

P40 to P47 can be set as AD0 to AD7 according to the contents of the memory expansion register (MM).

(i) AD0 to AD7 (Address/Data 0 to 7) ⋯ 3-state I/O

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

(6) P50 to P57 (Port 5) ⋯ 3-state I/O

Port 5 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

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

Port/control mode can be selected for each bit.

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

(a) Port mode

P50 to P57 can be set bit-wise as input or output pins according to the contents of the port 5 mode register (PM5).

(b) Control mode (external expansion mode)

P50 to P57 can be set as AD8 to AD15 according to the contents of the memory expansion register (MM).

(i) AD8 to AD15 (Address/Data 8 to 15) ⋯ 3-state I/O

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

(7) P60 to P65 (Port 6) ⋯ 3-state I/O

Port 6 is a 6-bit I/O port in which input and output pins can be specified in 1-bit units.

P60 to P65 can function as I/O port pins and as address buses (A16 to A21) when memory is expanded externally.

Port/control mode can be selected for each bit.

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

(a) Port mode

P60 to P65 can be set in 1-bit units as input or output pins according to the contents of the port 6 mode register (PM6).

(b) Control mode (external expansion mode)

P60 to P65 can be set as A16 to A21 according to the contents of the memory expansion register (MM).

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

These comprise an address bus that is used for external access. These pins operate as the high-order 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 as inactive, the previous bus cycle's address is retained.

(8) P70 to P77 (Port 7), P80 to P87 (Port 8) ⋯ input

Port 7 and port 8 are 8-bit input-only ports.

P70 to P77 and P80 to P87 can function as input ports and as analog input pins for the A/D converter in control mode. However, they cannot be switched between these input port and analog input pin.

(a) Port mode

P70 to P77 and P80 to P87 are input-only pins.

(b) Control mode

P70 to P77 also function as pins ANI0 to ANI7 and P80 to P87 also function as ANI8 to ANI15, but these alternate functions are not switchable.

(i) ANI0 to ANI15 (Analog Input 0 to 15) ⋯ input

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

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

(9) P90 to P96 (Port 9) ⋯ 3-state I/O

Port 9 is a 7-bit I/O port in which input and output pins can be specified in 1-bit units.

P90 to P96 can function as I/O port pins, 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-order bit is ignored during a write operation and is read as a “0” during a read operation.

Port/control can be selected for each bit.

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

(a) Port mode

P90 to P96 can be set in 1-bit units as input or output pins according to the contents of the port 9 mode register (PM9).

(b) Control mode (external expansion mode)

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

(i) $\overline{\text{LBEN}}$ (Lower Byte Enable) ⋯ output

This is a lower byte enable signal output pin for an external 16-bit data bus. The output changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. When the timing sets the bus cycle as inactive, the previous bus cycle’s address is retained.

(ii) $\overline{\text{UBEN}}$ (Upper Byte Enable) ⋯ output

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

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

(iii) R/$\overline{\text{W}}$ (Read/Write Status) ⋯ output

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

(iv) $\overline{\text{DSTB}}$ (Data Strobe) ⋯ output

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

(v) ASTB (Address Strobe) ⋯ output

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

(vi) $\overline{\text{HLDAK}}$ (Hold Acknowledge) ⋯ output

This is an output pin for the acknowledge signal that indicates high impedance status for the address bus, data bus, and control bus when the V850/SV1 receives a bus hold request.

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

(vii) $\overline{\text{HLDRQ}}$ (Hold Request) ⋯ input

This is an input pin by which an external device requests the V850/SV1 to release the address bus, data bus, and control bus. This pin accepts asynchronous input for CLKOUT. When this pin is active, the address bus, data bus, and control bus are set to high impedance status. This occurs either when the V850/SV1 completes execution of the current bus cycle or immediately if no bus cycle is being executed, then the $\overline{\text{HLDAK}}$ signal is set as active and the bus is released.

(viii) $\overline{\text{WRL}}$ (Write Strobe Low Level Data) ⋯ output

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

(ix) $\overline{\text{WRH}}$ (Write Strobe High Level Data) ⋯ output

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

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

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

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

Port 10 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P100 to P107 can function as I/O port pins and a real-time output port.

P100 to P107 can select normal output and N-ch open-drain output.

(a) Port mode

P100 to P107 can be set bit-wise as input or output pins according to the contents of the port 10 mode register (PM10).

(b) Control mode

(i) RTP00 to RTP07 (Real-time Output Ports 00 to 07) ⋯ output

These pins comprise a real-time output port.

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

Port 11 is a 4-bit I/O port in which input and output pins can be specified in 1-bit units.

P110 to P113 can function as I/O port pins only. P110 to P113 can be set bit-wise as inputs or outputs according to the contents of port 11 mode register (PM11).

(12) P120 to P127 (Port 12) ⋯ I/O

Port 12 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P120 to P127 can function as I/O port pins, input or output pins for serial interfaces, output pins for the clock signals, and input or output pins for the timer/counter. Port/control mode can be selected for each bit.

P121 and P122 can select normal output and N-ch open-drain output.

(a) Port mode

P120 to P127 can be set bit-wise as input or output pins according to the contents of the port 12 mode register (PM12).

(b) Control mode

(i) SI4 (Serial Input 4) ⋯ input

This is the serial receive data input pin for CSI4.

(ii) SO4 (Serial Output 4) ⋯ output

This is the serial transmit data output pin for CSI4.

(iii) $\overline{\text{SCK4}}$ (Serial Clock 4) ⋯ 3-state I/O

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

(iv) CLO (Clock Output (Divided)) … output

This is the output pin for the system clock (dividable).

(v) TI6, TI7, TI10, TI11 (Timer Input 6, 7, 10, 11) … input

These pins accept external count clock inputs from timer 2, timer 3, timer 10, and timer 11.

(vi) TO6, TO7, TO10, TO11 (Timer Output 6, 7, 10, 11) …output

These are the pulse signal output pins of timer 6, timer 7, timer 10, and timer 11.

(13) P130 to P137 (Port 13) … I/O

Port 13 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P130 to P137 can function as I/O port pins, input or output pins for the timer/counter, and input pins for the external interrupt requests. Port/control mode can be selected for each bit. The pin's valid edge is specified by the EGP2 and EGN2 registers.

(a) Port mode

P130 to P137 can be set bit-wise as input or output pins according to the contents of the port 13 mode register (PM13).

(b) Control mode

(i) INTCP80 to INTCP83 (Interrupt Request from Peripherals) … input

These are external interrupt request input pins.

(ii) INTTI8 (Interrupt Request from Peripherals) … input

This is an external interrupt request input pin.

(iii) INTTCLR8 (Interrupt Request from Peripherals) … input

This is an external interrupt request input pin.

(iv) TCLR8 (Interrupt Request from Peripherals) … input

This is an external interrupt request input pin.

(v) TI8 (Timer Input 8) … input

This pin accepts an external count clock input from timer 8.

(vi) TO80, TO81 (Timer Output 80, 81) … output

These are the pulse signal output pins of timer 8.

(14) P140 to P147 (Port 14) ⋯ I/O

Port 14 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P140 to P147 can function as I/O port pins, input pins for the timer/counter, input pins for external interrupt requests, and external triggers for the real-time output port. Port/control mode can be selected for each bit. The pin's valid edge is specified by the EGP3 and EGN3 registers.

(a) Port mode

P140 to P147 can be set bit-wise as input or output pins according to the contents of the port 14 mode register (PM14).

(b) Control mode

(i) INTCP90 to INTCP93 (Interrupt Request from Peripherals) ⋯ input

These are external interrupt request input pins.

(ii) INTTI9 (Interrupt Request from Peripherals) ⋯ input

This is an external interrupt request input pin.

(iii) TI9 (Timer Input 9) ⋯ input

This pin accepts an external count clock input from timer 9.

(iv) RTPTRG1 (Real-time Output Port Trigger Input 1) ⋯ input

This is the real-time output port's external trigger input pin. This pin is controlled with real-time output port control register 1 (RTPC1).

(15) P150 to P157 (Port 15) ⋯ I/O

Port 15 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P150 to P157 can function as I/O port pins and real-time output port. Port/control mode can be selected for each bit.

(a) Port mode

P150 to P157 can be set bit-wise as input or output pins according to the contents of the port 15 mode register (PM15).

(b) Control mode

(i) RTP10 to RTP17 (Real-time Output Port 10-17) ⋯ output

These pins are real-time output ports.

(16) P160 to P167 (Port 16) ⋯ I/O

Port 16 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P160 to P167 can function as I/O port pins, output pins for the PWM, and input or output pins for the Vsync/Hsync separator. Port/control mode can be selected for each bit.

(a) Port mode

P160 to P167 can be set bit-wise as input or output pins according to the contents of the port 16 mode register (PM16).

(b) Control mode

(i) PWM0 to PWM3 (Pulse Width Modulation 0 to 3) ⋯ output

These are PWM pulse signal output pins.

(ii) CSYNCIN (Csync Input) ⋯ input

This pin accepts a composite synchronizing signal (Csync signal).

(iii) VSOUT (Vsync Output) ⋯ output

This pin outputs a vertical synchronizing signal (Vsync signal).

(iv) HSOUT0, HSOUT1 (Hsync Output 0, 1) ⋯ output

These pins output a horizontal synchronizing signal (Hsync signal).

(17) P170 to P177 (Port 17) ⋯ I/O

Port 17 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P170 to P177 can function as I/O ports and a key return input.

(a) Port mode

P170 to P177 can be set bit-wise as input or output pins according to the contents of the port 17 mode register (PM17).

(b) Control mode

(i) KR0 to KR7 (Key Return 0 to 7) ⋯ input

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

(18) P180 to P187 (Port 18) ⋯ I/O

Port 18 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P180 to P187 can function only as I/O port pins.

(19) P190 to P197 (Port 19) ⋯ I/O

Port 19 is an 8-bit I/O port in which input and output pins can be specified in 1-bit units.

P190 to P197 can function only as I/O port pins.

(20) $\overline{\text{RESET}}$ (Reset) ⋯ input

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

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

(21) $\overline{\text{WAIT}}$ (Wait) ⋯ input

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

Caution When using the wait function, set BV_{DD} to the same potential as V_{DD}.

(22) X1, X2 (Crystal)

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

(23) XT1, XT2 (Crystal for Sub-clock)

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

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

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

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

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

(26) AV_{REF} (Analog Reference Voltage) … input

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

(27) BV_{DD} (Power Supply for Bus Interface)

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

(28) BV_{SS} (Ground for Bus Interface)

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

(29) V_{DD} (Power Supply)

These are the positive power supply pins. All V_{DD} pins should be connected to a positive power source.

(30) V_{SS} (Ground)

These are the ground pins. All V_{SS} pins should be grounded.

(31) V_{PP} (Programming Power Supply)

This is the positive power supply pin used for flash memory programming mode. Connect to V_{SS} in normal operating mode.

This pin is used in the μPD70F3040 and 70F3040Y.

(32) IC (Internally Connected)

This is an internally connected pin used in the μPD703039, 703040, 703041, 703039Y, 703040Y, and 703041Y. Connect directly to V_{SS} in normal operating mode.

2.4 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 Method
P00	NMI	5-W	V_{DD}	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
P01 to P04	INTP0 to INTP3			
P05	INTP4/ADTRG			
P06	INTP5/RTPTRG0			
P07	INTP6			
P10	SI0/SDA0	10-F	V_{DD}	
P11	SO0	10-E		
P12	$\overline{SCK0}$/SCL0	10-F		
P13	SI1/RXD0	5-W		
P14	SO1/TXD0	10-E		
P15	$\overline{SCK1}$/ASCK0	10-F		
P20	SI2/SDA1	10-F	V_{DD}	
P21	SO2	10-E		
P22	$\overline{SCK2}$/SCL1	10-F		
P23	SI3/RXD1	5-W		
P24	SO3/TXD1	10-E		
P25	$\overline{SCK3}$/ASCK1	10-F		
P26, P27	TI2/TO2, TI3/TO3	5-W		
P30, P31	TI000, TI001	5-W	V_{DD}	
P32, P33	TI010, TI011			
P34, P35	TO0, TO1	5-A		
P36	TI4/TO4	5-W		
P37	TI5/TO5			
P40 to P47	AD0 to AD7	5	BV_{DD}	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
P50 to P57	AD8 to AD15	5	BV_{DD}	
P60 to P65	A16 to A21	5	BV_{DD}	
P70 to P77	ANI0 to ANI7	9	AV_{DD}	Connect to AV_{SS}
P80 to P87	ANI8 to ANI15	9	AV_{DD}	
P90	$\overline{LBEN}$/$\overline{WRL}$	5	BV_{DD}	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor Output: Leave open
P91	$\overline{UBEN}$			
P92	R/$\overline{W}$/$\overline{WRH}$			
P93	$\overline{DSTB}$/$\overline{RD}$			
P94	ASTB			
P95	$\overline{HLDAK}$			
P96	$\overline{HLDRQ}$			

(2/3)

Pin	Alternate Function	I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection Method
P100 to P107	RTP00 to RTP07	10-E	V_{DD}	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
P110 to P113	–	5	V_{DD}	
P120	SI4	5-K	V_{DD}	
P121	SO4	10-G	V_{DD}	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
P122	$\overline{SCK4}$	10-H		
P123	CLO	5		
P124	TI6/TO6	5-K		
P125	TI7/TO7			
P126	TI10/TO10			
P127	TI11/TO11			
P130 to P133	INTCP80 to INTCP83	5-K	V_{DD}	
P134	TI8/INTTI8			
P135	TCLR8/INTTCLR8			
P136, P137	TO80, TO81	5		
P140 to P143	INTCP90 to INTCP93	5-K	V_{DD}	
P144	TI9/INTTI9			
P145	RTPTRG1			
P146, P147	–	5		
P150 to P157	RTP10 to RTP17	5	V_{DD}	
P160 to P163	PWM0 to PWM3	5	V_{DD}	
P164	CSYNCIN	5-K		
P165	VSOUT	5		
P166	HSOUT0			
P167	HSOUT1			
P170 to P177	KR0 to KR7	5-K	V_{DD}	
P180 to P187	–	5	V_{DD}	
P190 to P197	–	5	V_{DD}	
CLKOUT	–	4	BV_{DD}	Leave open
$\overline{WAIT}$	–	1	BV_{DD}	Connect to V_{DD} via a resistor
$\overline{RESET}$	–	2	V_{DD}	–
X1	–	–	V_{DD}	–
X2	–	–	V_{DD}	Leave open
XT1	–	–	V_{DD}	Connect to V_{SS}
XT2	–	–	V_{DD}	Leave open
AV_{REF}	–	–	–	Connect to AV_{SS}

(3/3)

Pin	Alternate Function	I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection Method
V_{PP}[Note 1]	–	–	–	Connect to V_{SS}
IC[Note 2]	–	–	–	Connect directly to V_{SS}
V_{DD}	–	–	–	–
V_{SS}	–	–	–	–
AV_{DD}	–	–	–	Connect to V_{DD}
AV_{SS}	–	–	–	Connect to V_{SS}
BV_{DD}	–	–	–	Connect to V_{DD}
BV_{SS}	–	–	–	Connect to V_{SS}

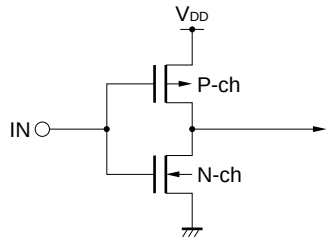
Notes 1. Applies to the μPD70F3040 and 70F3040Y.

2. Applies to the μPD70309, 703039Y, 703040, 703040Y, 703041, and 703041Y.

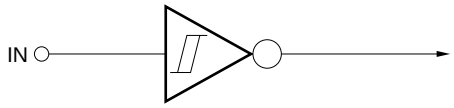
2.5 I/O Circuit of Pins

(1/2)

Type 1

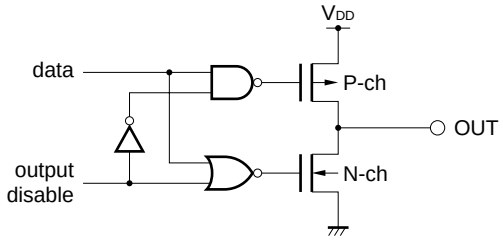


Type 2



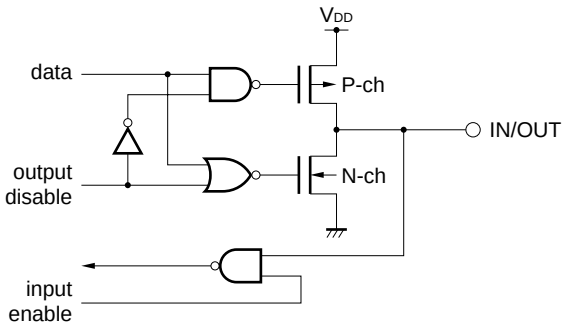
Schmitt-triggered input with hysteresis characteristics

Type 4

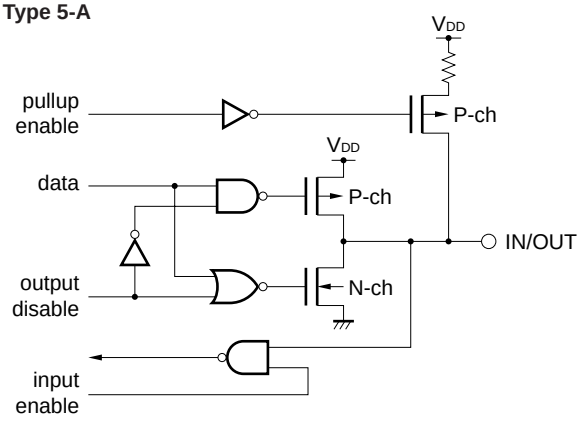


Push-pull output that can be set for high impedance output (both P-ch and N-ch are off)

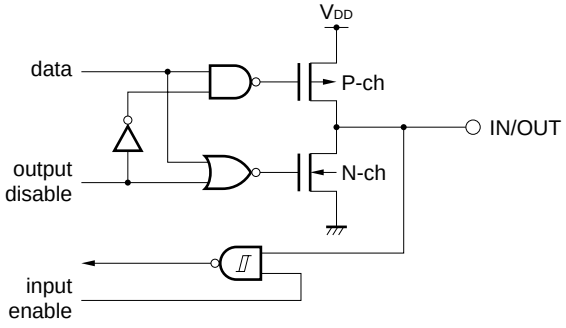
Type 5



Type 5-A

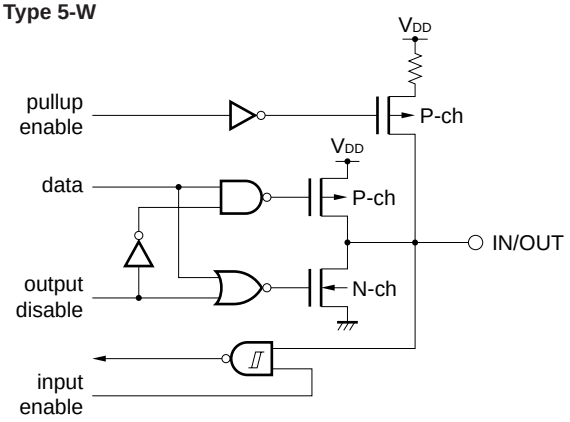


Type 5-K

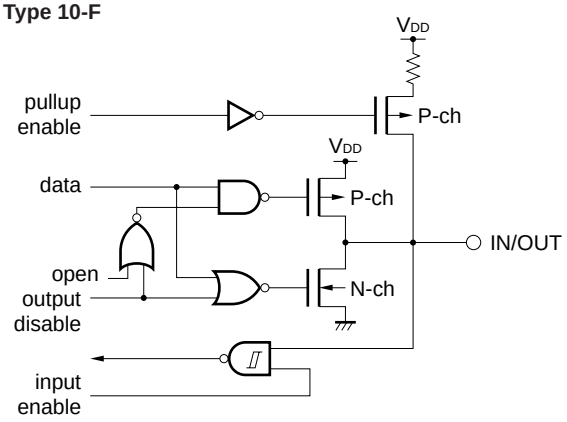


(2/2)

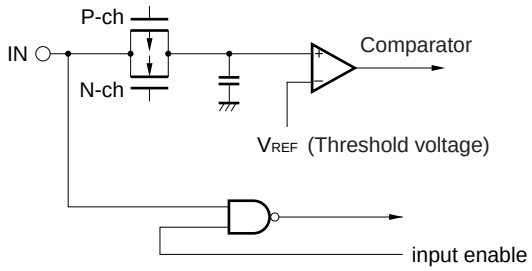
Type 5-W



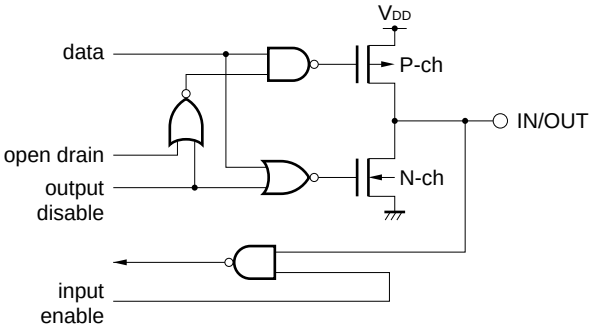
Type 10-F



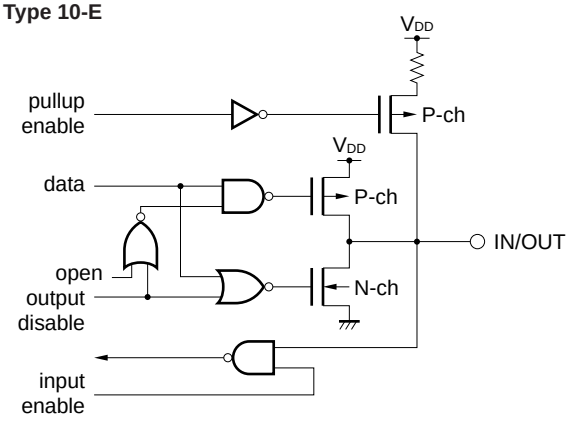
Type 9



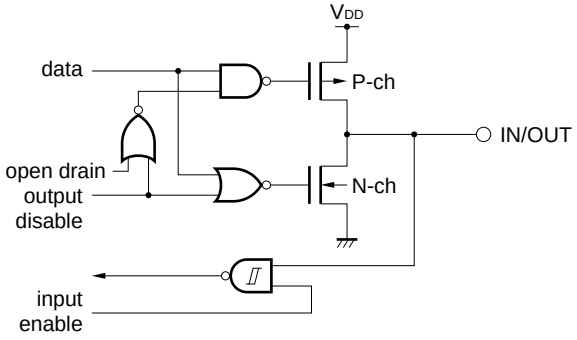
Type 10-G



Type 10-E



Type 10-H



[MEMO]

CHAPTER 3 CPU FUNCTIONS

The CPU of the V850/SV1 is based on the RISC architecture and executes most instructions in one clock cycle by using a 5-stage pipeline.

3.1 Features

- Minimum instruction execution time: 50 ns (at 20 MHz: μPD703039, 703040, 703041, and 70F3040)
  59 ns (at 17 MHz: μPD703039Y, 703040Y, 703041Y, and 70F3040Y)
- Address space: 16 M-byte linear (physical address space: 4 Mbytes)
- Thirty-two 32-bit general registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instruction with long/short format
- Four types of bit manipulation instructions
  - Set
  - Clear
  - Not
  - Test

3.2 CPU Register Set

The CPU registers of the V850/SV1 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers are 32 bits width. For details, refer to **V850 Family User's Manual-Architecture**.

Program register set

31		0
r0	Zero Register	
r1	Reserved for Address Register	
r2	Interrupt Stack Pointer	
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 registers and a program counter.

(1) General registers

Thirty-two general 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 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.

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	Interrupt stack pointer	Stack pointer for interrupt handler
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area[Note]
r6 to r29	–	Address/data variable registers
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function
PC	Program counter	Holds instruction address during program execution

Note Area in which program code is mapped.

(2) Program counter

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

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

After reset: 00000000H

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

3.2.2 System register set

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

Table 3-2. System Register Numbers

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

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

(1) Interrupt source register (ECR)

After reset: 00000000H

Symbol	31	16	15	0
ECR	FECC		EICC	

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

(2) Program status word (PSW)

After reset: 00000020H

Symbol	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	Indicates that NMI processing is in progress. This flag is set when NMI is accepted, and disables multiple interrupts.
EP	Indicates that exception processing is in progress. This flag is set when exception is generated. Moreover, interrupt requests can be accepted when this bit is sets.
ID	Indicates that accepting external interrupt request is disabled.
SAT	This flag is set if the result of executing saturated operation instruction overflows. If overflow does not occur, value of previous operation is held.
CY	This flag is set if carry or borrow occurs as result of operation. If carry or borrow does not occur, it is reset.
OV	This flag is set if overflow occurs during operation. If overflow does not occur, it is reset.
S	This flag is set if the result of operation is negative. It is reset if the result is positive.
Z	This flag is set if the result of operation is zero. If the result is not zero, it is reset.

3.3 Operation Modes

The V850/SV1 has the following operations modes.

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

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

(2) Flash memory programming mode

This mode is provided only in the μPD70F3040 and 70F3040Y. The internal flash memory is programmable or erasable when the V_{PP} voltage is applied to the V_{PP} pin.

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

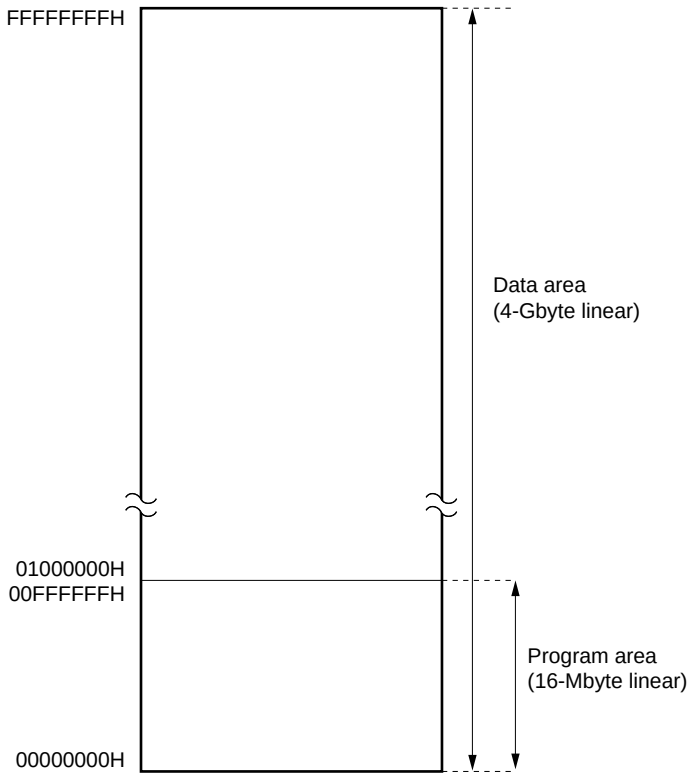
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850/SV1 is of 32-bit architecture and supports up to 4 Gbytes of linear address space (data space) during operand addressing (data access). When referencing instruction addresses, linear address space (program space) of up to 16 Mbytes is supported.

Figure 3-1 shows the CPU address space.

Figure 3-1. CPU Address Space

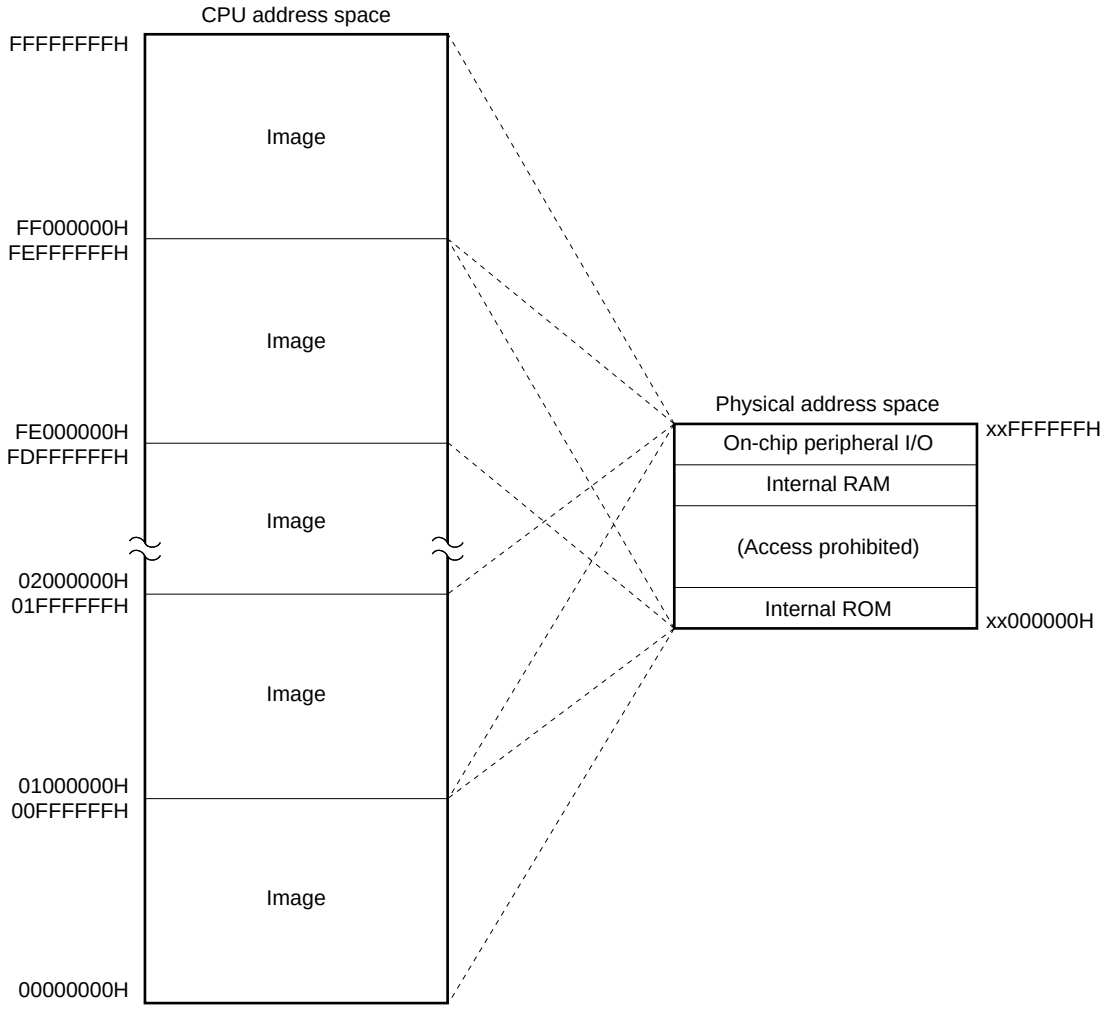


3.4.2 Image

The core CPU supports 4 Gbytes of “virtual” addressing space, or 256 memory blocks, each containing 16-Mbyte memory locations. In actuality, the same 16-Mbyte block is accessed regardless of the values of bits 31 to 24 of the CPU address. Figure 3-2 shows the image of the virtual addressing space.

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

Figure 3-2. Image on Address Space



3.4.3 Wrap-around of CPU address space

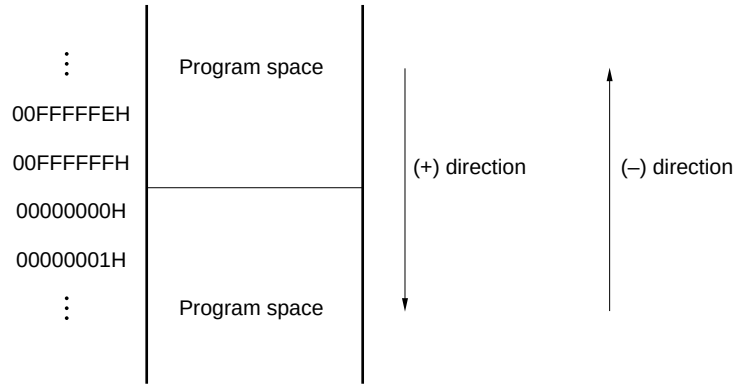
(1) Program space

Of the 32 bits of the PC (program counter), the higher 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-Kbyte 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 3-3. Program Space

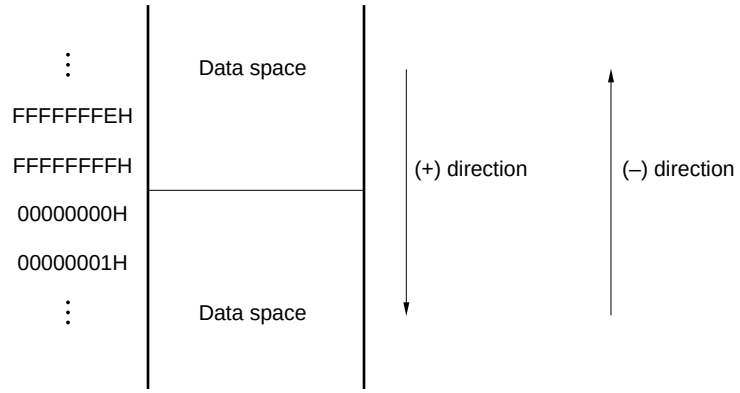


(2) Data space

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

Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address FFFFFFFFH are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.

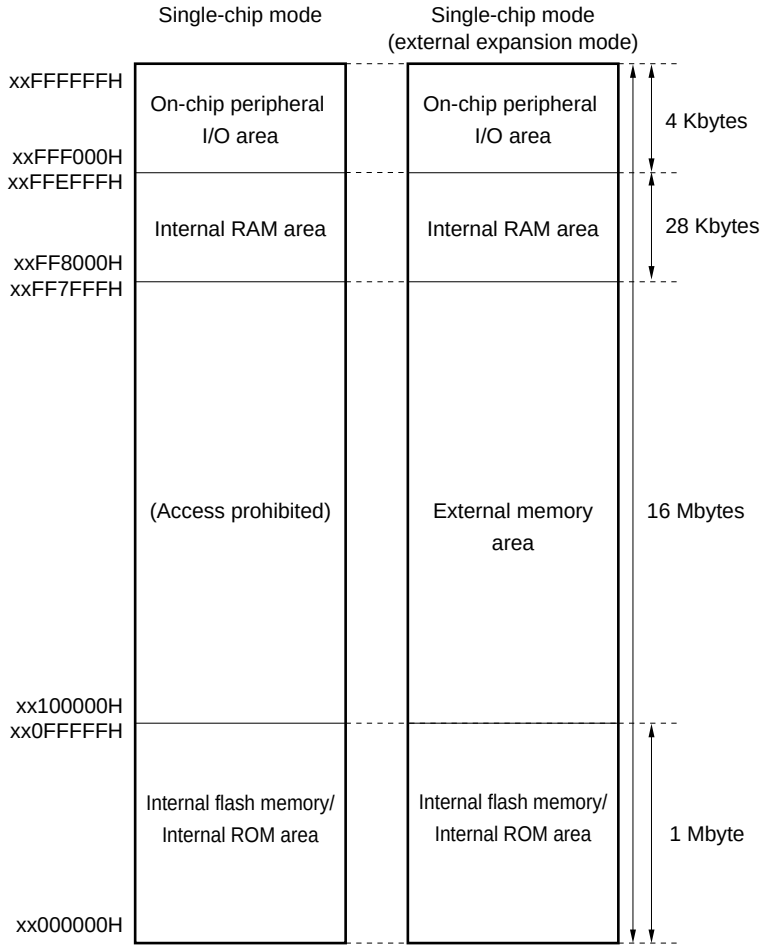
Figure 3-4. Data Space



3.4.4 Memory map

The V850/SV1 reserves areas as shown below.

Figure 3-5. Memory Map



3.4.5 Area

(1) Internal ROM/Internal flash memory area

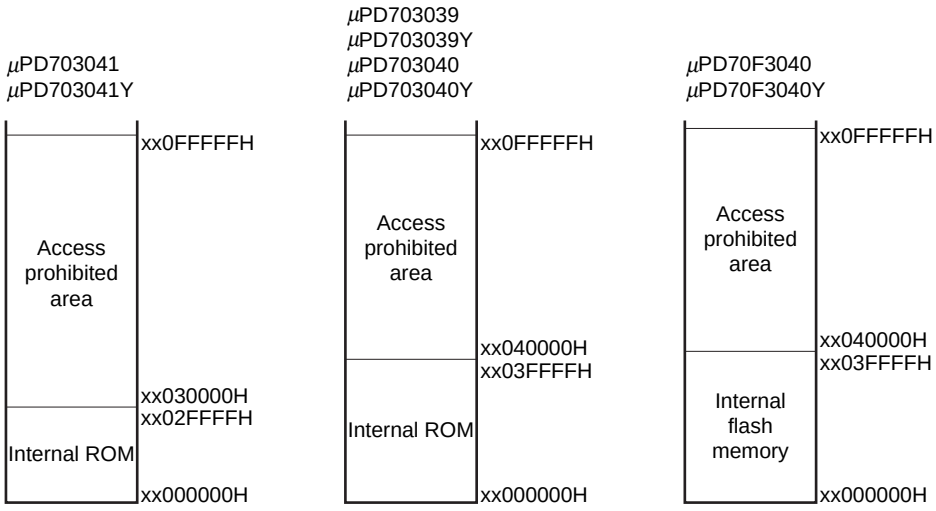
An area of 1-Mbyte maximum is reserved for the internal ROM/internal flash memory area.
The internal ROM area of the V850/SV1 varies depending on the model.

- μPD703041 and 703041Y
  Internal ROM: xx000000H to xx02FFFFH (192 Kbytes)
  Access to the area between xx030000H and xx0FFFFFH is prohibited.

- μPD703039, 703039Y, 703040, and 703040Y
  Internal ROM: xx000000H to xx03FFFFH (256 Kbytes)
  Access to the area between xx040000H and xx0FFFFFH is prohibited.

- μPD70F3040 and 70F3040Y
  Internal flash memory: xx000000H to xx03FFFFH (256 Kbytes)
  Access to the area between xx040000H and xx0FFFFFH is prohibited.

Figure 3-6. Internal ROM/Internal Flash Memory Area



Interrupt/exception table

The V850/SV1 increases the interrupt response speed by assigning handler addresses corresponding to interrupts/exceptions.
The collection of these handler addresses is called an interrupt/exception table, which is located in the internal ROM area. When an interrupt/exception request is granted, execution jumps to the handler address, and the program written at that memory address is executed. Table 3-3 shows the sources of interrupts/exceptions, and the corresponding addresses.

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	000001F0H	INTTM010
00000010H	NMI	00000200H	INTTM011
00000020H	INTWDT	00000210H	INTTM2
00000040H	TRAP0n (n = 0 to F)	00000220H	INTTM3
00000050H	TRAP1n (n = 0 to F)	00000230H	INTTM4
00000060H	ILGOP	00000240H	INTTM5
00000080H	INTWDTM	00000250H	INTTM6
00000090H	INTP0	00000260H	INTTM7
000000A0H	INTP1	00000270H	INTTM10
000000B0H	INTP2	00000280H	INTTM11
000000C0H	INTP3	00000290H	INTIIC0[Note]/INTCSI0
000000D0H	INTP4	000002A0H	INTSER0
000000E0H	INTP5	000002B0H	INTSR0/INTCSI1
000000F0H	INTP6	000002C0H	INTST0
00000100H	INTWTNI	000002D0H	INTCSI2
00000110H	INTOV8/INTTCLR8/ INTTI8	000002E0H	INTIIC1[Note]
00000120H	INTOV9/INTTI9	000002F0H	INTSER1
00000130H	INTCP80/INTCM80	00000300H	INTSR1/INTCSI3
00000140H	INTCP81/INTCM81	00000310H	INTST1
00000150H	INTCP82/INTCM82	00000320H	INTCSI4
00000160H	INTCP83/INTCM83	00000330H	INTAD
00000170H	INTCP90	00000340H	INTDMA0
00000180H	INTCP91	00000350H	INTDMA1
00000190H	INTCP92	00000360H	INTDMA2
000001A0H	INTCP93	00000370H	INTDMA3
000001B0H	INTCM90	00000380H	INTDMA4
000001C0H	INTCM91	00000390H	INTDMA5
000001D0H	INTTM000	000003A0H	INTWTN
000001E0H	INTTM001	000003B0H	INTKR

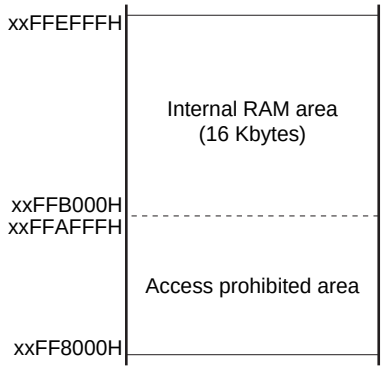
Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

(2) Internal RAM area

The internal RAM area of the V850/SV1 varies depending on the model.

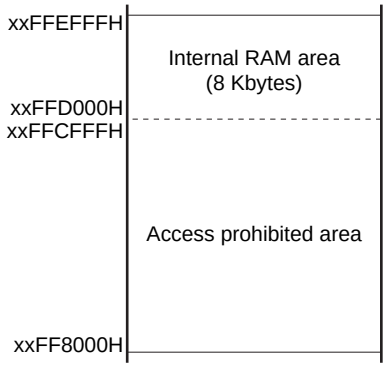
- μPD703040, 703040Y, 70F3040, and 70F3040Y: FFB000 to FFEFFFH (16 Kbytes)
- μPD703039, 703039Y, 703041, and 703041Y: FFD000 to FFEFFFH (8 Kbytes)

Figure 3-7. Internal RAM Area (μPD703040, 703040Y, 70F3040, and 70F3040Y)



Caution Access to the area between FF8000H and FFAFFFH is prohibited.

Figure 3-8. Internal RAM Area (μPD703039, 703039Y, 703041, and 703041Y)



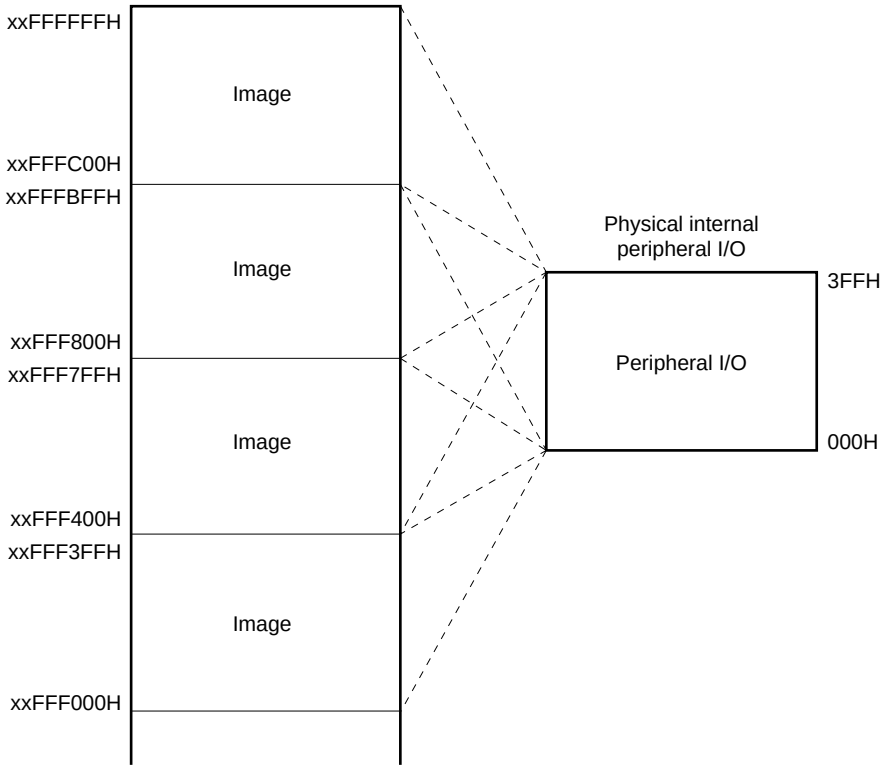
Caution Access to the area between FF8000H and FFCFFFH is prohibited.

(3) Internal peripheral I/O area

A 4-Kbyte area of addresses FFF000H to FFFFFFH is reserved as an internal peripheral I/O area. The V850/SV1 is provided with a 1-Kbyte area of addresses FFF000H to FFF3FFH as a physical internal peripheral I/O area, and its image can be seen on the rest of the area (FFF400H to FFFFFFH).

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

Figure 3-9. Internal 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 half-word units, the higher 8 bits become undefined, if the access is a read operation. If a write access is made, only the data in the lower 8 bits is written to the register.

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

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

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

(4) External memory

The V850/SV1 can use an area of up to 16 Mbytes (xx100000H to xxFF7FFFH) for external memory accesses (in single-chip mode: external expansion).

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

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

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

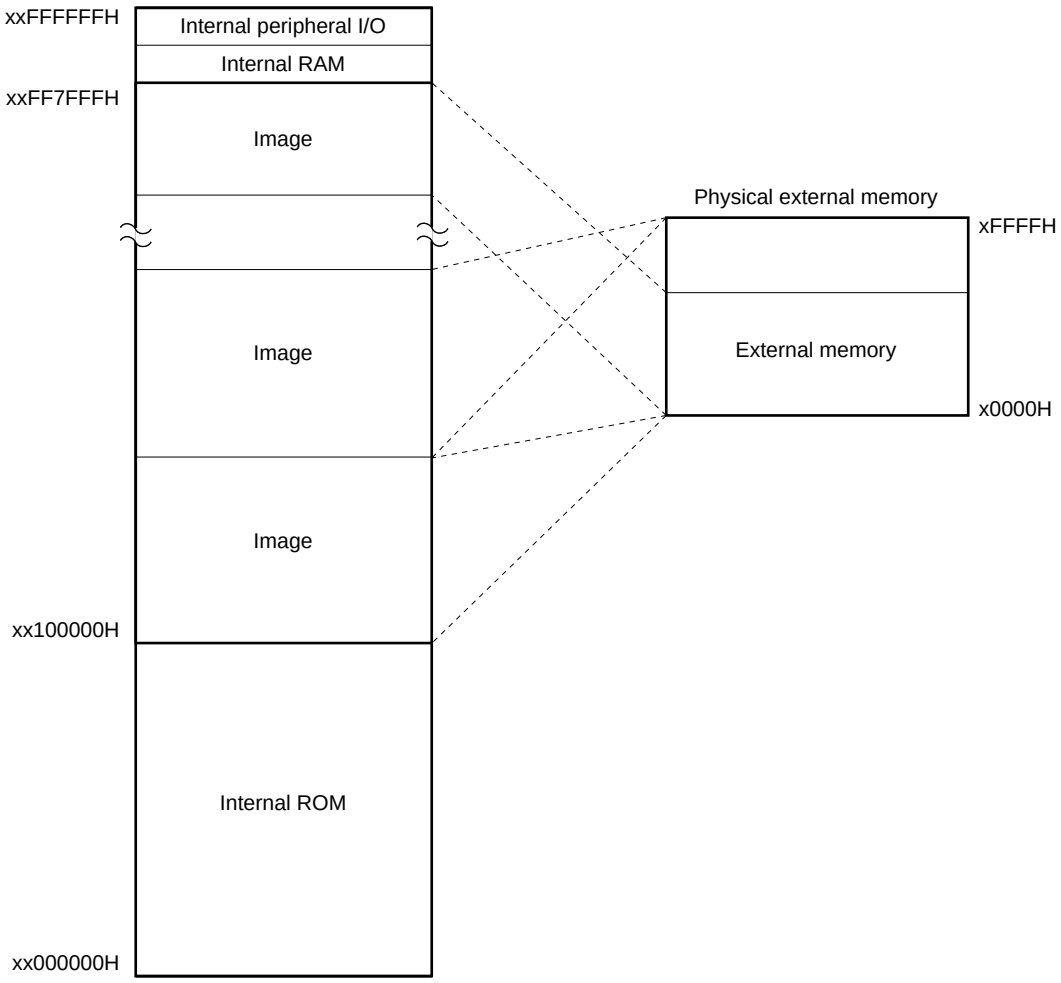
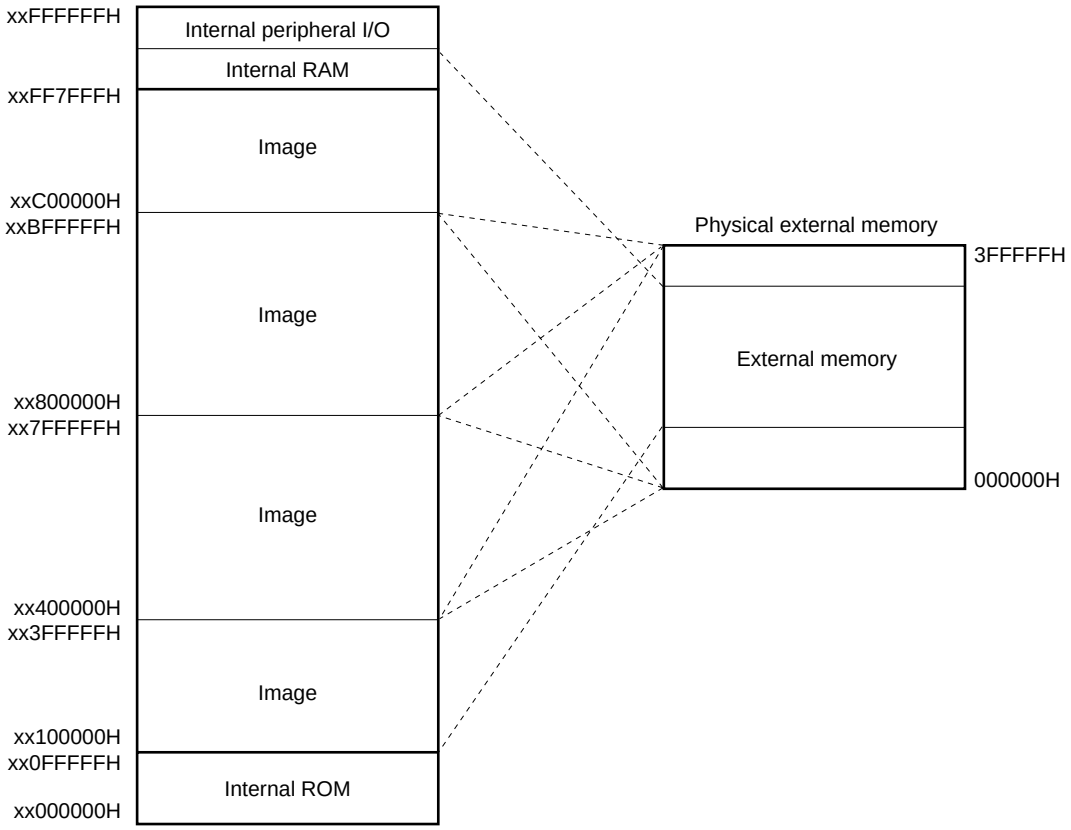


Figure 3-11. External Memory Area (When Expanded to 4 Mbytes)



3.4.6 External expansion mode

The V850/SV1 allows external devices to be connected to the external memory space by using the pins of ports 4, 5, 6, and 9. To connect an external device, the port pins must be set in the external expansion mode by using the memory expansion mode register (MM).

Because the V850/SV1 is fixed to single-chip mode in the normal operation mode, the port/control mode alternate pins become the port mode, thereby the external memory cannot be used. When the external memory is used (external expansion mode), specify the MM register by the program.

(1) Memory expansion mode register (MM)

This register sets the mode of each pin of ports 4, 5, 6, and 9. In the external expansion mode, an external device can be connected to the external memory area of up to 4 Mbytes. However, the external device cannot be connected to the internal RAM area, internal 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.

Caution Before switching to external expansion mode by setting the MM register, bits 3 and 4 of the P9 register must be set to 1 (refer to 16.2.8 Port 9).

Figure 3-12. Memory Expansion Mode Register (MM)

After reset: 00H R/W Address: FFFFF04CH

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

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

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

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

3.4.7 Recommended use of address space

The architecture of the V850/SV1 requires that a register that serves as a pointer be secured for address generation in operand data accessing for data space. The address in this pointer register ±32 Kbytes can be accessed directly from instruction. However, general register used as a pointer register is limited. Therefore, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general 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 map of the V850/SV1, 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-Mbyte space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

(2) Data space

For the efficient use of resources to be performed through the wrap-around feature of the data space, the continuous 8-Mbyte address spaces 00000000H to 007FFFFFH and FF800000H to FFFFFFFFH of the 4-Gbyte CPU are used as the data space. With the V850/SV1, 16-Mbyte physical address space is seen as 256 images in the 4-Gbyte CPU address space. The highest bit (bit 23) of this 24-bit address is assigned as address sign-extended to 32 bits.

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 Kbytes can be referenced with the sign-extended, 16-bit displacement value. By mapping the external memory in the 24-Kbyte area in the figure, all resources including on-chip hardware can be accessed with one pointer.

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

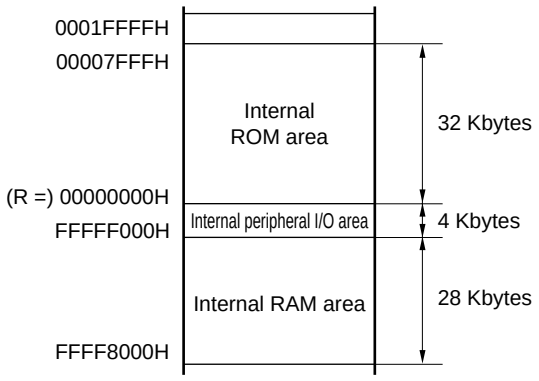
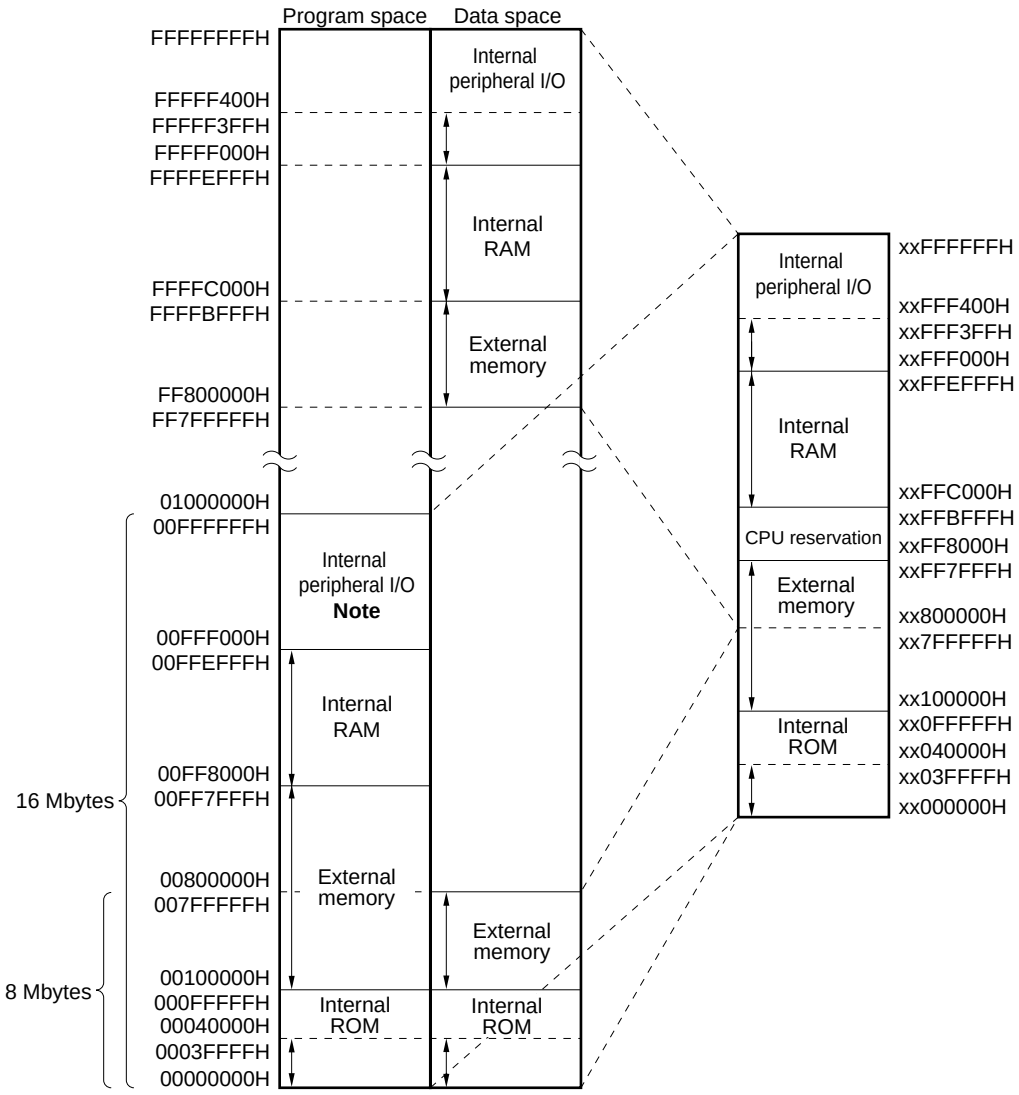


Figure 3-13. Recommended Memory Map Example (Flash Memory Version)



Note This area cannot be used as a program area.

Remark The arrows indicate the recommended area.

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		√	√			3FH
FFFFF024H	Port 2 mode register	PM2		√	√			FFH
FFFFF026H	Port 3 mode register	PM3		√	√			
FFFFF028H	Port 4 mode register	PM4		√	√			
FFFFF02AH	Port 5 mode register	PM5		√	√			
FFFFF02CH	Port 6 mode register	PM6		√	√			3FH
FFFFF032H	Port 9 mode register	PM9		√	√			7FH
FFFFF034H	Port 10 mode register	PM10		√	√			FFH
FFFFF036H	Port 11 mode register	PM11		√	√			0FH
FFFFF038H	Port 12 mode register	PM12		√	√			FFH
FFFFF03AH	Port 13 mode register	PM13		√	√			
FFFFF03CH	Port 14 mode register	PM14		√	√			
FFFFF03EH	Port 15 mode register	PM15		√	√			
FFFFF04CH	Memory expansion mode register	MM		√	√			00H
FFFFF060H	Data wait control register	DWC				√		FFFFH
FFFFF062H	Bus cycle control register	BCC				√		AAAAH
FFFFF064H	System control register	SYC		√	√			00H

Note Resetting initializes registers to input mode and 00H cannot actually be read.

(2/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF070H	Power save control register	PSC	R/W	√	√			C0H
FFFFF074H	Processor clock control register	PCC		√	√			03H
FFFFF078H	System status register	SYS		√	√			00H
FFFFF080H	Port 16	P16		√	√			00H[Note]
FFFFF082H	Port 17	P17		√	√			
FFFFF084H	Port 18	P18		√	√			
FFFFF086H	Port 19	P19		√	√			
FFFFF090H	Port 16 mode register	PM16		√	√			FFH
FFFFF092H	Port 17 mode register	PM17		√	√			
FFFFF094H	Port 18 mode register	PM18		√	√			
FFFFF096H	Port 19 mode register	PM19		√	√			
FFFFF0A0H	Pull-up resistor option register 0	PU0		√	√			00H
FFFFF0A2H	Pull-up resistor option register 1	PU1		√	√			
FFFFF0A4H	Pull-up resistor option register 2	PU2		√	√			
FFFFF0A6H	Pull-up resistor option register 3	PU3		√	√			
FFFFF0A8H	Pull-up resistor option register 10	PU10		√	√			
FFFFF0AAH	Pull-up resistor option register 17	PU17		√	√			
FFFFF0B0H	Port 1 function register	PF1		√	√			
FFFFF0B2H	Port 2 function register	PF2		√	√			
FFFFF0B4H	Port 10 function register	PF10		√	√			
FFFFF0B6H	Port 12 function register	PF12		√	√			
FFFFF0C0H	Rising edge specification register 0	EGP0		√	√			
FFFFF0C2H	Falling edge specification register 0	EGN0		√	√			
FFFFF0C8H	Rising edge specification register 2	EGP2		√	√			
FFFFF0CAH	Falling edge specification register 2	EGN2		√	√			
FFFFF0CCH	Rising edge specification register 3	EGP3		√	√			
FFFFF0CEH	Falling edge specification register 3	EGN3		√	√			
FFFFF0D0H	Vsync up/down counter	VSUDC	R		√			
FFFFF0D2H	Vsync compare register	VSCMP	R/W		√			
FFFFF0D4H	Hsync compare register	HSCMP			√			
FFFFF0D6H	Hsync mask width register	HMCMP			√			
FFFFF0D8H	Hsync compensation regiser	HCCMP			√			
FFFFF0DAH	Vsync control register	VSC		√	√			
FFFFF0E0H	PWM control register 0	PWMC0		√	√			05H
FFFFF0E2H	PWM modulo register 0	PWM0				√		Undefined
FFFFF0E4H	PWM prescaler register 0	PWPR0			√			00H

Note Resetting initializes registers to input mode and 00H cannot actually be read.

(3/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF0E8H	PWM control register 1	PWMC1	R/W	√	√			05H
FFFFF0EAH	PWM modulo register 1	PWM1				√		Undefined
FFFFF0ECH	PWM prescaler register 1	PWPR1			√			00H
FFFFF0F0H	PWM control register 2	PWMC2		√	√			05H
FFFFF0F2H	PWM modulo register 2	PWM2				√		Undefined
FFFFF0F4H	PWM prescaler register 2	PWPR2			√			00H
FFFFF0F8H	PWM control register 3	PWMC3		√	√			05H
FFFFF0FAH	PWM modulo register 3	PWM3				√		Undefined
FFFFF0FCH	PWM prescaler register 3	PWPR3			√			00H
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	WTNIIC		√	√			
FFFFF112H	Interrupt control register	OVIC8		√	√			
FFFFF114H	Interrupt control register	OVIC9		√	√			
FFFFF116H	Interrupt control register	CC8IC0		√	√			
FFFFF118H	Interrupt control register	CC8IC1		√	√			
FFFFF11AH	Interrupt control register	CC8IC2		√	√			
FFFFF11CH	Interrupt control register	CC8IC3		√	√			
FFFFF11EH	Interrupt control register	CP9IC0		√	√			
FFFFF120H	Interrupt control register	CP9IC1		√	√			
FFFFF122H	Interrupt control register	CP9IC2		√	√			
FFFFF124H	Interrupt control register	CP9IC3		√	√			
FFFFF126H	Interrupt control register	CM9IC0		√	√			
FFFFF128H	Interrupt control register	CM9IC1		√	√			
FFFFF12AH	Interrupt control register	TMIC000		√	√			
FFFFF12CH	Interrupt control register	TMIC001		√	√			
FFFFF12EH	Interrupt control register	TMIC010		√	√			
FFFFF130H	Interrupt control register	TMIC011		√	√			
FFFFF132H	Interrupt control register	TMIC2		√	√			
FFFFF134H	Interrupt control register	TMIC3		√	√			

(4/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF136H	Interrupt control register	TMIC4	R/W	√	√			47H
FFFFF138H	Interrupt control register	TMIC5		√	√			
FFFFF13AH	Interrupt control register	TMIC6		√	√			
FFFFF13CH	Interrupt control register	TMIC7		√	√			
FFFFF13EH	Interrupt control register	TMIC10		√	√			
FFFFF140H	Interrupt control register	TMIC11		√	√			
FFFFF142H	Interrupt control register	CSIC0		√	√			
FFFFF144H	Interrupt control register	SERIC0		√	√			
FFFFF146H	Interrupt control register	CSIC1		√	√			
FFFFF148H	Interrupt control register	STIC0		√	√			
FFFFF14AH	Interrupt control register	CSIC2		√	√			
FFFFF14CH	Interrupt control register[Note]	IICIC1		√	√			
FFFFF14EH	Interrupt control register	SERIC1		√	√			
FFFFF150H	Interrupt control register	CSIC3		√	√			
FFFFF152H	Interrupt control register	STIC1		√	√			
FFFFF154H	Interrupt control register	CSIC4		√	√			
FFFFF156H	Interrupt control register	ADIC		√	√			
FFFFF158H	Interrupt control register	DMAIC0		√	√			
FFFFF15AH	Interrupt control register	DMAIC1		√	√			
FFFFF15CH	Interrupt control register	DMAIC2		√	√			
FFFFF15EH	Interrupt control register	DMAIC3		√	√			
FFFFF160H	Interrupt control register	DMAIC4		√	√			
FFFFF162H	Interrupt control register	DMAIC5		√	√			
FFFFF164H	Interrupt control register	WTNIC		√	√			
FFFFF166H	In-service priority register	ISPR	R	√	√			00H
FFFFF168H	Interrupt control register	KRIC	R/W	√	√			47H
FFFFF170H	Command register	PRCMD	W		√			Undefined
FFFFF180H	DMA peripheral I/O address register 0	DIOA0	R/W			√		
FFFFF182H	DMA internal RAM address register 0	DRA0				√		
FFFFF184H	DMA byte count register 0	DBC0			√			
FFFFF186H	DMA channel control register 0	DCHC0		√	√			00H
FFFFF190H	DMA peripheral I/O address register 1	DIOA1				√		Undefined
FFFFF192H	DMA internal RAM address register 1	DRA1				√		
FFFFF194H	DMA byte count register 1	DBC1			√			
FFFFF196H	DMA channel control register 1	DCHC1		√	√			00H

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y

(5/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF1A0H	DMA peripheral I/O address register 2	DIOA2	R/W			√		Undefined
FFFFF1A2H	DMA internal RAM address register 2	DRA2				√		
FFFFF1A4H	DMA byte count register 2	DBC2			√			
FFFFF1A6H	DMA channel control register 2	DCHC2		√	√			00H
FFFFF1B0H	DMA peripheral I/O address register 3	DIOA3				√		Undefined
FFFFF1B2H	DMA internal RAM address register 3	DRA3				√		
FFFFF1B4H	DMA byte count register 3	DBC3			√			
FFFFF1B6H	DMA channel control register 3	DCHC3		√	√			00H
FFFFF1C0H	DMA peripheral I/O address register 4	DIOA4				√		Undefined
FFFFF1C2H	DMA internal RAM address register 4	DRA4				√		
FFFFF1C4H	DMA byte count register 4	DBC4			√			
FFFFF1C6H	DMA channel control register 4	DCHC4		√	√			00H
FFFFF1D0H	DMA peripheral I/O address register 5	DIOA5				√		Undefined
FFFFF1D2H	DMA internal RAM address register 5	DRA5				√		
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	**Note**			√		
FFFFF206H	Prescaler mode register 00	PRM00	R/W		√			00H
FFFFF208H	16-bit timer mode control register 0	TMC0		√	√			
FFFFF20AH	Capture/compare control register 0	CRC0		√	√			
FFFFF20CH	Timer output control register 0	TOC0		√	√			
FFFFF20EH	Prescaler mode register 01	PRM01			√			
FFFFF210H	16-bit timer register 1	TM1	R			√		0000H
FFFFF212H	16-bit capture/compare register 10	CR10	**Note**			√		
FFFFF214H	16-bit capture/compare register 11	CR11	**Note**			√		
FFFFF216H	Prescaler mode register 10	PRM10	R/W		√			00H
FFFFF218H	16-bit timer mode control register 1	TMC1		√	√			
FFFFF21AH	Capture/compare control register 1	CRC1		√	√			
FFFFF21CH	Timer output control register 1	TOC1		√	√			
FFFFF21EH	Prescaler mode register 11	PRM11			√			
FFFFF230H	Real-time output buffer register L0	RTBL0			√			
FFFFF232H	Real-time output buffer register H0	RTBH0			√			
FFFFF234H	Real-time output port mode register 0	RTPM0		√	√			

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	
FFFFF236H	Real-time output port control register 0	RTPC0	R/W	√	√			00H
FFFFF238H	Real-time output buffer register L1	RTBL1			√			
FFFFF23AH	Real-time output buffer register H1	RTBH1			√			
FFFFF23CH	Real-time output port mode register 1	RTPM1		√	√			
FFFFF23EH	Real-time output port control register 1	RTPC1		√	√			
FFFFF240H	8-bit counter 2	TM2	R		√			
FFFFF242H	8-bit compare register 2	CR20	R/W		√			
FFFFF244H	Timer clock select register 20	TCL20			√			
FFFFF246H	8-bit timer mode control register 2	TMC2		√	√			04H[Note]
FFFFF24AH	16-bit counter 23 (during cascade connection only)	TM23	R			√		0000H
FFFFF24CH	16-bit compare register 23 (during cascade connection only)	CR23	R/W			√		
FFFFF24EH	Timer clock select register 21	TCL21			√			00H
FFFFF250H	8-bit counter 3	TM3	R		√			
FFFFF252H	8-bit compare register 3	CR30	R/W		√			
FFFFF254H	Timer clock select register 30	TCL30			√			
FFFFF256H	8-bit timer mode control register 3	TMC3		√	√			04H[Note]
FFFFF25EH	Timer clock select register 31	TCL31			√			00H
FFFFF260H	8-bit counter 4	TM4	R		√			
FFFFF262H	8-bit compare register 4	CR40	R/W		√			
FFFFF264H	Timer clock select register 40	TCL40			√			
FFFFF266H	8-bit timer mode control register 4	TMC4		√	√			04H[Note]
FFFFF26AH	16-bit counter 45 (during cascade connection only)	TM45	R			√		0000H
FFFFF26CH	16-bit compare register 45 (during cascade connection only)	CR45	R/W			√		
FFFFF26EH	Timer clock select register 41	TCL41			√			00H
FFFFF270H	8-bit counter 5	TM5	R		√			
FFFFF272H	8-bit compare register 5	CR50	R/W		√			
FFFFF274H	Timer clock select register 50	TCL50			√			
FFFFF276H	8-bit timer mode control register 5	TMC5		√	√			04H[Note]
FFFFF27EH	Timer clock select register 51	TCL51			√			00H
FFFFF280H	8-bit counter 6	TM6	R		√			
FFFFF282H	8-bit compare register 6	CR60	R/W		√			

Note Although the hardware state is initialized to 04H, 00H is read from the hardware.

(7/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF284H	Timer clock select register 60	TCL60	R/W		√			00H
FFFFF286H	8-bit timer mode control register 6	TMC6		√	√			04H[Note]
FFFFF28AH	16-bit counter 67 (during cascade connection only)	TM67	R			√		0000H
FFFFF28CH	16-bit compare register 67 (during cascade connection only)	CR67	R/W			√		
FFFFF28EH	Timer clock select register 61	TCL61			√			00H
FFFFF290H	8-bit counter 7	TM7	R		√			
FFFFF292H	8-bit compare register 7	CR70	R/W		√			
FFFFF294H	Timer clock select register 70	TCL70			√			
FFFFF296H	8-bit timer mode control register 7	TMC7		√	√			04H[Note]
FFFFF29EH	Timer clock select register 71	TCL71			√			00H
FFFFF2A0H	Serial I/O shift register 0	SIO0			√			
FFFFF2A2H	Serial operation mode register 0	CSIM0		√	√			
FFFFF2A4H	Serial clock select register 0	CSIS0			√			
FFFFF2B0H	Serial I/O shift register 1	SIO1			√			
FFFFF2B2H	Serial operation mode register 1	CSIM1		√	√			
FFFFF2B4H	Serial clock select register 1	CSIS1			√			
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 00	ASIM00		√	√			00H
FFFFF302H	Asynchronous serial interface status register 0	ASIS0	R	√	√			
FFFFF304H	Baud rate generator control register 0	BRGC0	R/W		√			

Note Although the hardware state is initialized to 04H, 00H is read from the hardware.

(8/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF306H	Transmit shift register 0	TXS0	W		√			FFH
FFFFF308H	Receive buffer register 0	RXB0	R		√			
FFFFF30EH	Baud rate generator mode control register 00	BRGMC00	R/W		√			00H
FFFFF310H	Asynchronous serial interface mode register 10	ASIM10		√	√			
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		√			
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			√			
FFFFF324H	Asynchronous serial interface mode register 01	ASIM01		√	√			
FFFFF326H	Asynchronous serial interface mode register 11	ASIS11		√	√			
FFFFF328H	A/D converter mode register 0	ADM0		√	√			
FFFFF32AH	A/D converter mode register 1	ADM1		√	√			
FFFFF330H	A/D conversion result register 0	ADCR0	R			√		0000H
FFFFF332H	A/D conversion result register 1	ADCR1				√		
FFFFF334H	A/D conversion result register 2	ADCR2				√		
FFFFF336H	A/D conversion result register 3	ADCR3				√		
FFFFF338H	A/D conversion result register 4	ADCR4				√		
FFFFF33AH	A/D conversion result register 5	ADCR5				√		
FFFFF33CH	A/D conversion result register 6	ADCR6				√		
FFFFF33EH	A/D conversion result register 7	ADCR7				√		
FFFFF340H	IIC control register 0[Note]	IICC0	R/W	√	√			00H
FFFFF342H	IIC status register 0[Note]	IICS0	R	√	√			
FFFFF344H	IIC clock select register 0[Note]	IICCL0	R/W	√	√			
FFFFF346H	Slave address register 0[Note]	SVA0			√			
FFFFF348H	IIC shift register 0[Note]	IIC0			√			
FFFFF34AH	IIC function expansion register 0[Note]	IICX0		√	√			
FFFFF350H	IIC control register 1[Note]	IICC1		√	√			
FFFFF352H	IIC status register 1[Note]	IICS1	R	√	√			
FFFFF354H	IIC clock select register 1[Note]	IICCL1	R/W	√	√			

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y

(9/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF356H	Slave address register 1[Note]	SVA1	R/W		√			00H
FFFFF358H	IIC shift register 1[Note]	IIC1			√			
FFFFF35AH	IIC function expansion register 1[Note]	IICX1		√	√			
FFFFF360H	Watch timer mode register	WTNM		√	√			
FFFFF364H	Watch timer clock select register	WTNCS			√			
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 stabilization time select register	OSTS			√			04H
FFFFF382H	Watchdog timer clock select register	WDCS			√			00H
FFFFF384H	Watchdog timer mode register	WDTM		√	√			
FFFFF388H	Key return mode register	KRM		√	√			
FFFFF38AH	Clock output mode register	CLOM		√	√			
FFFFF38CH	Noise elimination control register	NCC			√			
FFFFF390H	Timer 8	TM8	R				√	00000000H
FFFFF394H	Capture/compare register 80	CC80	R/W				√	Undefined
FFFFF398H	Capture/compare register 81	CC81					√	
FFFFF39CH	Capture/compare register 82	CC82					√	
FFFFF3A0H	Capture/compare register 83	CC83					√	
FFFFF3A4H	24-bit timer mode control register 80	TMC80		√	√			01H
FFFFF3A6H	24-bit timer mode control register 81	TMC81		√	√			00H
FFFFF3A8H	24-bit timer mode control register 82	TMC82		√	√			
FFFFF3AAH	Timer output control register 8	TOC8		√	√			
FFFFF3ACH	Timer overflow status register	TOVS		√	√			
FFFFF3B0H	Timer 9	TM9	R				√	00000000H
FFFFF3B4H	Compare register 90	CM90	R/W				√	Undefined
FFFFF3B8H	Compare register 91	CM91					√	
FFFFF3BCH	Capture register 90	CP90	R				√	
FFFFF3C0H	Capture register 91	CP91					√	
FFFFF3C4H	Capture register 92	CP92					√	
FFFFF3C8H	Capture register 93	CP93					√	
FFFFF3CCH	24-bit timer mode control register 90	TMC90	R/W	√	√			01H

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y

(10/10)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1 bit	8 bits	16 bits	32 bits	
FFFFF3CEH	24-bit timer mode control register 91	TMC91	R/W	√	√			00H
FFFFF3D2H	Event divide counter 0	EDV0	R	√	√			
FFFFF3D4H	Event divide counter 1	EDV1		√	√			
FFFFF3D6H	Event divide counter 2	EDV2		√	√			
FFFFF3D8H	Event divide control register 0	EDVC0	R/W	√	√			
FFFFF3DAH	Event divide control register 1	EDVC1		√	√			
FFFFF3DCH	Event divide control register 2	EDVC2		√	√			
FFFFF3DEH	Event select register	EVS		√	√			
FFFFF3E0H	8-bit counter 10	TM10	R		√			
FFFFF3E2H	8-bit compare register 100	CR100	R/W		√			
FFFFF3E4H	Timer clock select register 100	TCL100			√			
FFFFF3E6H	8-bit timer mode control register 10	TMC10		√	√			04H[Note]
FFFFF3EAH	16-bit counter 1011 (during cascade connection only)	TM1011	R			√		0000H
FFFFF3ECH	16-bit compare register 1011 (during cascade connection only)	CR1011	R/W			√		
FFFFF3EEH	Timer clock select register 101	TCL101			√			00H
FFFFF3F0H	8-bit counter 11	TM11	R		√			
FFFFF3F2H	8-bit compare register 110	CR110	R/W		√			
FFFFF3F4H	Timer clock select register 110	TCL110			√			
FFFFF3F6H	8-bit timer mode control register 11	TMC11		√	√			04H[Note]
FFFFF3FEH	Timer clock select register 111	TCL111			√			00H

Note Although the hardware state is initialized to 04H, 00H is read from the hardware.

3.4.9 Specific registers

Specific registers are registers that are protected from being written with illegal data due to erroneous program execution, etc. The write access of these specific registers is executed in a specific sequence, and if abnormal store operations occur, it is notified by the system status register (SYS). The V850/SV1 has two specific registers, the power save control register (PSC) and processor clock control register (PCC). For details of the PSC register, refer to **6.4.3,** and for details of the PCC register, refer to **6.4.1.**

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

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

<5> Return the PSW NP bit to 0 (interrupt disable canceled).
<6> Insert the NOP instructions (2 or 5 instructions).
<7> If necessary, enable DMA operation.

No special sequence is required when reading the specific registers.

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

The above also applies when a bit manipulation instruction is used to set a specific register.

Moreover, to ensure that the execution routine following release of the STOP/IDLE mode is performed correctly, insert the NOP instruction as a dummy instruction (<6>). If the value of the ID bit of PSW does not change as the result of execution of the instruction to return the NP bit to 0 (<5>), insert two NOP instructions, and if the value of the ID bit of PSW changes, insert five NOP instructions.

A description example is given below.

[Description example]: In case of PSC register

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

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

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

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

3. Always stop the DMA prior to accessing special registers.

(1) Command register (PRCMD)

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

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

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

Figure 3-14. Command Register (PRCMD)

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

(2) System status register (SYS)

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

Figure 3-15. System Status Register (SYS)

After reset: 00H R/W Address: FFFFF078H

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

PRERR	Detection of Protection Error
0	Protection error does not occur
1	Protection error occurs

Operation conditions of PRERR flag are shown as follows.

(a) Set conditions (PRERR = 1)

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

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

(b) Reset conditions (PRERR = 0)

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

(2) At system reset.

[MEMO]

CHAPTER 4 BUS CONTROL FUNCTION

The V850/SV1 is provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

4.1 Features

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

4.2 Bus Control Pins and Control Register

4.2.1 Bus control pins

The following pins are used for interfacing to external devices:

Table 4-1. Bus Control Pins

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

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

4.2.2 Control register

(1) System control register (SYC)

This register switches control signals for bus interface.

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

Figure 4-1. System Control Register (SYC)

After reset: 00H R/W Address: FFFFF064H

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

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

4.3 Bus Access

4.3.1 Number of access clocks

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

Table 4-2. Number of Access Clocks

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

Remarks 1. Unit: Clock/access

2. n: Number of wait insertions

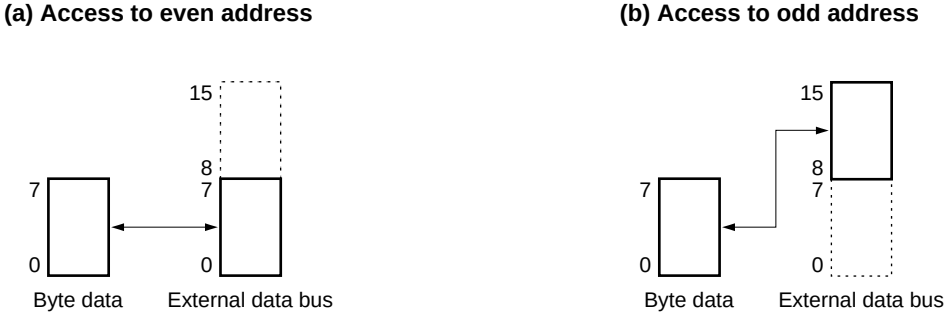
4.3.2 Bus width

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

(1) Byte access (8 bits)

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

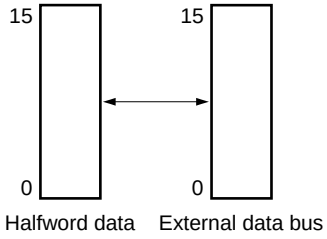
Figure 4-2. Byte Access (8 Bits)



(2) Halfword access (16 bits)

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

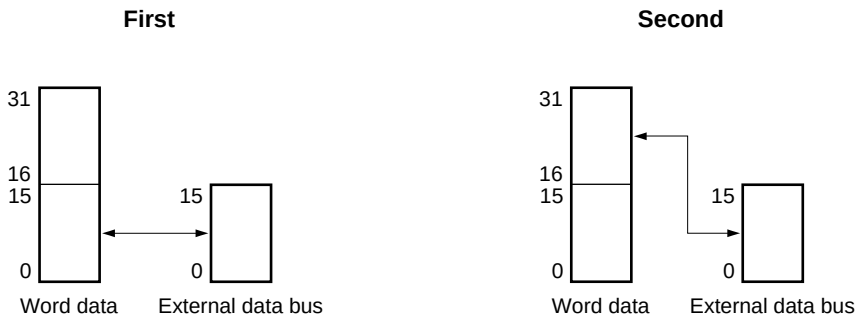
Figure 4-3. Halfword Access (16 Bits)



(3) Word access (32 bits)

In word access to external memory, low-order halfword is accessed first and then the high-order halfword is accessed.

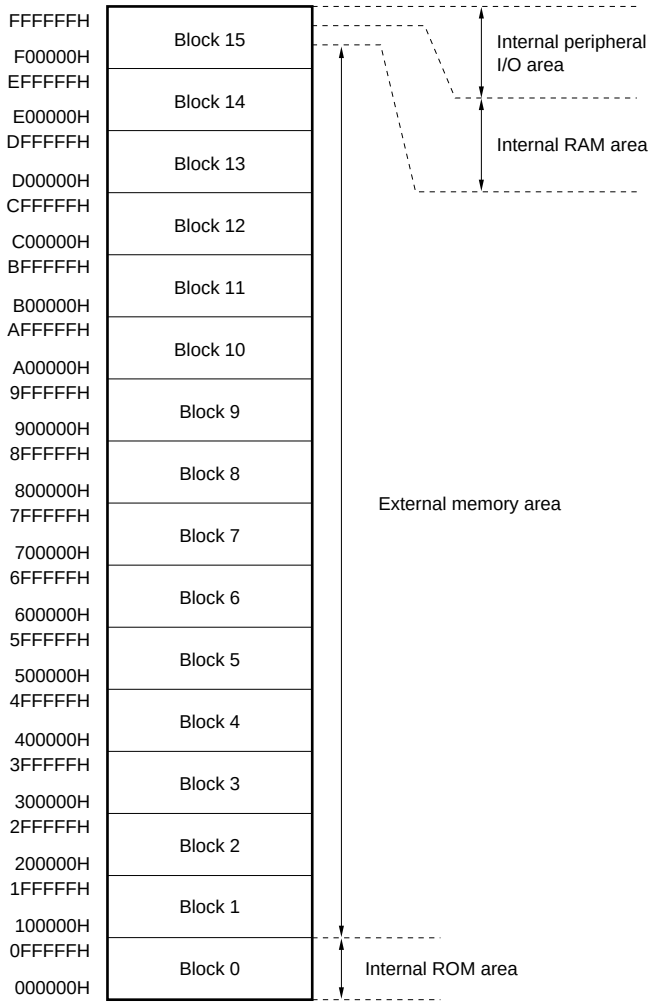
Figure 4-4. Word Access (32 Bits)



4.4 Memory Block Function

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

Figure 4-5. Memory Space



4.5 Wait Function

4.5.1 Programmable wait function

To facilitate interfacing with low-speed memories and I/O devices, up to 3 data 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 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.

Figure 4-6. Data Wait Control Register (DWC)

After reset: FFFFH R/W Address: FFFFF060H

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

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

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

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

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

4.5.2 External wait function

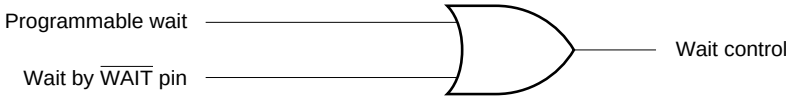
When an extremely slow device, I/O, or asynchronous system is connected, any number of wait states can be inserted in a bus cycle by sampling the external wait pin ($\overline{\text{WAIT}}$) to synchronize with the external device.

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

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

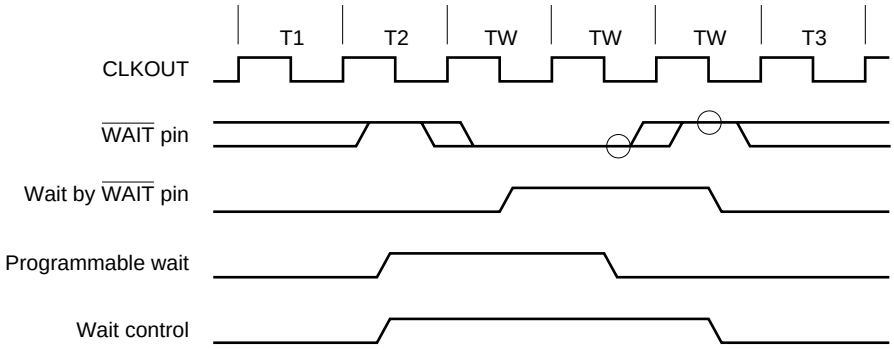
4.5.3 Relationship between programmable wait and external wait

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



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

Figure 4-7. Example of Inserting Wait States



Remark ○: Valid sampling timing

4.6 Idle State Insertion Function

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

Figure 4-8. Bus Cycle Control Register (BCC)

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 the idle state.

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

4.7 Bus Hold Function

4.7.1 Outline of function

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

When the $\overline{\text{HLDRQ}}$ pin becomes active (low) indicating that another bus master is requesting acquisition of the bus, the external address/data bus and strobe pins go into a high-impedance state, 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 bus hold period, the internal operation continues until the next external memory access.

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

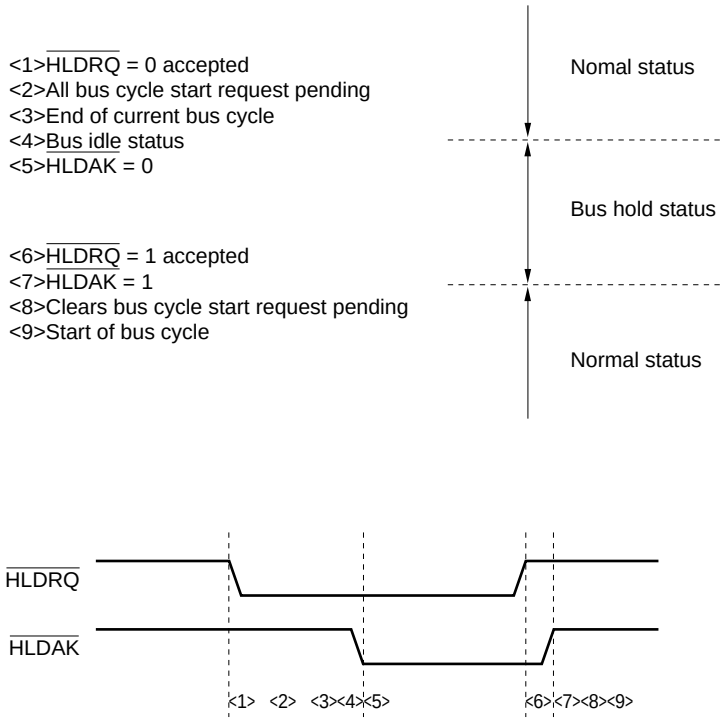
This feature can be used to design a system where two or more bus masters exist, such as when multi-processor configuration is used and when a DMA controller is connected.

Bus hold request is not acknowledged between the first and the second word access, and not acknowledged between read access and write access in read modify write access of bit manipulation instruction either.

4.7.2 Bus hold procedure

The procedure of the bus hold function is illustrated below.

Figure 4-9. Bus Hold Procedure



4.7.3 Operation in power save mode

In the STOP or IDLE 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.

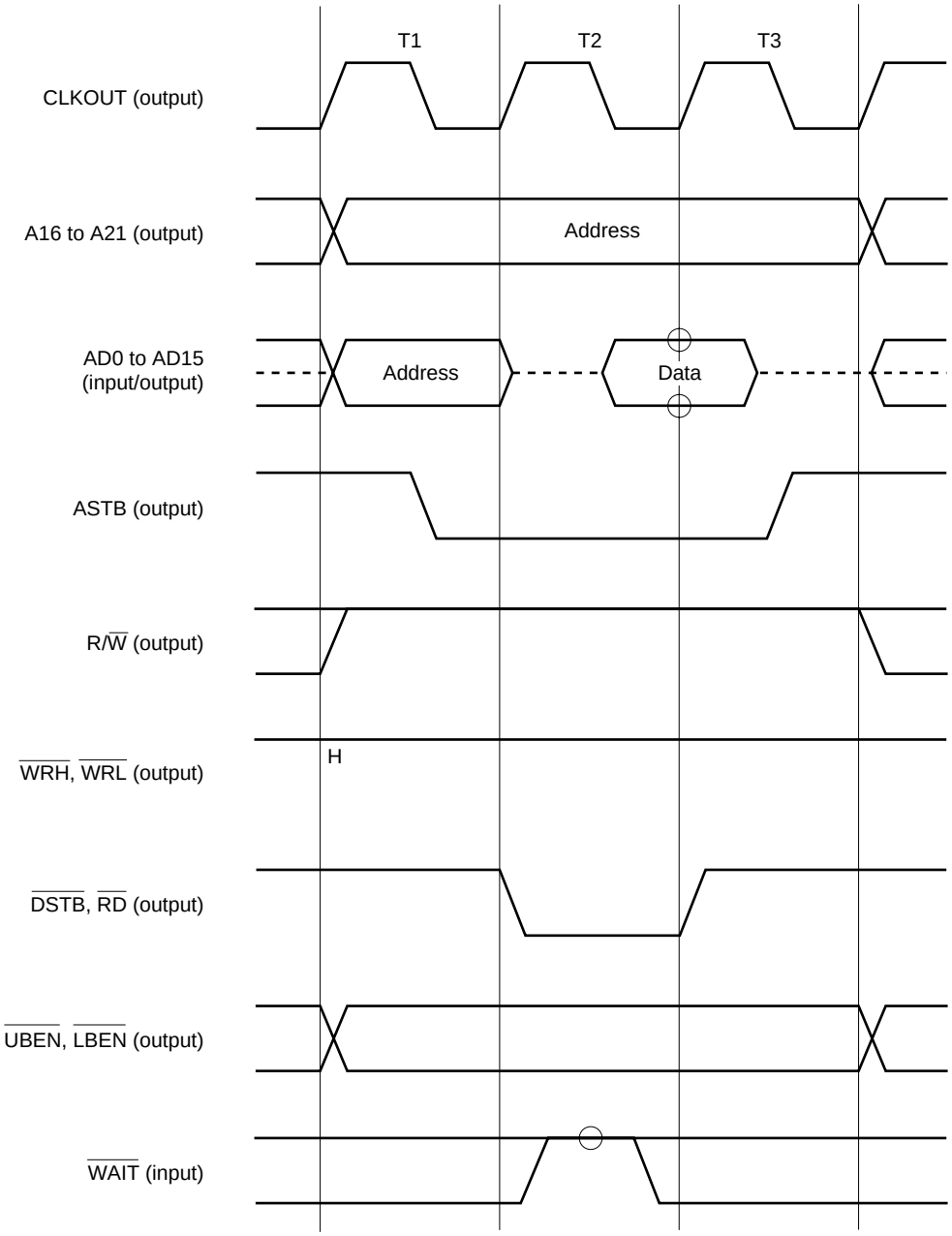
4.8 Bus Timing

The V850/SV1 can execute the read/write control for an external device by 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).

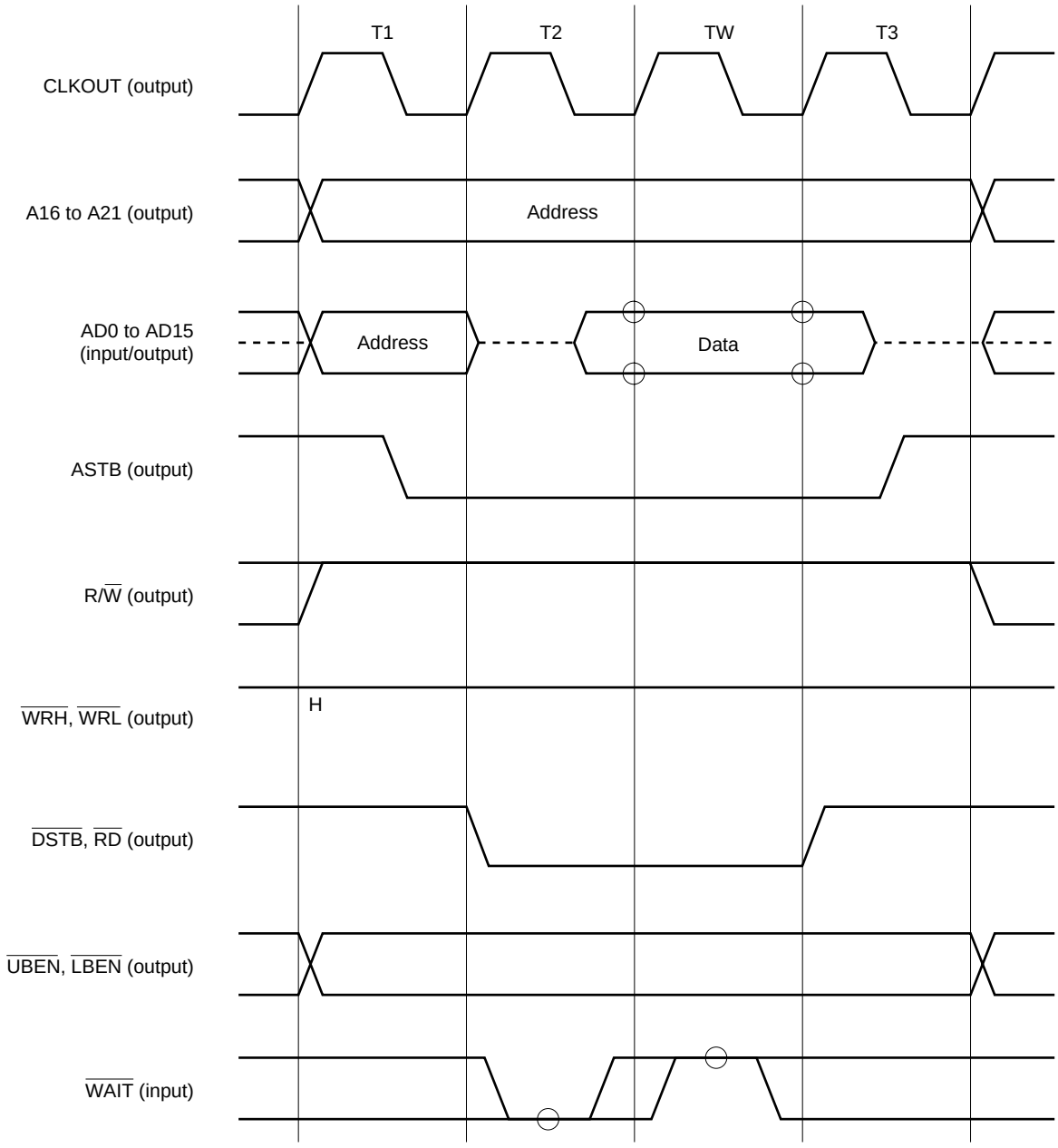
Figure 4-10. Memory Read (0 Wait)



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

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

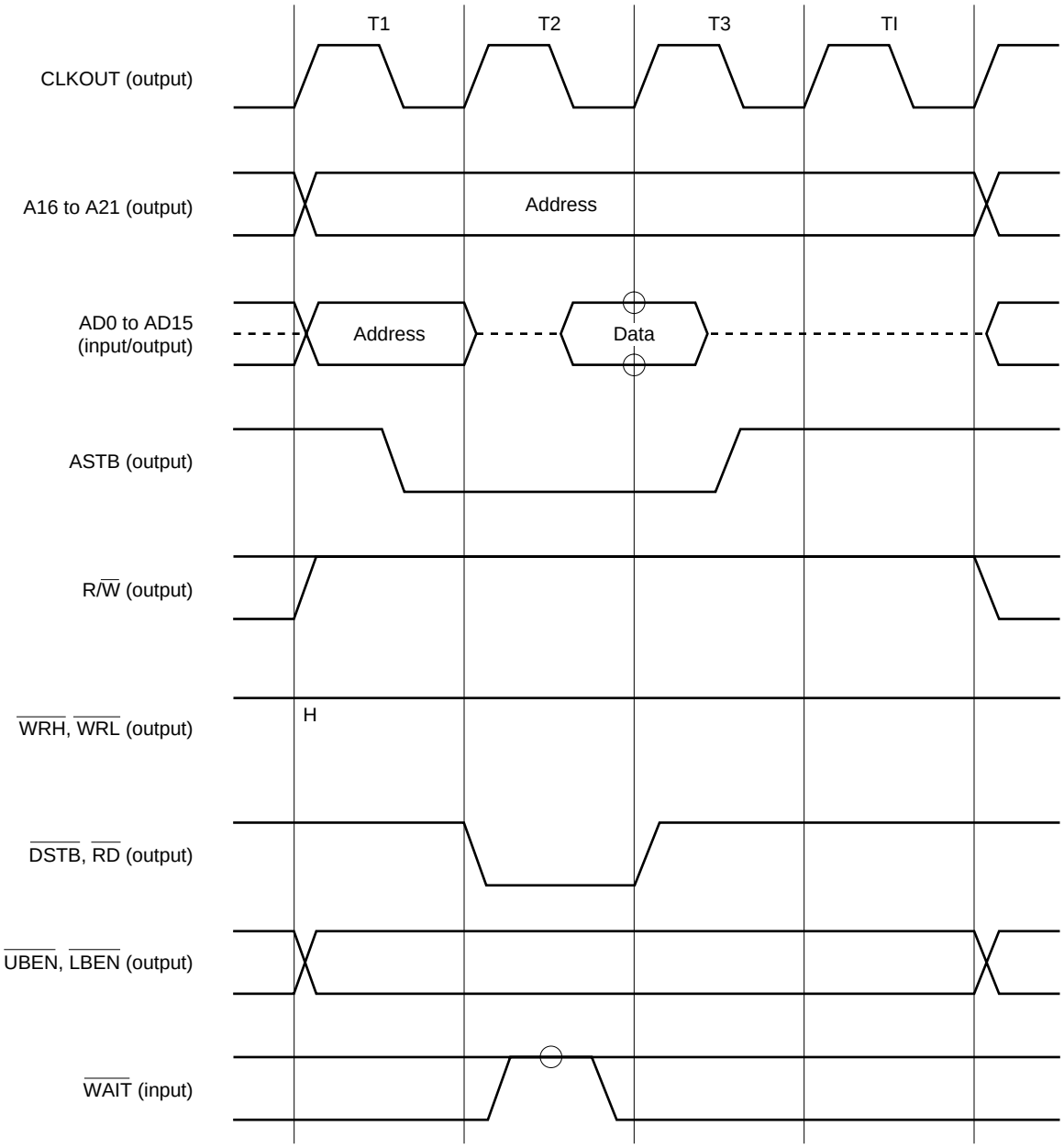
Figure 4-11. Memory Read (1 Wait)



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

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

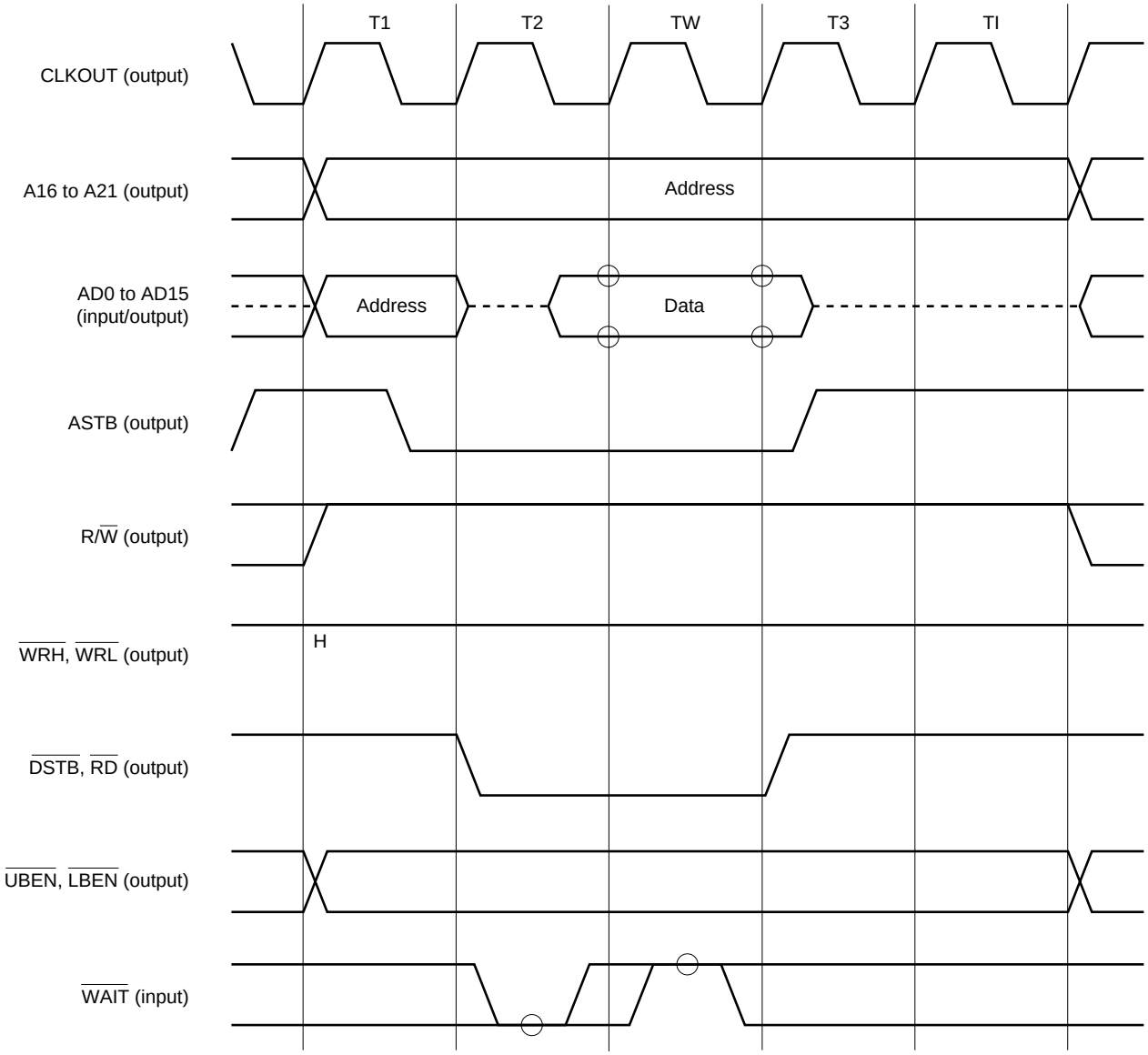
Figure 4-12. Memory Read (0 Wait, Idle State)



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

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

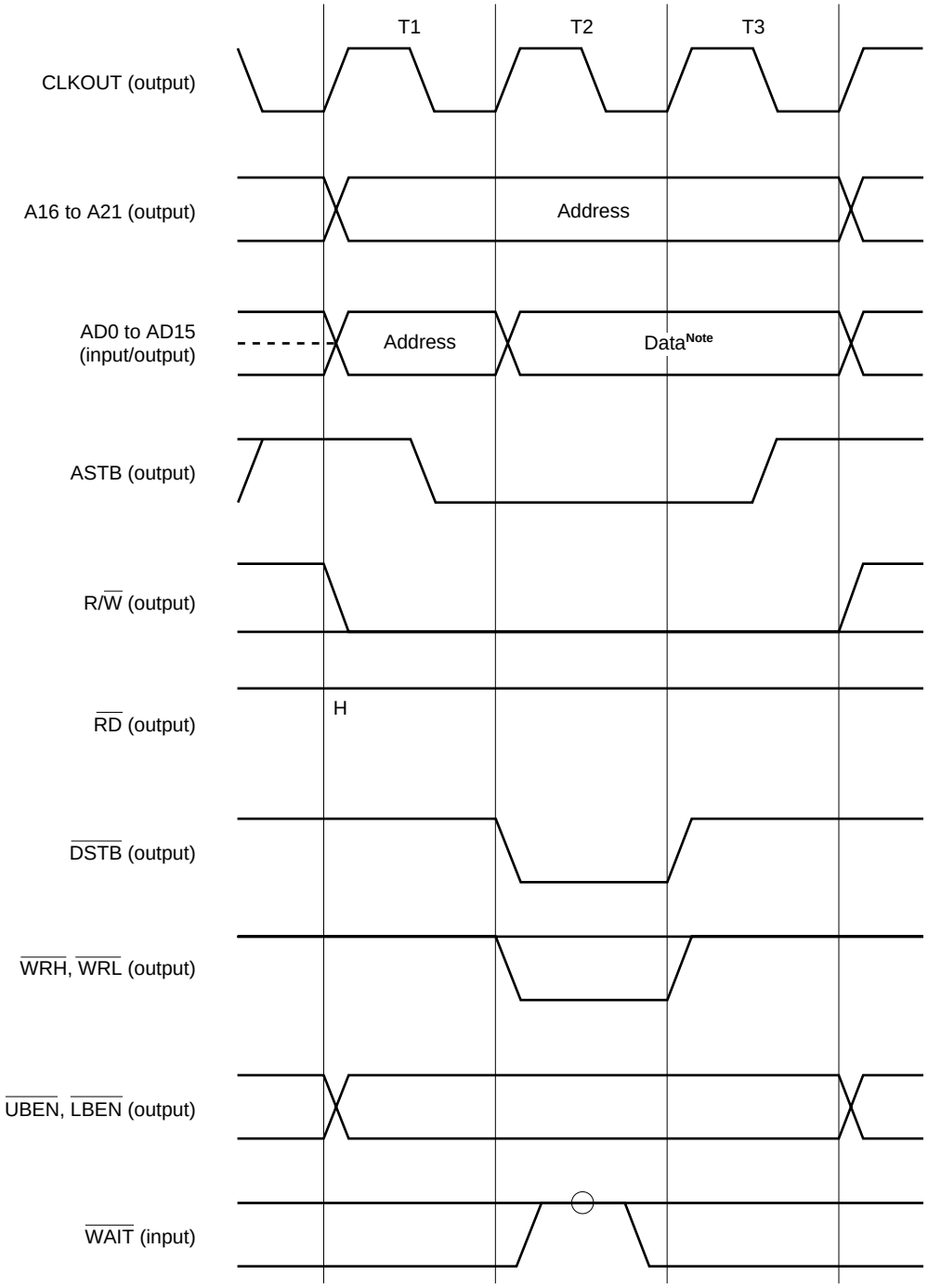
Figure 4-13. Memory Read (1 Wait, Idle State)



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

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

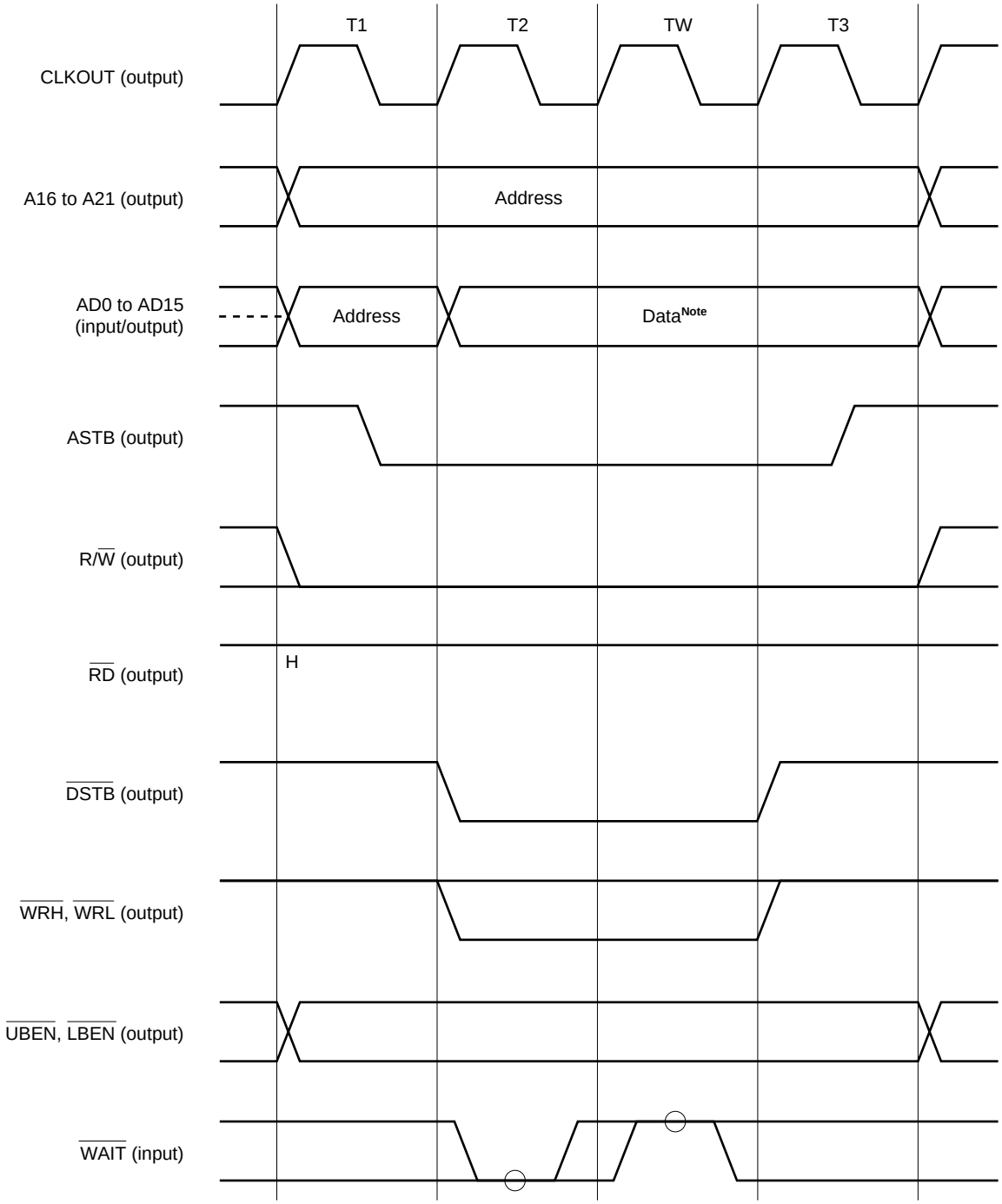
Figure 4-14. Memory Write (0 Wait)



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

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

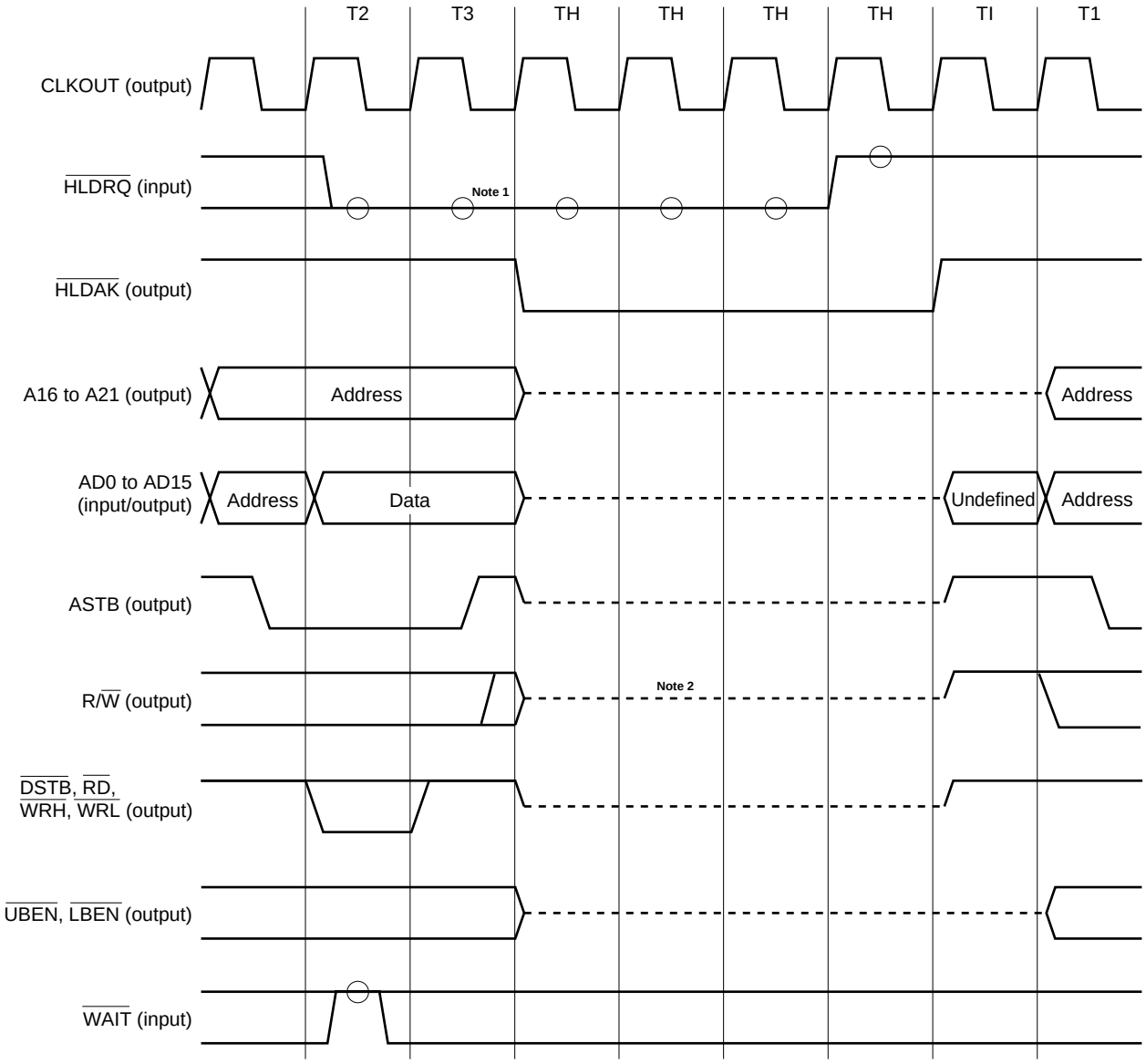
Figure 4-15. Memory Write (1 Wait)



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

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

Figure 4-16. Bus Hold Timing



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

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

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

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

4.9 Bus Priority

There are four external bus cycles: bus hold, 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 low-order half-word access and high-order half-word access of word access operations.

Table 4-3. Bus Priority

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

4.10 Memory Boundary Operation Condition

4.10.1 Program space

(1) Do not execute branch to the on-chip peripheral I/O area or continuous fetch from the internal RAM area to peripheral I/O area. If branch or instruction fetch is executed nevertheless, the NOP instruction code is continuously fetched and not fetched from external memory.
(2) A prefetch operation straddling over the on-chip peripheral I/O area (invalid fetch) does not take place if a branch instruction exists at the upper-limit address of the internal RAM area.

4.10.2 Data space

Only the address aligned at the half-word boundary (when the least significant bit of the address is “0”)/word boundary (when the lowest 2 bits of the address are “0”) boundary is accessed by data half-word (16 bits)/word (32 bits) long.

Therefore, access that straddles over the memory or memory block boundary does not take place.

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

[MEMO]

CHAPTER 5 INTERRUPT/EXCEPTION PROCESSING FUNCTION

5.1 Outline

The V850/SV1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and realizes a high-powered interrupt function that can service interrupt requests from a total of 51 sources for the μPD703039, 703040, 703041, and 70F3040, and a total of 52 sources for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

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

The V850/SV1 can process interrupt requests from the internal 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).

5.1.1 Features

- Interrupts
  - Non-maskable interrupt: 2 sources
  - Maskable interrupt: (the number of maskable interrupt sources is different depending on the product)
    μPD703039, 703040, 703041, 70F3040: 51 sources
    μPD703039Y, 703040Y, 703041Y, 70F3040Y: 52 sources
  - 8 levels programmable priorities
  - Mask specification for the interrupt request according to priority
  - Mask can be specified to each maskable interrupt request.
  - Noise elimination, edge detection, and valid edge of external interrupt request signal can be specified.
- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 1 source (illegal op code exception)

Interrupt/exception sources are listed in Table 5-1.

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

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	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]	TRAP instruction	–	004nH[Note]	00000040H	nextPC	–
	Exception	–	TRAP1n[Note]	TRAP instruction	–	005nH[Note]	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	INTWTNI	Watch timer prescaler	WT	0100H	00000100H	nextPC	WTIIC
		9	INTOV8/ INTTCLR8/ INTTI8	TM8 overflow/ TM8 external clear input/ TM8 external clock input	TM8	0110H	00000110H	nextPC	OVIC8
		10	INTOV9/ INTTI9	TM9 overflow/TM9 external clock input	TM9	0120H	00000120H	nextPC	OVIC9
		11	INTCP80/ INTCM80	CC80 capture input/ CC80 compare match	TM8	0130H	00000130H	nextPC	CC8IC0
		12	INTCP81/ INTCM81	CC81 capture input/ CC81 compare match	TM8	0140H	00000140H	nextPC	CC8IC1
		13	INTCP82/ INTCM82	CC82 capture input/ CC82 compare match	TM8	0150H	00000150H	nextPC	CC8IC2
		14	INTCP83/ INTCM83	CC83 capture input/ CC83 compare match	TM8	0160H	00000160H	nextPC	CC8IC3
		15	INTCP90	CP90 capture input	TM9	0170H	00000170H	nextPC	CP9IC0
		16	INTCP91	CP91 capture input	TM9	0180H	00000180H	nextPC	CP9IC1
		17	INTCP92	CP92 capture input	TM9	0190H	00000190H	nextPC	CP9IC2
		18	INTCP93	CP93 capture input	TM9	01A0H	000001A0H	nextPC	CP9IC3
		19	INTCM90	CM90 compare match	TM9	01B0H	000001B0H	nextPC	CM9IC0
		20	INTCM91	CM91 compare match	TM9	01C0H	000001C0H	nextPC	CM9IC1
		21	INTTM000	INTTM000	TM0	01D0H	000001D0H	nextPC	TMIC000
		22	INTTM001	INTTM001	TM0	01E0H	000001E0H	nextPC	TMIC001
		23	INTTM010	INTTM010	TM1	01F0H	000001F0H	nextPC	TMIC010
		24	INTTM011	INTTM011	TM1	0200H	00000200H	nextPC	TMIC010
		25	INTTM2	TM2 compare match/OVF	TM2	0210H	00000210H	nextPC	TMIC2

Note n: Value of 0 to FH

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

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	26	INTTM3	TM3 compare match/OVF	TM3	0220H	00000220H	nextPC	TMIC3
		27	INTTM4	TM4 compare match/OVF	TM4	0230H	00000230H	nextPC	TMIC4
		28	INTTM5	TM5 compare match/OVF	TM5	0240H	00000240H	nextPC	TMIC5
		29	INTTM6	TM6 compare match/OVF	TM6	0250H	00000250H	nextPC	TMIC6
		30	INTTM7	TM7 compare match/OVF	TM7	0260H	00000260H	nextPC	TMIC7
		31	INTTM10	TM10 compare match/OVF	TM10	0270H	00000270H	nextPC	TMIC10
		32	INTTM11	TM11 compare match/OVF	TM11	0280H	00000280H	nextPC	TMIC11
		33	INTIIC0[Note]/ INTCSI0	I^2C0 interrupt/ CSI0 transmit end	I^2C0/ CSI0	0290H	00000290H	nextPC	CSIC0
		34	INTSER0	UART0 serial error	UART0	02A0H	000002A0H	nextPC	SERIC0
		35	INTSR0/ INTCSI1	UART0 receive end/ CSI1 transmit end	UART0/ CSI1	02B0H	000002B0H	nextPC	CSIC1
		36	INTST0	UART0 transmit end	UART0	02C0H	000002C0H	nextPC	STIC0
		37	INTCSI2	CSI2 transmit end	CSI2	02D0H	000002D0H	nextPC	CSIC2
		38	INTIIC1[Note]	I^2C1 interrupt	I^2C1	02E0H	000002E0H	nextPC	IICIC1
		39	INTSER1	UART1 serial error	UART1	02F0H	000002F0H	nextPC	SERIC1
		40	INTSR1/ INTCSI3	UART1 receive end/ CSI3 transmit end	UART1/ CSI3	0300H	00000300H	nextPC	CSIC3
		41	INTST1	UART1 transmit end	UART1	0310H	00000310H	nextPC	STIC1
		42	INTCSI4	CSI4 transmit end	CSI4	0320H	00000320H	nextPC	CSIC4
		43	INTAD	A/D conversion end	A/D	0330H	00000330H	nextPC	ADIC
		44	INTDMA0	DMA0 transfer end	DMA0	0340H	00000340H	nextPC	DMAIC0
		45	INTDMA1	DMA1 transfer end	DMA1	0350H	00000350H	nextPC	DMAIC1
		46	INTDMA2	DMA2 transfer end	DMA2	0360H	00000360H	nextPC	DMAIC2
		47	INTDMA3	DMA3 transfer end	DMA3	0370H	00000370H	nextPC	DMAIC3
		48	INTDMA4	DMA4 transfer end	DMA4	0380H	00000380H	nextPC	DMAIC4
		49	INTDMA5	DMA5 transfer end	DMA5	0390H	00000390H	nextPC	DMAIC5
		50	INTWTN	Watch timer OVF	WT	03A0H	000003A0H	nextPC	WTNIC
		51	INTKR	Key return interrupt	KR	03B0H	000003B0H	nextPC	KRIC

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

Remarks 1. Default Priority: Priority that takes precedence 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 restored PC saved when an interrupt is granted during the 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. Restored PC of the interrupt/exception other than RESET is the value of the PC (when an event occurred) + 1.

5.2 Non-Maskable Interrupt

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

Non-maskable interrupt of the V850/SV1 are available for the following two requests:

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

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

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

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

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

5.2.1 Operation

If the non-maskable interrupt is generated, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored PC to FEPC.
(2) Saves the current PSW to FEPSW.
(3) Writes exception code 0010H to the higher half-word (FECC) of ECR.
(4) Sets the NP and ID bits of PSW and clears the EP bit.
(5) Loads the handler address (00000010H, 00000020H) of the non-maskable interrupt routine to the PC, and transfers control.

Figure 5-1. Non-Maskable Interrupt Servicing

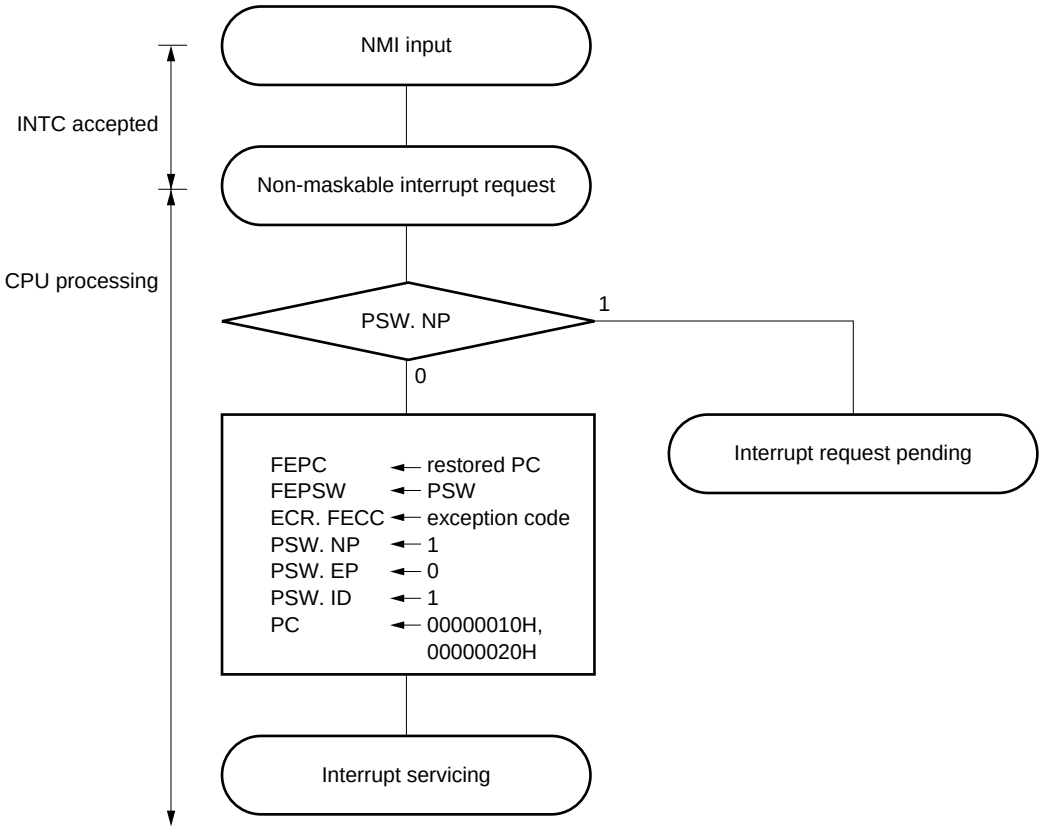
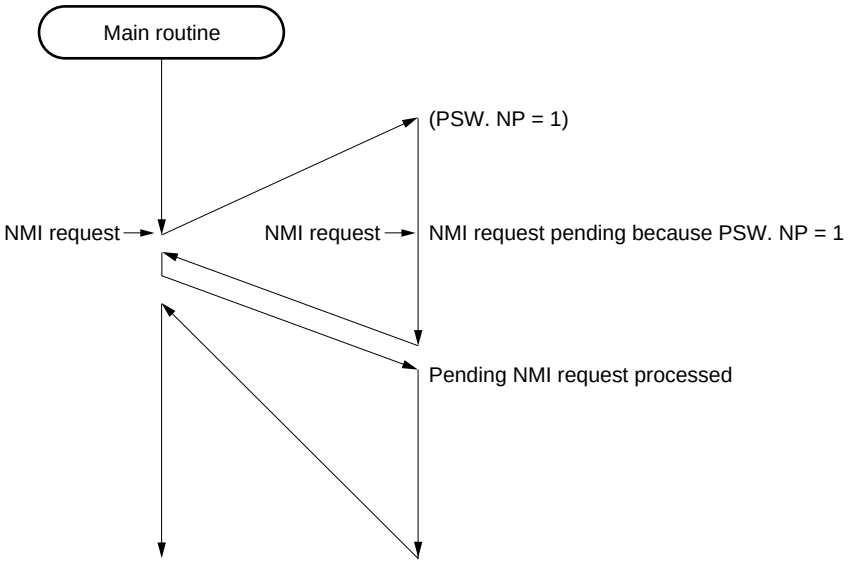
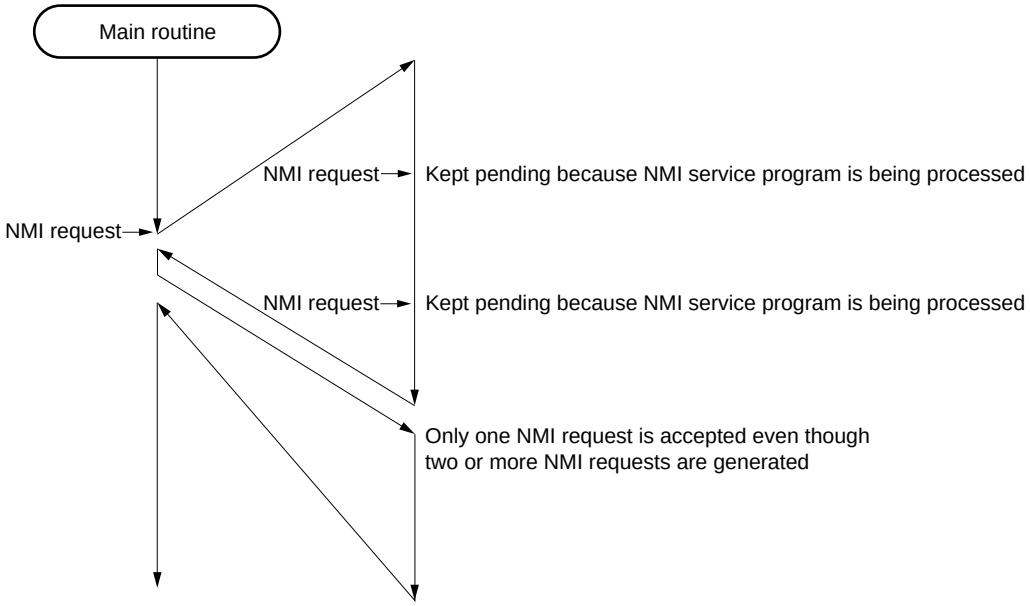


Figure 5-2. Acknowledging Non-Maskable Interrupt Request

(a) If a new NMI request is generated while an NMI service routine is executing:



(b) If a new NMI request is generated twice while an NMI service routine is executing:



5.2.2 Restore

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

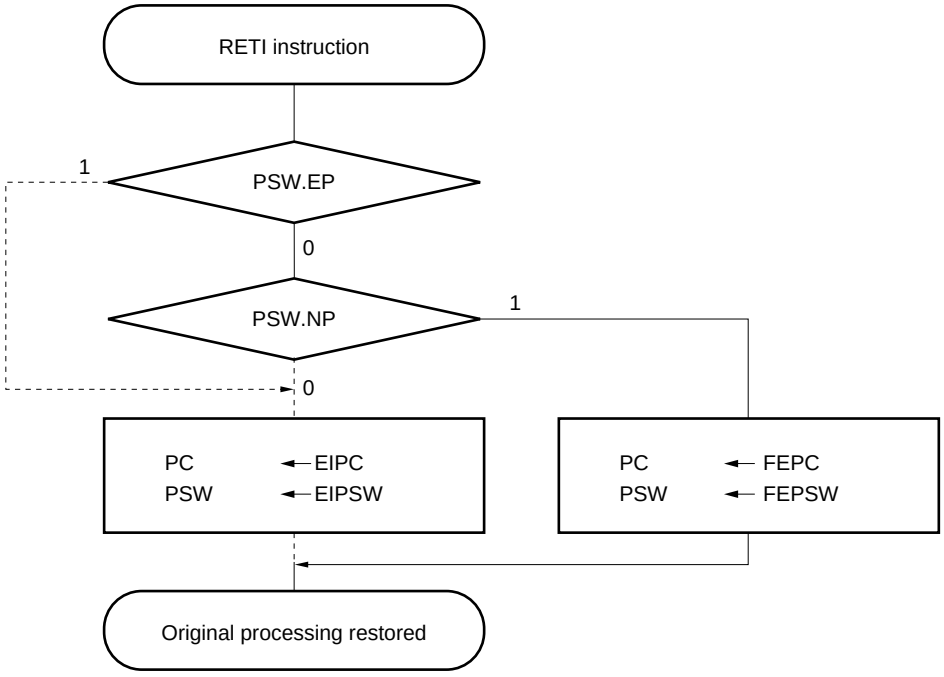
Operation of RETI instruction

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 PC and PSW from FEPC and FEPSW, respectively, because the EP bit of PSW is 0 and the NP bit of PSW is 1.
(2) Transfers control back to the address of the restored PC and PSW.

Figure 5-3 illustrates how the RETI instruction is processed.

Figure 5-3. RETI Instruction Processing



Caution When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during the non-maskable interrupt service, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

5.2.3 NP flag

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

After reset: 00000020H

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

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

5.2.4 Noise elimination circuit of NMI pin

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

NMI pin is used for canceling the software stop mode. In the software stop mode, noise elimination does not use system clock for noise elimination because the internal system clock is stopped.

5.2.5 Edge detection function of NMI pin

The NMI pin valid edge can be selected from the following four types: falling edge, rising edge, both edges, detects neither rising nor falling edge.

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

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

When using P00 as an output port, set the NMI valid edge to “detects neither rising nor falling edge”.

Figure 5-4. Rising Edge Specification Register 0 (EGP0)

After reset: 00H R/W Address: FFFFF0C0H

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

EGP0n	Rising Edge Valid Control
0	No interrupt request signal 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

Figure 5-5. Falling Edge Specification Register 0 (EGN0)

After reset: 00H R/W Address: FFFFF0C2H

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

EGN0n	Falling Edge Valid Control
0	No interrupt request signal 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

5.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850/SV1 has 51 maskable interrupt sources for the μPD703039, 703040, 703041, and 70F3040 and 52 maskable interrupt sources for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers, allowing programmable priority control.

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

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

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

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

5.3.1 Operation

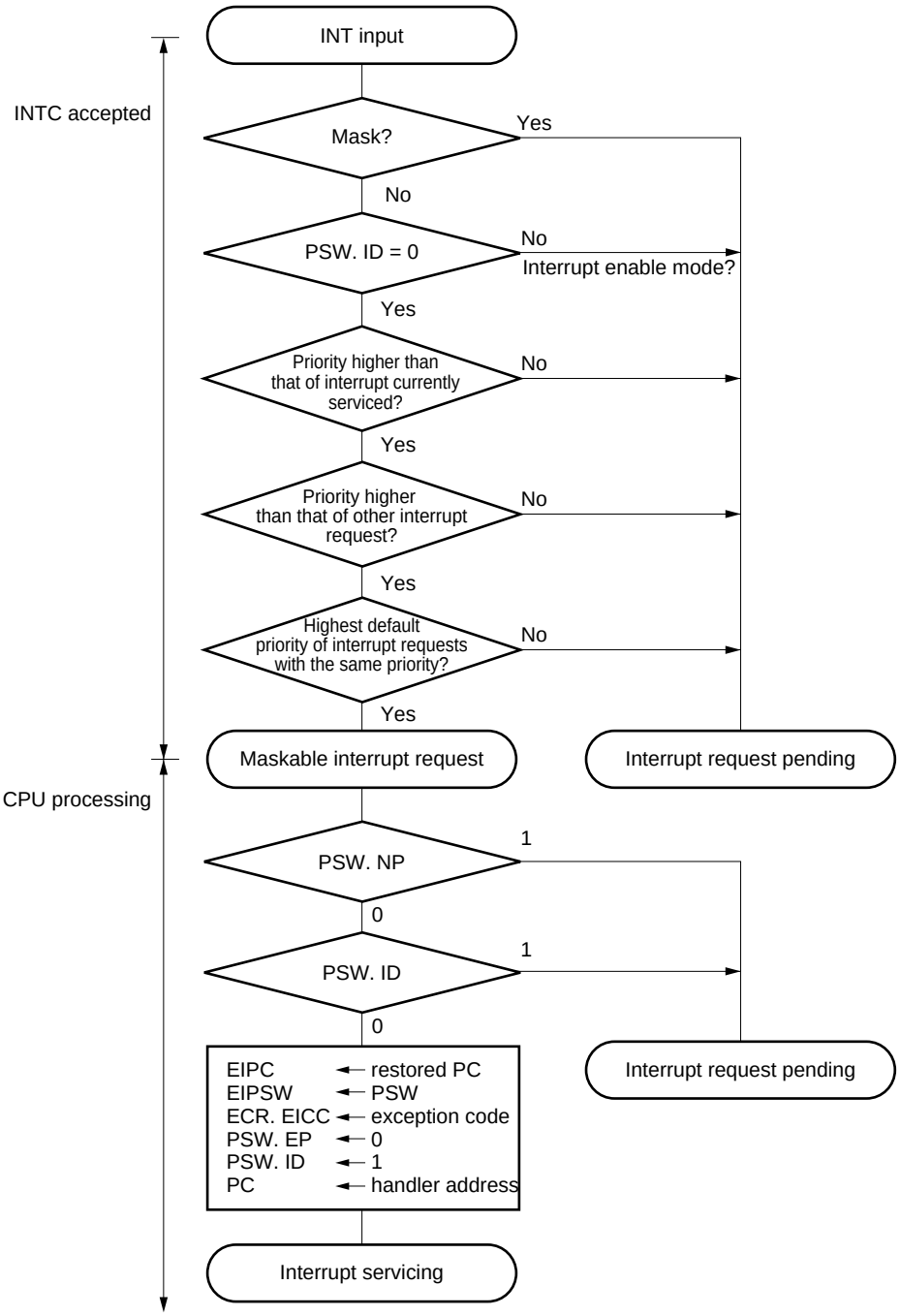
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine:

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower half-word of ECR (EICC).
(4) Sets the ID bit of PSW and clears the EP bit.
(5) Loads the corresponding handler address to the PC, and transfers control.

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

Figure 5-6 illustrates how the maskable interrupts are serviced.

Figure 5-6. Maskable Interrupt Servicing



5.3.2 Restore

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

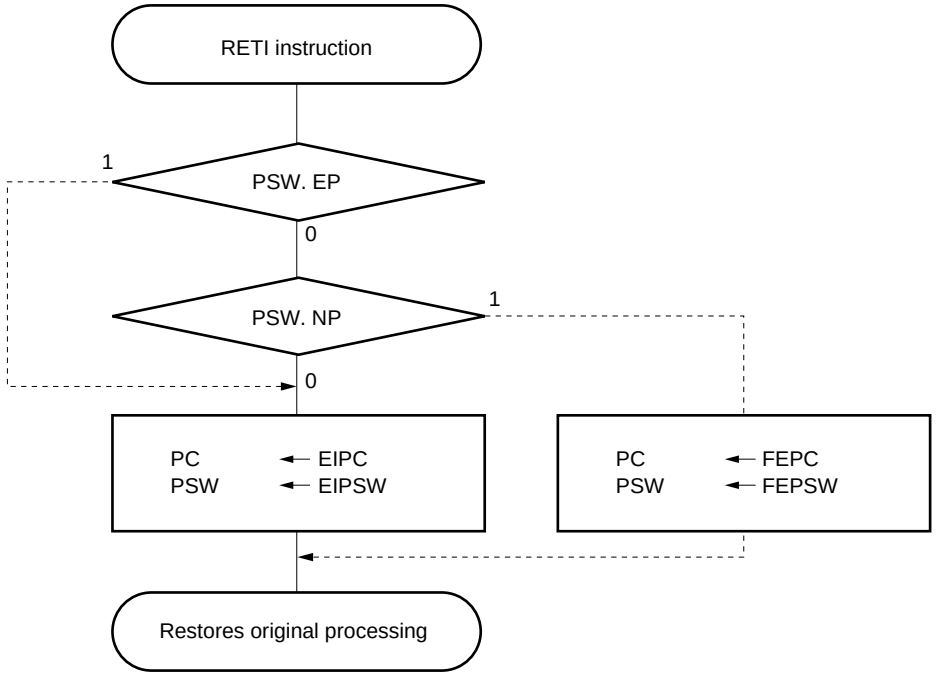
Operation of RETI instruction

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 PC and PSW from EIPC and EIPSW because the EP bit of PSW is 0 and the NP bit of PSW is 0.
(2) Transfers control to the address of the restored PC and PSW.

Figure 5-7 illustrates the processing of the RETI instruction.

Figure 5-7. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the maskable interrupt service, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

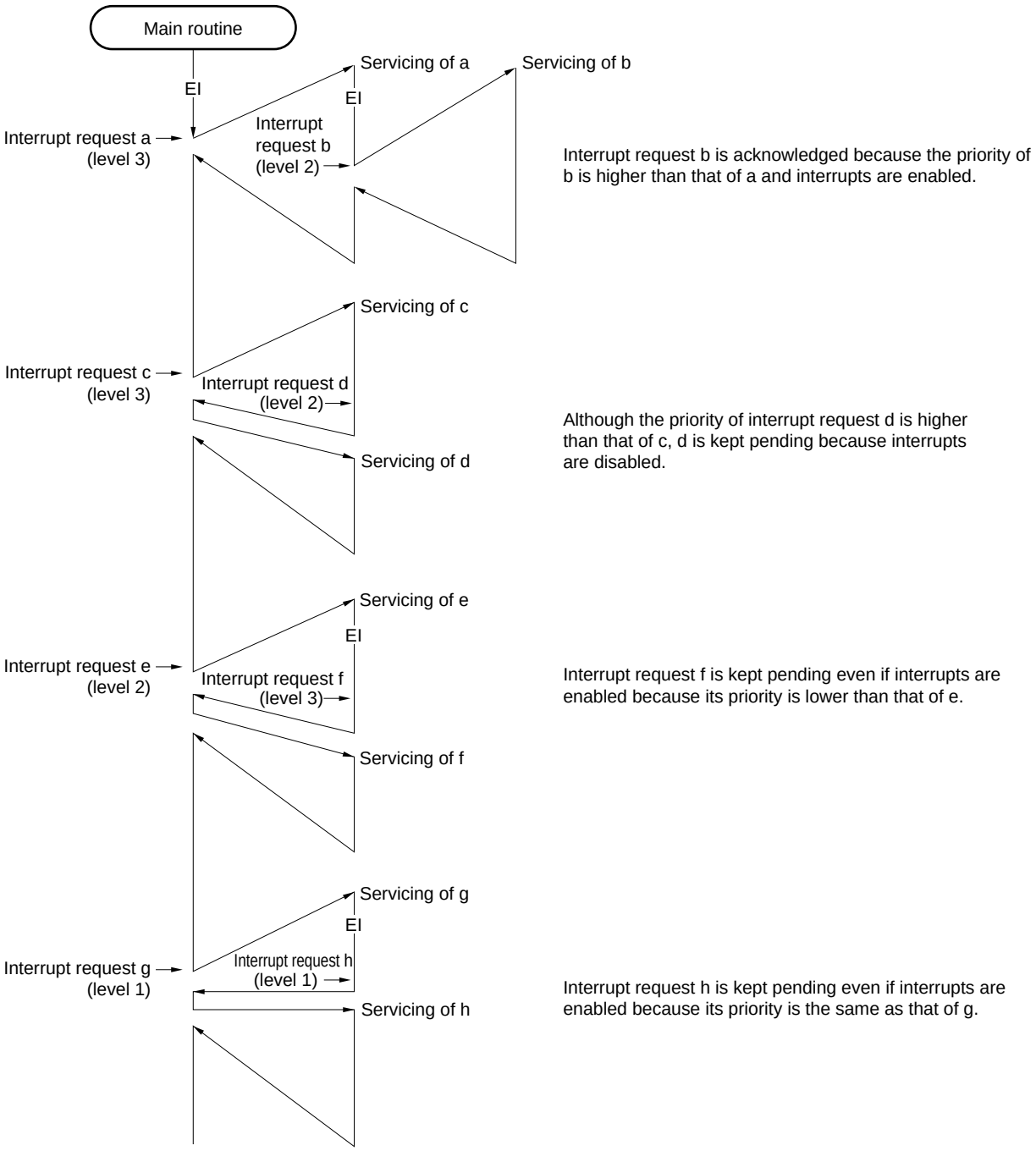
5.3.3 Priorities of maskable interrupts

The V850/SV1 provides multiple interrupt service that acknowledges an interrupt while servicing another interrupt. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels which are specified by 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 types (default priority level) beforehand. For more information, refer to **Table 5-1 Interrupt Source List**. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of PSW is automatically set to ‘‘1’’. Therefore, when multiple interrupts are to be used, clear the ID flag to ‘‘0’’ beforehand (for example, by placing the EI instruction into the interrupt service program) to set the interrupt enable mode.

Figure 5-8. Example of Interrupt Nesting Service (1/2)

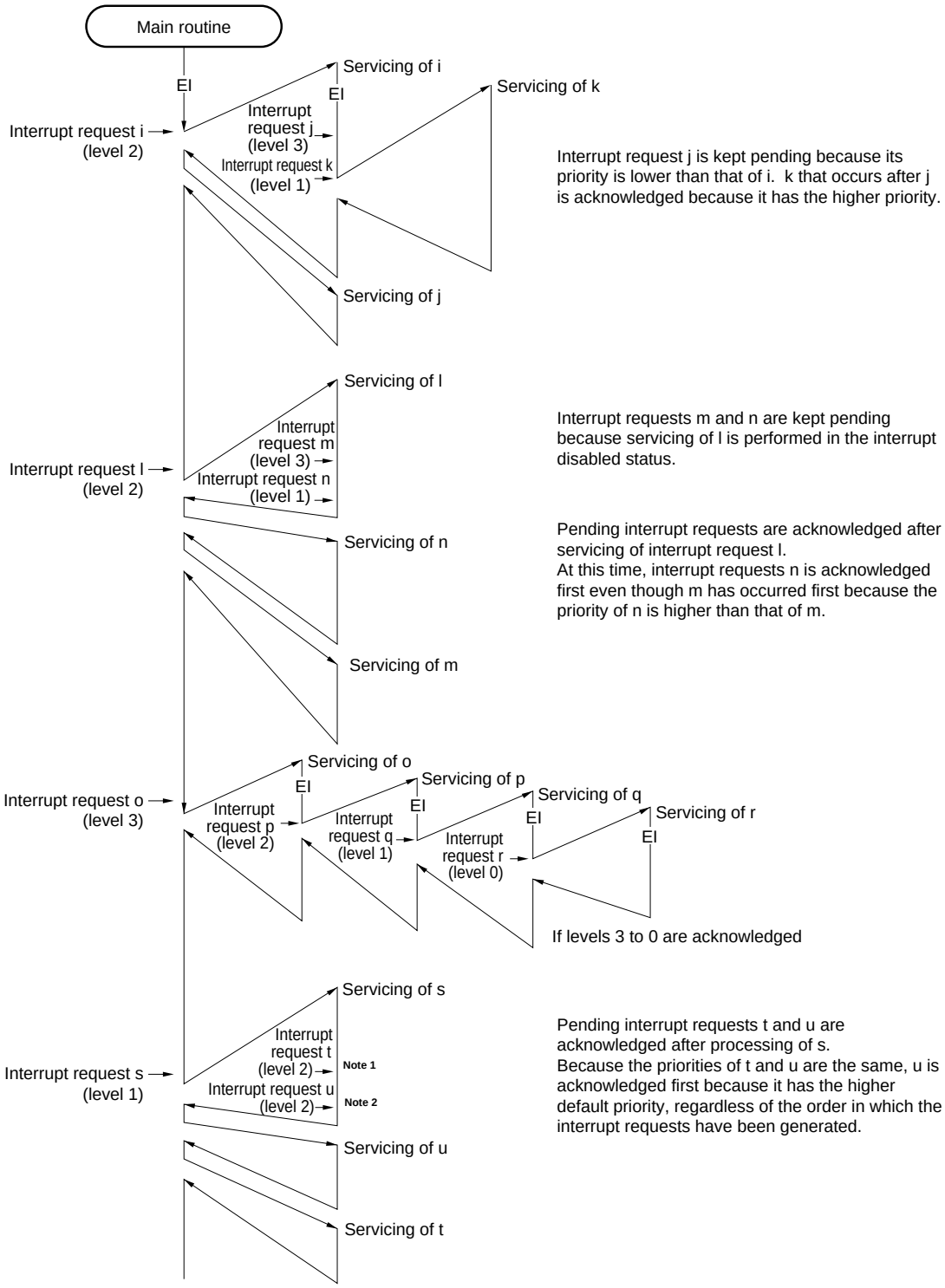


Caution The values of EIPC and EIPSW must be saved before executing multiple interrupts.

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

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

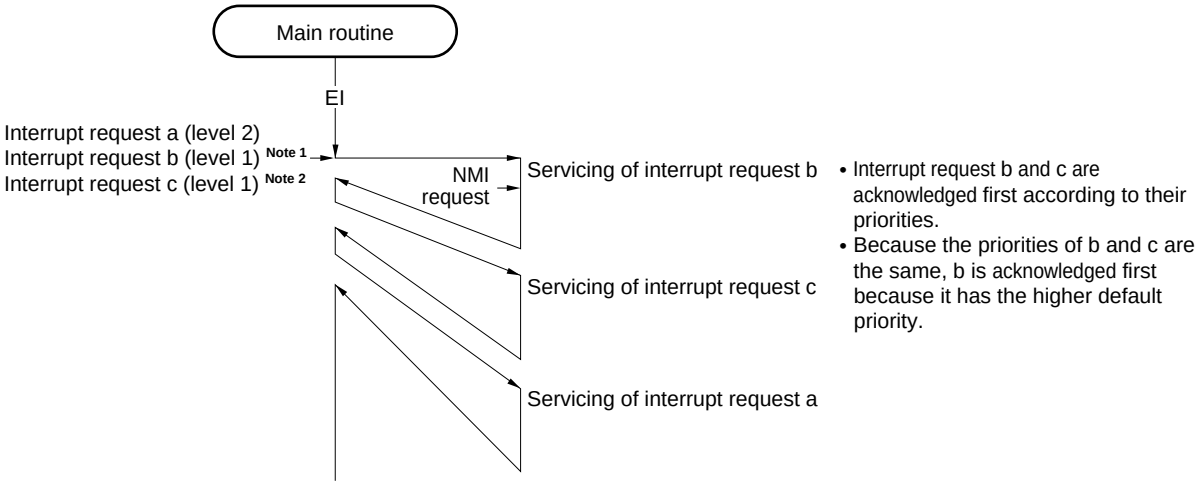
Figure 5-8. Example of Interrupt Nesting Service (2/2)



Notes 1. Lower default priority

2. Higher default priority

Figure 5-9. Example of Servicing Interrupt Requests Simultaneously Generated



Notes 1. Higher default priority

2. Lower default priority

5.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each maskable interrupt and sets the control conditions for each maskable interrupt request.

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

Figure 5-10. Interrupt Control Register (xxICn)

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

Symbol	7	6	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt Request Flag [Note]
0	Interrupt request not generated
1	Interrupt request generated

xxMKn	Interrupt Mask Flag
0	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: Identification name of each peripheral unit (WDT, P, WTNI, OV, CC8, CC9, CM9, TM, CS, SER, ST, AD, DMA, WTN, IIC, or KR)

n: Peripheral unit number (Refer to **Table 5-2**).

Address and bit of each interrupt control register is as follows:

Table 5-2. Interrupt Control Register (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	WTNIIC	WTNIIF	WTNIMK	0	0	0	WTNIPR2	WTNIPR1	WTNIPR0
FFFFF112H	OVIC8	OVIF8	OVMK8	0	0	0	OVPR82	OVPR81	OVPR80
FFFFF114H	OVIC9	OVIF9	OVMK9	0	0	0	OVPR92	OVPR91	OVPR90
FFFFF116H	CC8IC0	CC8IF0	CC8MK0	0	0	0	CC8PR02	CC8PR01	CC8PR00
FFFFF118H	CC8IC1	CC8IF1	CC8MK1	0	0	0	CC8PR12	CC8PR11	CC8PR10
FFFFF11AH	CC8IC2	CC8IF2	CC8MK2	0	0	0	CC8PR22	CC8PR21	CC8PR20
FFFFF11CH	CC8IC3	CC8IF3	CC8MK3	0	0	0	CC8PR32	CC8PR31	CC8PR30
FFFFF11EH	CP9IC0	CP9IF0	CP9MK0	0	0	0	CP9PR02	CP9PR01	CP9PR00
FFFFF120H	CP9IC1	CP9IF1	CP9MK1	0	0	0	CP9PR12	CP9PR11	CP9PR10
FFFFF122H	CP9IC2	CP9IF2	CP9MK2	0	0	0	CP9PR22	CP9PR21	CP9PR20
FFFFF124H	CP9IC3	CP9IF3	CP9MK3	0	0	0	CP9PR32	CP9PR31	CP9PR30
FFFFF126H	CM9IC0	CM9IF0	CM9MK0	0	0	0	CM9PR02	CM9PR01	CM9PR00
FFFFF128H	CM9IC1	CM9IF1	CM9MK1	0	0	0	CM9PR12	CM9PR11	CM9PR10
FFFFF12AH	TMIC000	TMIF000	TMMK000	0	0	0	TMPR0002	TMPR0001	TMPR0000
FFFFF12CH	TMIC001	TMIF001	TMMK001	0	0	0	TMPR0012	TMPR0011	TMPR0010
FFFFF12EH	TMIC010	TMIF010	TMMK010	0	0	0	TMPR0102	TMPR0101	TMPR0100
FFFFF130H	TMIC011	TMIF011	TMMK011	0	0	0	TMPR0112	TMPR0111	TMPR0110
FFFFF132H	TMIC2	TMIF2	TMMK2	0	0	0	TMPR22	TMPR21	TMPR20
FFFFF134H	TMIC3	TMIF3	TMMK3	0	0	0	TMPR32	TMPR31	TMPR30
FFFFF136H	TMIC4	TMIF4	TMMK4	0	0	0	TMPR42	TMPR41	TMPR40
FFFFF138H	TMIC5	TMIF5	TMMK5	0	0	0	TMPR52	TMPR51	TMPR50
FFFFF13AH	TMIC6	TMIF6	TMMK6	0	0	0	TMPR62	TMPR61	TMPR60
FFFFF13CH	TMIC7	TMIF7	TMMK7	0	0	0	TMPR72	TMPR71	TMPR70
FFFFF13EH	TMIC10	TMIF10	TMMK10	0	0	0	TMPR102	TMPR101	TMPR100
FFFFF140H	TMIC11	TMIF11	TMMK11	0	0	0	TMPR112	TMPR111	TMPR110
FFFFF142H	CSIC0	CSIF0	CSMK0	0	0	0	CSPR02	CSPR01	CSPR00
FFFFF144H	SERIC0	SERIF0	SERMK0	0	0	0	SERPR02	SERPR01	SERPR00
FFFFF146H	CSIC1	CSIF1	CSMK1	0	0	0	CSPR12	CSPR11	CSPR10

Table 5-2. Interrupt Control Register (xxICn) (2/2)

Address	Register	Bit							
		7	6	5	4	3	2	1	0
FFFFF148H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF14AH	CSIC2	CSIF2	CSMK2	0	0	0	CSPR22	CSPR21	CSPR20
FFFFF14CH	IICIC1[Note]	IICIF1	IICMK1	0	0	0	IICPR12	IICPR11	IICPR10
FFFFF14EH	SERIC1	SERIF1	SERMK1	0	0	0	SERPR12	SERPR11	SERPR10
FFFFF150H	CSIC3	CSIF3	CSMK3	0	0	0	CSPR32	CSPR31	CSPR30
FFFFF152H	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF154H	CSIC4	CSIF4	CSMK4	0	0	0	CSPR42	CSPR41	CSPR40
FFFFF156H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF158H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF15AH	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF15CH	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF15EH	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF160H	DMAIC4	DMAIF4	DMAMK4	0	0	0	DMAPR42	DMAPR41	DMAPR40
FFFFF162H	DMAIC5	DMAIF5	DMAMK5	0	0	0	DMAPR52	DMAPR51	DMAPR50
FFFFF164H	WTNIC	WTNIF	DTNMK	0	0	0	WTNPR2	WTNPR1	WTNPR0
FFFFF168H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

5.3.5 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the set (1) bit, in the ISPR register, corresponding to the interrupt request having the highest priority is automatically reset to 0 by hardware. However, it is not reset when execution is returned from non-maskable processing or exception processing.

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

Figure 5-11. In-Service Priority Register (ISPR)

After reset: 00H R Address: FFFFF166H

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

ISPRn	Indicates Priority of Interrupt Currently Acknowledged
0	Interrupt request with priority n not acknowledged
1	Interrupt request with priority n acknowledged

Remark n: 0 to 7 (priority level)

5.3.6 Maskable interrupt status flag

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

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

Note Interrupt disable flag (ID) function

It is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupt and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware.

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

5.3.7 Watchdog timer mode register (WDTM)

Read/write is available in 8- or 1-bit units (for details, refer to **CHAPTER 9 WATCHDOG TIMER**).

Figure 5-12. Watchdog Timer Mode Register (WDTM)

After reset: 00H R/W Address: FFFFF384H

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

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

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

Caution If 1 is set to RUN or WDTM4 bit, no operation other than the reset input is available for clearing this register.

5.3.8 Noise elimination

(1) Noise elimination of INTP0 to INTP3 pins

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

(2) Noise elimination of INTP4 and INTP5 pins

INTP4 and INTP5 pins incorporate the digital noise elimination circuit. If an input level of the INTP pin is detected with the sampling clock (f_{xx}) and the same level is not detected three successive times, the input pulse is eliminated as a noise. Note the followings:

- In the case that the input pulse width is between 2 and 3 clocks, whether the input pulse is detected as a valid edge or eliminated as a noise is indefinite.
- To securely detect the level as a pulse, 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) Noise elimination of INTP6 pin

The INTP6 pin incorporates a digital noise elimination circuit. 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 digital noise elimination clock for the INTP6 pin.

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

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

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

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

Figure 5-13. 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	Noise Elimination Clock	Reliably Eliminated Noise Width[Note]		
				f_{XX} = 20 MHz	f_{XX} = 10 MHz	f_{XX} = 2 MHz
0	0	0	f_{XX}	100 ns	200 ns	1 μs
0	0	1	$f_{XX}/64$	6.4 μs	12.8 μs	64 μs
0	1	0	$f_{XX}/128$	12.8 μs	25.6 μs	128 μs
0	1	1	$f_{XX}/256$	25.6 μs	51.2 μs	256 μs
1	0	0	$f_{XX}/512$	51.2 μs	102.4 μs	512 μs
1	0	1	$f_{XX}/1024$	102.4 μs	204.8 μs	1 ms
1	1	0	Setting prohibited			
1	1	1	f_{XT}	61 μs		

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

5.3.9 Edge detection function

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

- Rising edge
- Falling edge
- Both rising and falling edges
- Detects neither rising nor falling edge

The validity of the rising edge is controlled by the rising edge specification register (EGP0), and the validity of the falling edge is controlled by the falling edge specification register (EGN0). Refer to **Figures 5-4** and **5-5** for details of EGP0 and EGN0.

After reset, the valid edges of the INTP0 to INTP6 pins are set to the “detects neither rising nor falling edge” state. Therefore, the INTP0 to INTP6 pins function as a normal port and an interrupt request cannot be acknowledged, unless a valid edge is specified by using the EGP0 and EGN0 registers.

When using P01 to P07 as output ports, set valid edges of INTP0 to INTP6 to “detects neither rising nor falling edge” or mask the interrupt request.

5.4 Software Exception

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

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

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

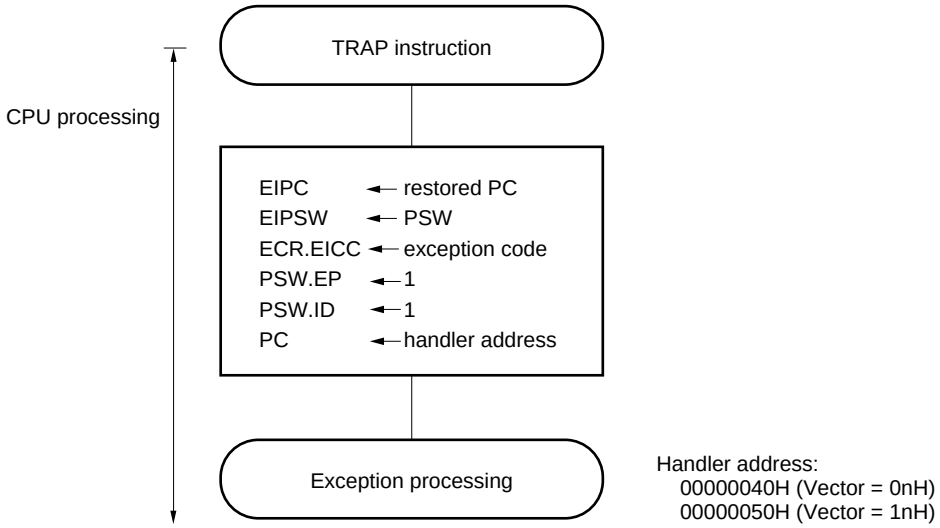
5.4.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
(4) Sets the EP and ID bits of PSW.
(5) Loads the handler address (00000040H or 00000050H) of the software exception routine in the PC, and transfers control.

Figure 5-14 illustrates how a software exception is processed.

Figure 5-14. Software Exception Processing



5.4.2 Restore

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

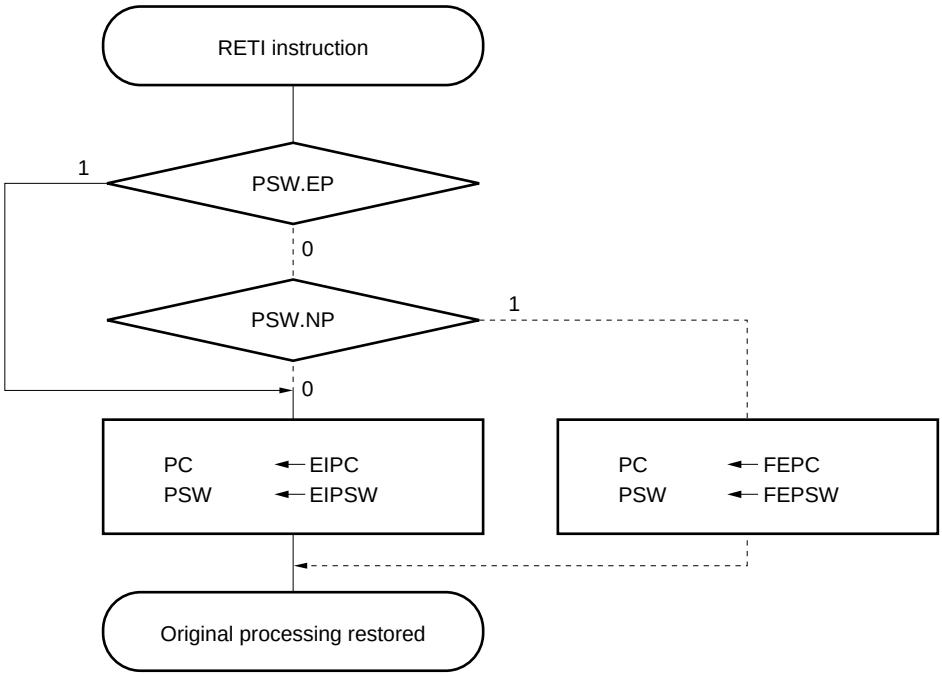
Operation of RETI instruction

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

(1) Restores the restored PC and PSW from EIPC and EIPSW because the EP bit of PSW is 1.
(2) Transfers control to the address of the restored PC and PSW.

Figure 5-15 illustrates the processing of the RETI instruction.

Figure 5-15. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception process, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

5.4.3 EP flag

The EP flag is a status flag used to indicate that exception processing is in progress. It is set when on exception occurs, and the interrupt is disabled.

After reset: 00000020H

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

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

5.5 Exception Trap

The exception trap is an interrupt that is requested when illegal execution of an instruction takes place. In the V850/SV1, 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.

5.5.1 Illegal op code definition

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

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

x : don't care

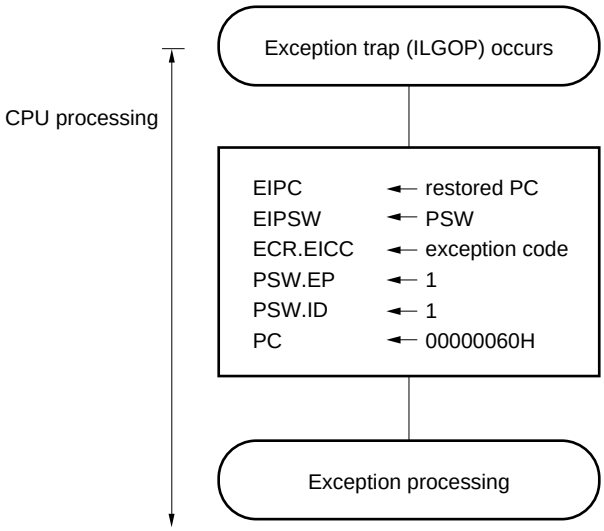
5.5.2 Operation

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

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

Figure 5-16 illustrates how the exception trap is processed.

Figure 5-16. Exception Trap Processing



5.5.3 Restore

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

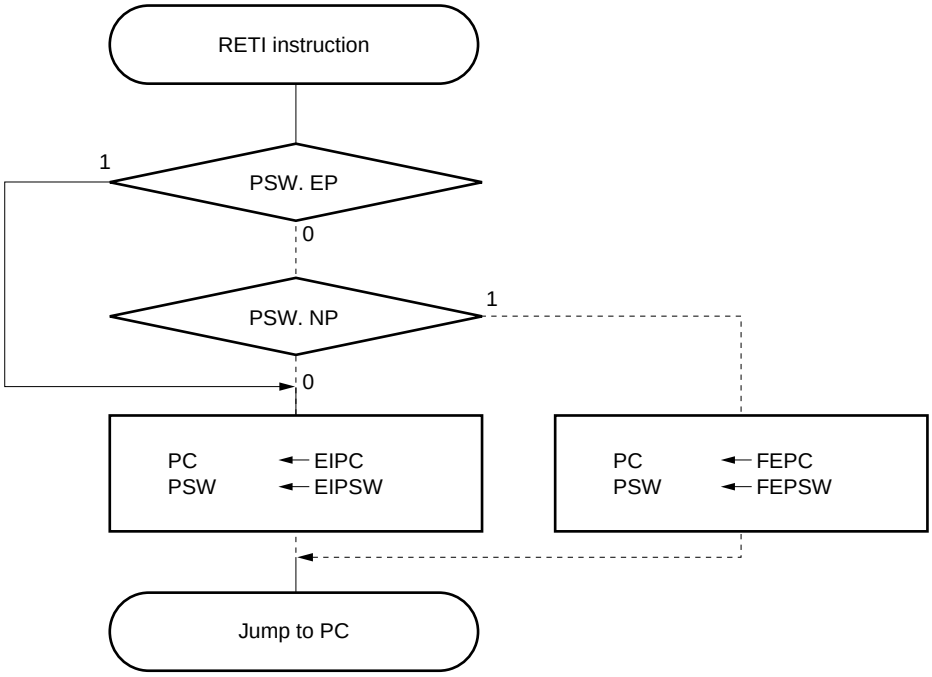
Operation of RETI instruction

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

(1) Restores the restored PC and PSW from EIPC and EIPSW because the EP bit of PSW is 1.
(2) Transfers control to the address of the restored PC and PSW.

Figure 5-17 illustrates the processing of the RETI instruction.

Figure 5-17. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the exception trap process, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

5.6 Priority Control

5.6.1 Priorities of interrupts and exceptions

	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.

5.6.2 Multiple interrupt processing

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

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

Multiple interrupt servicing control is performed when it is in the state of interrupt acknowledgement (ID = 0). Even in an interrupt servicing routine, this control must be set in the state of acknowledgement (ID = 0). If a maskable interrupt acknowledgement or exception is generated during a service program of maskable interrupt or exception, EIPC and EIPSW must be saved.

The following example shows the procedure of interrupt nesting.

(1) To acknowledge maskable interrupts in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• Saves EIPC to memory or register
• Saves EIPSW to memory or register
• EI instruction (enables interrupt acknowledge-
  ment)
        ...                                    ← Acknowledges interrupt such as INTP input.
        ...
• DI instruction (disables interrupt acknowledge-
  ment)
• Restores saved value to EIPSW
• Restores saved value to EIPC
• RETI instruction
```

(2) To generate exception in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• Saves EIPC to memory or register
• Saves EIPSW to memory or register
• EI instruction (enables interrupt acknowledge-
  ment)
        ...
• TRAP instruction                     ← Acknowledges exception such as TRAP instruction.
• Illegal op code                      ← Acknowledges exception such as illegal op code.
        ...
• Restores saved value to EIPSW
• Restores saved value to EIPC
• RETI instruction
```

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

Priorities of maskable interrupts

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

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

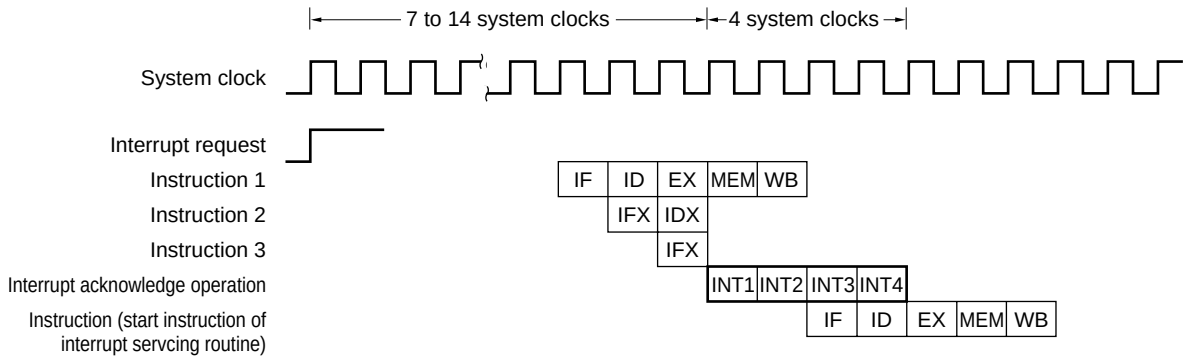
A pending interrupt request is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

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

5.7 Interrupt Latency Time

The following table describes the V850/SV1 interrupt latency time (from interrupt request generation to start of interrupt servicing).

Figure 5-18. Pipeline Operation at Interrupt Request Acknowledge



INT1 to INT4: Interrupt acknowledge processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt Latency Time (System Clock)			Condition
	Internal Interrupt	External Interrupt	
Minimum	11	13	These times are valid except the following: • 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	

5.8 Periods Where Interrupt Is Not Acknowledged

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

Interrupt request non-sample instruction

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

5.9 Key Interrupt Function

Key interrupt (INTKR) can be generated by inputting a falling edge to key input pins (KR0 to KR7) by means of 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.

Figure 5-19. Key Return Mode Register (KRM)

After reset: 00H R/W Address: FFFFF3D0H

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

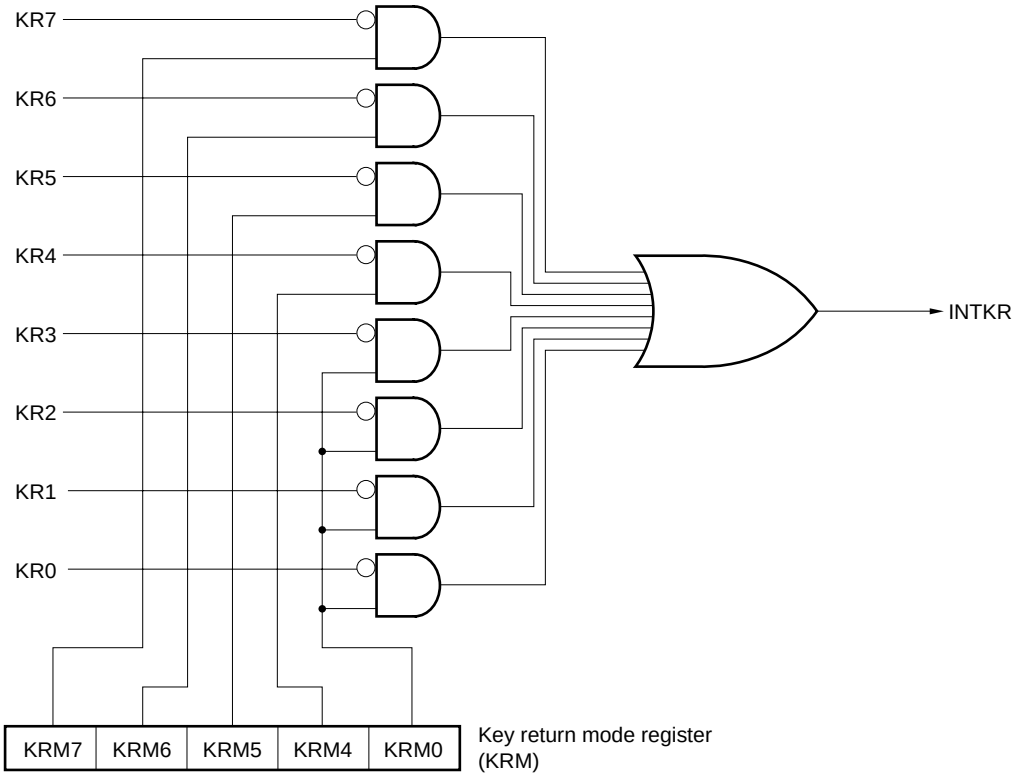
KRMn	Key Return Mode Control
0	Does not detect key return signal
1	Detects key return signal

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

Table 5-3. Description of Key Return Detection Pin

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

Figure 5-20. Key Return Function



CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Outline

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

(1) Main system clock oscillator

This oscillator has an oscillation frequency of 4 to 20 MHz. Oscillation can be stopped by executing a STOP instruction or by setting the processor clock control register (PCC). Oscillation is also stopped during a reset.

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

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

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

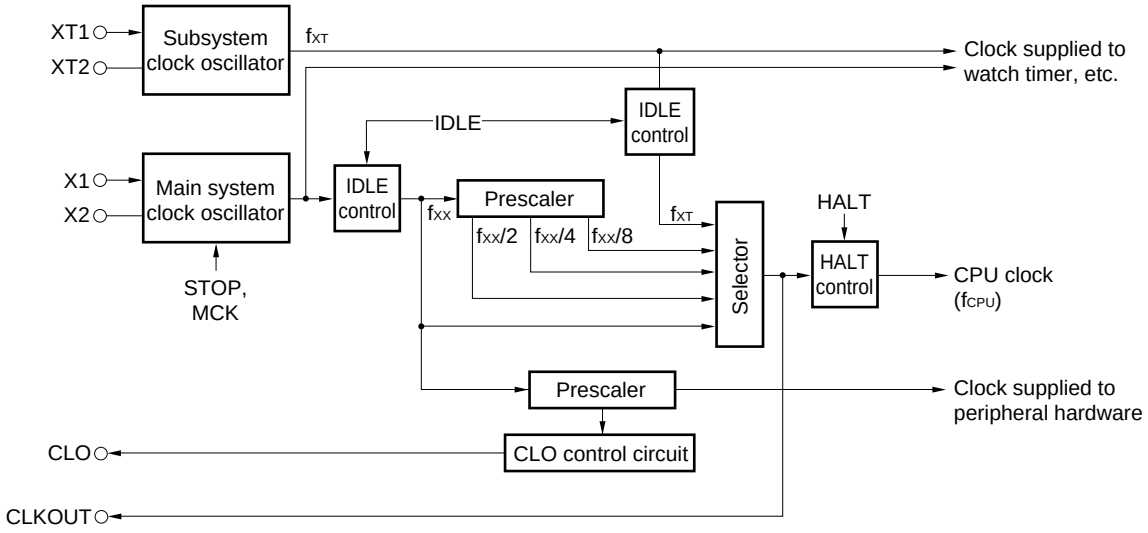
(2) Subsystem clock oscillator

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

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

6.2 Configuration

Figure 6-1. Clock Generation Function



Remark f_{XX}: Main system clock frequency

f_{XT}: Subsystem clock frequency

6.3 Clock Output Function

6.3.1 CLKOUT signal

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.

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

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

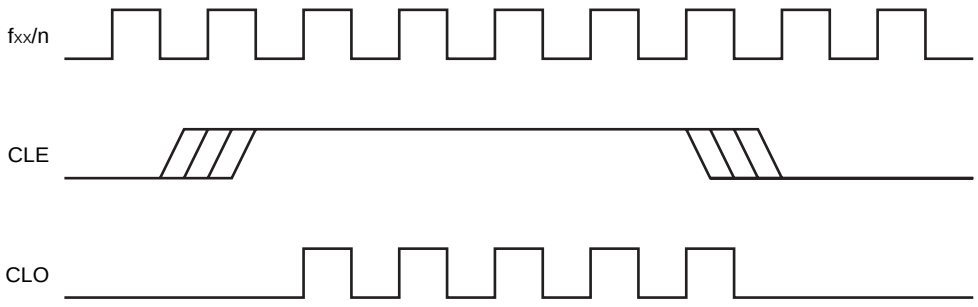
6.3.2 CLO signal

The FS1 and FS0 bits of the clock output mode register (CLOM) can be used to select the clock frequency ($f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, or $f_{XX}/16$) to be output to the CLO pin.

When the CLE bit is set to 1, the CLO signal synchronized with the clocks is output immediately. When the CLE bit is cleared to 0, the CLO signal stops and the signal goes low.

In HALT mode, the CLO signal continues to be output. In IDLE or STOP mode, the signal stops and the previous level is held.

Figure 6-2. CLO Signal Timing



Remarks 1. f_{XX}: Main system clock frequency

2. n = 2, 4, 8, 16

6.4 Control Registers

6.4.1 Processor clock control register (PCC)

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

Figure 6-3. Processor Clock Control Register (PCC)

After reset: 03H R/W Address: FFFFF074H

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

MCK	Operation of Main Clock (Main System Clock)
0	Operate
1	Stop

CK2[Note]	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} (sub clock)

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

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

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

3. Always set bits 7 and 5 to 3 all to 0.

Remark X: don't care

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

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

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

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

(b) Example of sub clock operation → main clock operation setup

<1> MCK ← 0: Main clock oscillation start

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

<3> CK2, CK1, CK0 ← CPU clock

<4> Main clock operation: If CK1 and CK0 are not changed from value of the CPU clock selected before the sub clock operation in <3>, a maximum of two instructions is required.
If CK1 and CK0 are changed, a maximum of ten instructions is required

6.4.2 Clock output mode register (CLOM)

This register can be read/written in 8- or 1-bit units.
However, the FS0 and FS1 bits should not be manipulated in 1-bit units.

Figure 6-4. Clock Output Mode Register (CLOM)

After reset: 00H R/W Address: FFFFF38AH

	7	6	5	4	3	2	1	0
CLOM	0	0	0	CLE	0	0	FS1	FS0

CLE	Operation of CLO Signal Clock Output
0	Output disabled (low level output)
1	CLO signal output

FS1	FS0	Selection of CPU Clock
0	0	$f_{XX}/2$
0	1	$f_{XX}/4$
1	0	$f_{XX}/8$
1	1	$f_{XX}/16$

While the CLO signal is output by setting the CLE bit to 1, the cycle of the CLO signal, set by the FS0 and FS1 bits, should not be changed. Also, changing of the FS0 and FS1 bits should not be performed at the same time as when the CLE bit is changed.

6.4.3 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, refer to **3.4.9 Specific registers**.

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

Figure 6-5. Power Save Control Register (PSC)

After reset: C0H R/W Address: FFFFF070H

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

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

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

STP	STOP Mode Setting
0	Normal mode
1	STOP mode[Note 2]

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

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

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

6.4.4 Oscillation stabilization time select register (OSTS)

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

Figure 6-6. Oscillation Stabilization Time Select Register (OSTS)

After reset: 04H R/W Address: FFFFF380H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time[Note]
0	0	0	$2^{14}/f_{XX}$ (819.2 μs)
0	0	1	$2^{16}/f_{XX}$ (3.3 ms)
0	1	0	$2^{17}/f_{XX}$ (6.6 ms)
0	1	1	$2^{18}/f_{XX}$ (13.1 ms)
1	0	0	$2^{19}/f_{XX}$ (26.2 ms)
Other than above			Setting prohibited

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

6.5 Power Save Functions

6.5.1 Outline

This product provides the following power saving functions.

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

(1) HALT mode

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

A 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's operating clock as well as the operating clock for on-chip peripheral functions while the clock oscillator is still operating. However, the sub clock 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 power save control register (PSC)'s IDLE bit is set to 1, the system switches to IDLE mode.

(3) Software STOP mode

This mode stops the entire system by stopping a clock oscillator that is not for a sub clock system. The sub clock continues to be supplied to keep on-chip peripheral functions operating. If a sub clock 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 sub clock.

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

(4) Sub clock operation

Under this mode, the CPU clock is set to operate using the sub clock and the PCC register's MCK bit is set to 1 to set low power consumption mode in which the entire system operates using only the sub clock.

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

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

6.5.2 HALT mode

(1) Settings and operating states

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

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

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

HALT Mode Setting / Item		When CPU Operates with Main Clock		When CPU Operates with Sub Clock	
		When Sub Clock Does Not Exist	When Sub Clock Exists	When Main Clock's Oscillation Continues	When Main Clock's Oscillation Is Stopped
CPU		Stopped			
ROM correction		Stopped			
Clock generator		Oscillation for main clock and sub clock Clock supply to CPU is stopped			
CLKOUT		Operating			
CLO		Operating			Stopped
24-bit timer (TM8, TM9)		Operating			Stopped
16-bit timer (TM0)		Operating			Operates when INTWTN is selected as count clock (f_{XT} is selected as watch timer)
16-bit timer (TM1)		Operating			Stopped
8-bit timer (TM2, TM3)		Operating			Stopped
8-bit timer (TM4, TM5)		Operating			Operates when f_{XT} is selected as count clock
8-bit timer (TM6, TM7)		Operating			Operates when TO0 is selected as count clock (only when TM0 is operating)
8-bit timer (TM10, TM11)		Operating			Stopped
PWM output (PWM0 to PWM3)		Operating			Stopped
Watch timer		Operates when main clock is selected as count clock	Operating		Operates when f_{XT} is selected as count clock
Watchdog timer		Operating (interval timer only)			
Serial interface	CSI0 to CSI3	Operating			Operates when an external clock is selected as the serial clock
	I^2C0[Note], I^2C1[Note]	Operating			Stopped
	UART0, UART1	Operating			Operates when an external clock is selected as the baud rate clock (only for transmission)
	CSI4	Operating			Operates when an external clock is selected as the serial clock

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

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

HALT Mode Setting / Item		When CPU Operates with Main Clock		When CPU Operates with Sub Clock	
		When Sub Clock Does Not Exist	When Sub Clock Exists	When Main Clock's Oscillation Continues	When Main Clock's Oscillation Is Stopped
A/D converter		Operating			Stopped
Vsync/Hsync separator		Operating			Stopped
DMA0 to DMA5		Operating			
Real-time output		Operating			
Port function		Held			
External bus interface		Only bus hold function operates			
External interrupt request	NMI	Operating			
	INTP0 to INTP3	Operating			
	INTP4, INTP5	Operating			Stopped
	INTP6				Operates when f_{XT} is selected as the noise elimination circuit
Key return function		Operating			
In external expansion mode	AD0 to AD15	High impedance[Note]			
	A16 to A21				
	$\overline{LBEN}$, $\overline{UBEN}$	Held[Note] (high impedance when $\overline{HLDAK}$ = 0)			
	R/$\overline{W}$	High level output[Note] (high impedance when $\overline{HLDAK}$ = 0)			
	$\overline{DSTB}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{RD}$				
	ASTB				
	$\overline{HLDAK}$	Operating			

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

(2) Cancellation of HALT mode

HALT mode can be canceled by an NMI request, an unmasked maskable interrupt request, or a $\overline{\text{RESET}}$ pin 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) Only HALT mode is canceled when an interrupt request that has a lower priority level than the interrupt currently being serviced occurs, and the lower-priority interrupt request is not acknowledged. The interrupt request itself is retained.

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

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

This is the same as for normal reset operations.

6.5.3 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 on-chip sub clock 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 function are stopped (except for peripheral functions that are operating with the sub clock). External bus hold requests ($\overline{\text{HLDRQ}}$) are not acknowledged.

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

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

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

IDLE Mode Settings / Item	When Sub Clock Exists	When Sub Clock Does Not Exist
CPU	Stopped	
ROM correction	Stopped	
Clock generator	Oscillation for a main clock and sub clock Clock supply to CPU and on-chip peripheral functions is stopped	
CLKOUT	Stopped	
CLO	Stopped	
24-bit timer (TM8, TM9)	Stopped	
16-bit timer (TM0)	Operates when INTWTN is selected as count clock (f_{XT} is selected as watch timer)	Stopped
16-bit timer (TM1)	Stopped	

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

Item \ IDLE Mode Settings		When Sub Clock Exists	When Sub Clock Does Not Exist
8-bit timer (TM2, TM3)		Stopped	
8-bit timer (TM4, TM5)		Operates when f_{XT} is selected as count clock	Stopped
8-bit timer (TM6, TM7)		Operates when TO0 is selected as count clock (only when TM0 is operating)	Stopped
8-bit timer (TM10, TM11)		Stopped	
PWM output (PWM0 to PWM3)		Stopped	
Watch timer		Operating	
Watchdog timer		Stopped	
Serial interface	CSI0 to CSI3	Operates when an external clock is selected as the serial clock	
	I^2C0 Note, I^2C1 Note	Stopped	
	UART0, UART1	Operates when an external clock is selected as the baud rate clock (only for transmission)	
	CSI4	Operates when an external clock is selected as the serial clock	
A/D converter		Stopped	
Vsync/Hsync separator		Stopped	
DMA0 to DMA5		Stopped	
Real-time output		Operates when INTTM4 or INTTM6 is selected (when TM4 or TM6 is operating)	Stopped
Port function		Held	
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3	Operating	
	INTP4, INTP5	Stopped	
	INTP6	Operates when f_{XT} is selected as sampling clock	Stopped
Key return function		Operating	
In external expansion mode	AD0 to AD15	High impedance	
	A16 to A21		
	$\overline{LBEN}$, $\overline{UBEN}$		
	R/$\overline{W}$		
	$\overline{DSTB}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{RD}$		
	ASTB		
	$\overline{HLDAK}$		

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

(2) Cancellation of IDLE mode

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

6.5.4 Software STOP mode

(1) Settings and operating states

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

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. In addition, on-chip peripheral functions stop operating.

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 sub clock has been selected as the CPU clock.

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

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

STOP Mode Settings / Item		When Sub Clock Exists	When Sub Clock Does Not Exist
CPU		Stopped	
ROM correction		Stopped	
Clock generator		Oscillation for main clock and sub clock Clock supply to CPU and on-chip peripheral functions is stopped	
CLKOUT		Stopped	
CLO		Stopped	
24-bit timer (TM8, TM9)		Stopped	
16-bit timer (TM0)		Operates when INTWTN is selected as count clock (f_{XT} is selected as watch timer)	Stopped
16-bit timer (TM1)		Stopped	
8-bit timer (TM2, TM3)		Stopped	
8-bit timer (TM4, TM5)		Operates when f_{XT} is selected as count clock	Stopped
8-bit timer (TM6, TM7)		Operates when TO0 is selected as count clock (when TM0 is operating)	Stopped
8-bit timer (TM10, TM11)		Stopped	
Watch timer		Operates when f_{XT} is selected as count clock	Stopped (operation prohibited)
Watchdog timer		Stopped	
Serial interface	CSI0 to CSI3	Operates when an external clock is selected as the serial clock	
	I^2C0[Note], I^2C1[Note]	Stopped	
	UART0, UART1	Operates when an external clock is selected as the baud rate clock (only for transmission)	
	CSI4	Operates when an external clock is selected as the serial clock	
A/D converter		Stopped	
Vsync/Hsync separator		Stopped	
DMA0 to DMA5		Stopped	

Note Available only for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

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

STOP Mode Settings / Item		When Sub Clock Exists	When Sub Clock Does Not Exist
Real-time output		Operates when INTTM4 or INTTM6 has been selected (when TM4 or TM6 is operating)	Stopped
Port function		Held	
External bus interface		Stopped	
External interrupt request	NMI	Operating	
	INTP0 to INTP3	Operating	
	INTP4, INTP5	Stopped	
	INTP6	Operates when f_{XT} is selected as sampling clock	Stopped
Key return function		Operating	
In external expansion mode	AD0 to AD15	High impedance	
	A16 to A21		
	$\overline{LBEN}$, $\overline{UBEN}$		
	R/$\overline{W}$		
	$\overline{DSTB}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{RD}$		
	ASTB		
	$\overline{HLDAK}$		

(2) Cancellation of software STOP mode

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

When the STOP mode is canceled, an oscillation stabilization time is secured.

6.6 Oscillation Stabilization Time

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

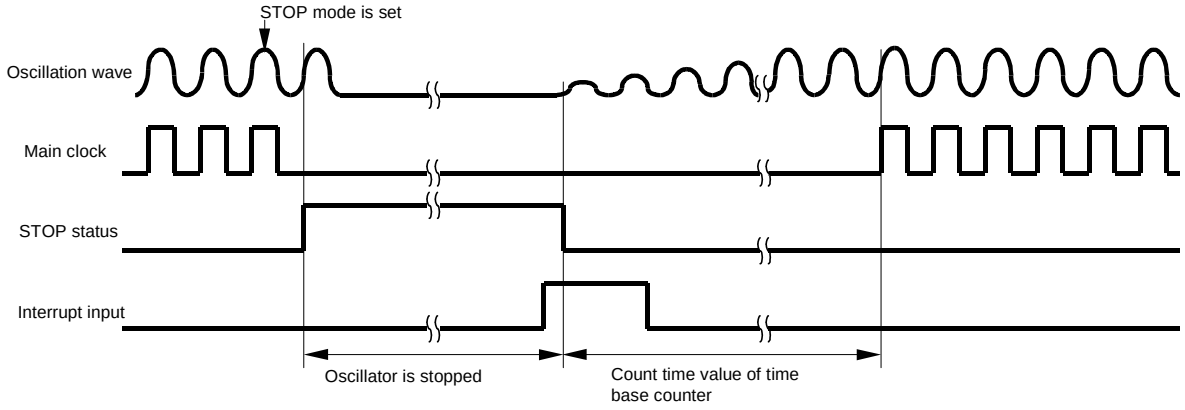
(1) Cancellation by non-maskable interrupt or by unmasked maskable interrupt request

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

Oscillation stabilization time ≒ WDT count time

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

Figure 6-7. 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 17 RESET FUNCTION**.

[MEMO]

CHAPTER 7 TIMER/COUNTER FUNCTION

7.1 24-Bit Timer (TM8 and TM9)

7.1.1 Outline

- 24-bit timer (TM8)
  - 24-bit timer/event counter (1-channel)
  - Capture/compare shared registers: 4 (CC80 to CC83)
  - Usable as a trigger of the A/D converter (CC83 match)
  - Set/reset outputs: 2 (TO80 and TO81)
  - Clear and start of timer
  - Capable of measuring external input pulse
  - Overflow interrupt request and overflow flag
  - Application: Measurement of pulse width and frequency, and pulse output of various forms.
- 24-bit timer (TM9)
  - 24-bit timer/event counter (1-channel)
  - Capture registers: 4 (CP90 to CP93)
  - Compare registers: 2 (CM90 and CM91)
  - INTCP90 to INTCP93 edge detection circuit with 1 to 64/1 to 128 frequency divider
  - Usable as trigger of the real-time output port (CM90 and CM91 match)
  - Overflow interrupt request and overflow flag
  - Clear and start of timer
  - Application: Measurement of pulse width and frequency of software servo, etc.

7.1.2 Differences in operation between the 24-bit timers of the V850/SV1 and the V854™

Table 7-1. Differences between TM0 of V854 and TM8 of V850/SV1

Parameter	V854	V850/SV1
Register name	TM0, CC00 to CC03, TMC00 to TMC02, TOC0	TM8, CC80 to CC83, TMC80 to TMC82, TOC8
Capture trigger/capture interrupt	INTP00 to INTP03	INTCP80 to INTCP83
Compare match interrupt	INTCC00 to INTCC03	INTCM80 to INTCM83
External clock clear input/interrupt	TCLR0/INTP04, TI0/INTP05	TCLR8/INTTCLR8, TI8/INTTI8
Timer output	TO00, TO01	TO80, TO81
Overflow interrupt	INTOV0	INTOV8
System clock	Φ	f_{XX}
16-bit access function	Provided (TM8L, CC0nL: n = 0 to 3)	Not provided
Timer read operation	During read, timer is not stopped	During read, timer is stopped (count clock is masked)
Valid edge specification of external capture/external clock/external clear	Specified by INTTM1 and INTTM2 (Default: falling edge is valid)	Specified by EGP2 and EGN2 (Default: edge is not valid)
Conflict operation of capture register read and capture timing	The value before the capture or after the capture is read	The read value is undefined. The normal value is captured in the capture register (countermeasures such as reading twice are required)

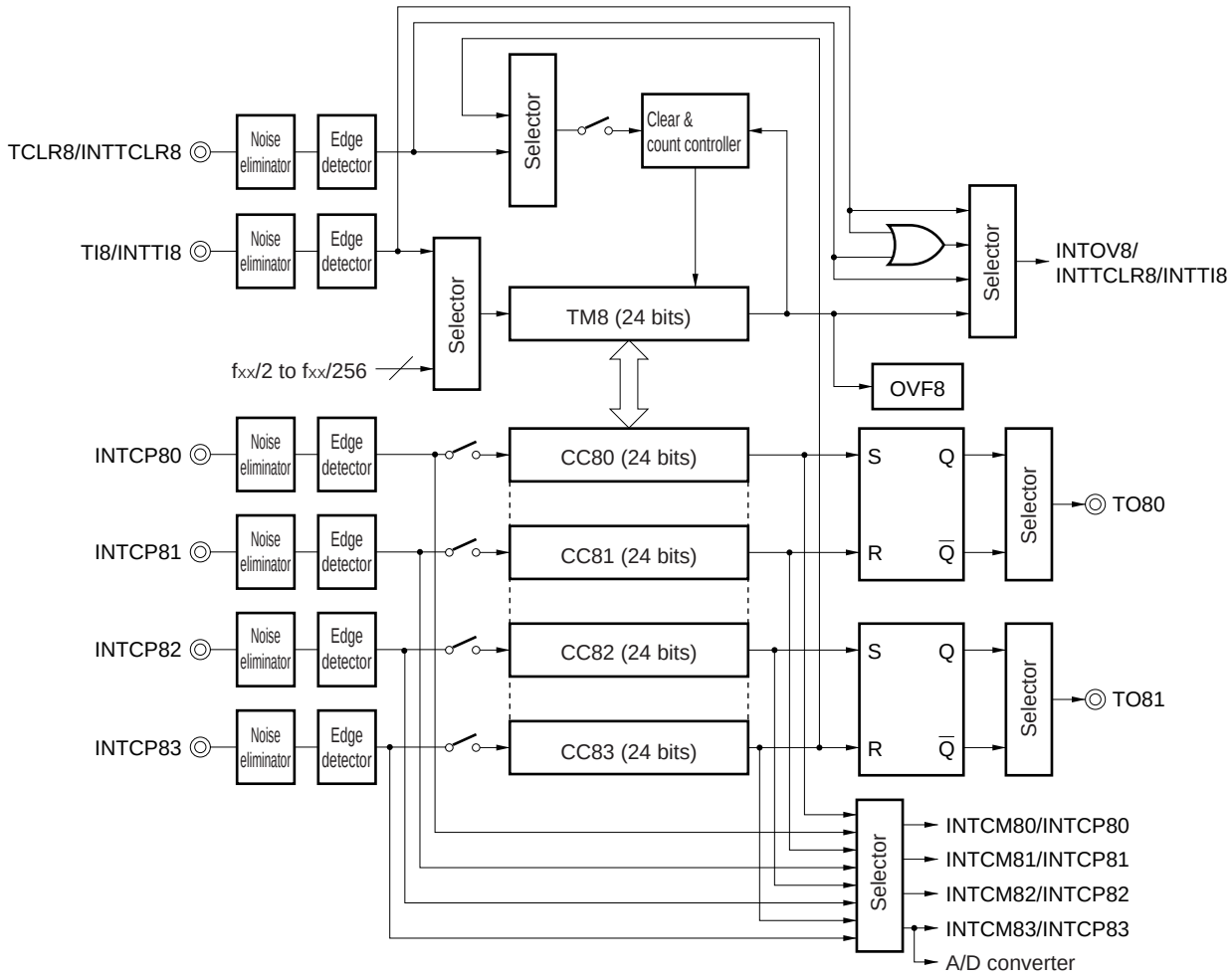
Table 7-2. Differences between TM1 of V854 and TM9 of V850/SV1

Parameter	V854	V850/SV1
Register name	TM1, CP10 to CP13, CM10 to CM11, TMC1	TM9, CP90 to CP93, CM90, CM91, TMC90, TMC91 (Newly added)
Capture trigger/capture interrupt	INTP10 to INTP13	INTCP90 to INTCP93
Compare match interrupt	INTCM10, INTCM11	INTCM90, INTCM91
External clock clear input/interrupt	TI1/INTP14	TI9/INTTI9
Overflow interrupt	INTOV1	INTOV9
System clock	Φ	f_{XX}
16-bit access function	Provided (TM9L, CP1nL, CM1nL: n = 0, 1)	Not provided
Timer read operation	During read, timer is not stopped	During read, timer is stopped (count clock is masked)
Valid edge specification of external capture/external clock/external clear	Specified by INTTM2 and INTTM3 (Default: falling edge is valid)	Specified by EGP3 and EGN3 (Default: edge is not valid)
Conflict operation of capture register read and capture timing	The value before the capture or after the capture is read	The read value is undefined. The normal value is captured in the capture register (countermeasures such as reading twice are required)
CPn1 capture trigger source (n = 1, 9)	INTP11 only	INTCP91 and Vsync signal are selectable

7.1.3 Function

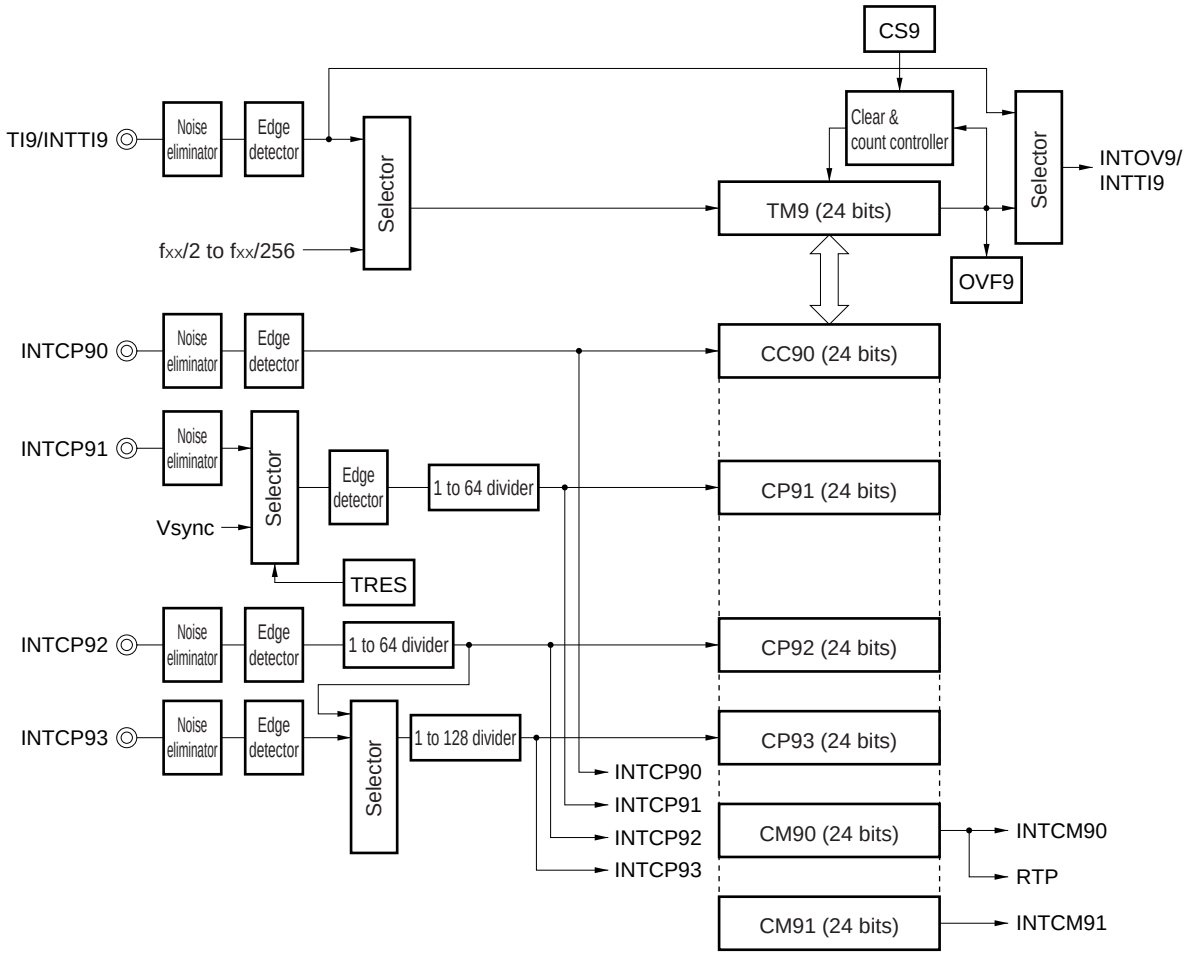
Figures 7-1 and 7-2 show block diagrams of the 24-bit timers.

Figure 7-1. Block Diagram of TM8



Remark fxx: Main system clock frequency

Figure 7-2. Block Diagram of TM9



Remark fxx: Main system clock frequency

7.1.4 Configuration

24-bit timers 8 and 9 consist of the following hardware:

Table 7-3. Configuration of Timers 8 and 9

Item	Configuration
Timer registers	24 bits × 2 (TM8 and TM9)
Registers	Capture/compare registers: 24 bits × 4 (CC8n) Compare registers: 24 bits × 2 (CM90 and CM91) Capture registers: 24 bits × 4 (CP9n)
Timer outputs	2 (TO80 and TO81)
Control registers	24-bit timer mode control register 8m (TMC8m) 24-bit timer mode control registers 90 and 91 (TMC90 and TMC91) 24-bit timer output control register 8 (TOC8) Timer overflow status register (TOVS)

Remark n = 4, m = 3

(1) Timer 8

(a) Timer 8 (TM8)

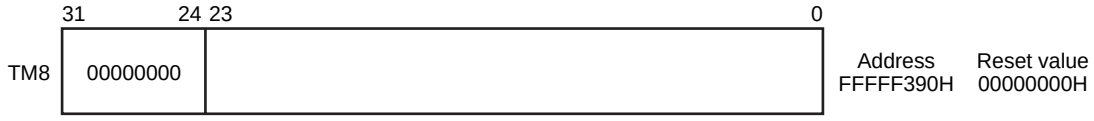
Timer 8 functions as a 24-bit interval timer or external signal event counter. This timer is mainly used for measuring a cycle or frequency. It can also be used for pulse output.

When this timer is accessed in units of 32 bits, 0 is stored in the eight high-order bits.

TM8 performs the counting up of internal count clocks. Start/stop of TM8 is controlled by the CE8 bit of 24-bit timer mode control register 80 (TMC80).

TM8 can be read in units of 32 bits only.

Figure 7-3. Timer 8 (TM8)



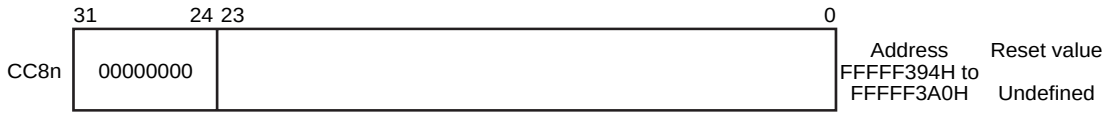
(b) Capture/compare registers 80 to 83 (CC80 to CC83)

The capture/compare registers are 24-bit registers connected to TM8. Each of these registers can be used as a capture or compare register, as specified in 24-bit timer mode control register 81 (TMC81).

When each register is accessed in units of 32 bits, 0 is stored in the eight high-order bits.

CC8n can be read and written in units of 32 bits. The values written to bits 24 to 31 are ignored.

Figure 7-4. Capture/Compare Registers 80 to 83 (CC80 to CC83)



Remark n = 0 to 3

(i) Setting CC8n as a capture register

When CC8n is set as a capture register, the valid edge of the corresponding external interrupt request INTCP8n (n = 0 to 3) is detected as the capture trigger. Timer 8 latches the count value (capture operation) in synchronization with the capture trigger. The latched value is retained in the capture register until the next capture operation is performed.

(ii) Setting CC8n as a compare register

When CC8n is set as a compare register, the timer value is compared with the register value every timer count clock. When these values coincide, an interrupt request is generated.

The compare register has a set/reset output function. This function sets or resets the corresponding timer output in synchronization with the generation of the coincidence signal.

(2) Timer 9

(a) Timer 9 (TM9)

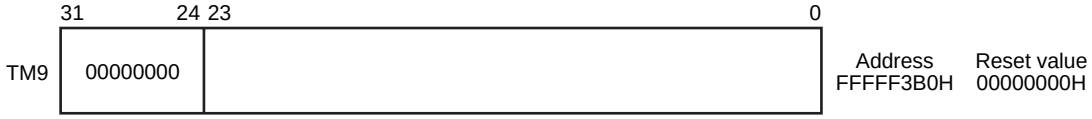
TM9 functions as a 24-bit free-running timer or external signal event counter. This timer is mainly used for measuring a cycle or frequency. It can also be used as a pulse output.

When this timer is accessed in units of 32 bits, 0 is stored in the eight high-order bits.

TM9 performs the counting up of internal count clocks. Start/stop of TM9 is controlled by the CE9 bit of 24-bit timer mode control register 90 (TMC90).

TM9 can be read in units of 32 bits only.

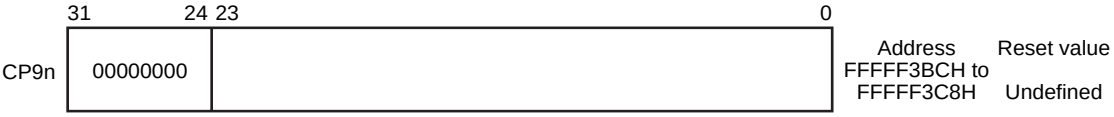
Figure 7-5. Timer 9 (TM9)



(b) Capture registers 90 to 93 (CP90 to CP93)

The capture registers are 24-bit registers connected to TM9. These registers can be read in units of 32 bits only.

Figure 7-6. Capture Registers 90 to 93 (CP90 to CP93)

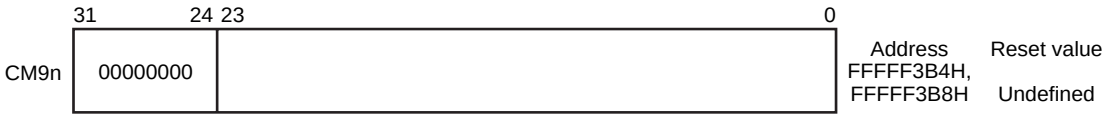


Remark n = 0 to 3

(c) Compare registers 90 and 91 (CM90 and CM91)

The compare registers are 24-bit registers that are connected to TM9. These registers can be read and written in units of 32 bits. The values written to bits 24 to 31 are ignored.

Figure 7-7. Compare Registers 90 and 91 (CM90 and CM91)



Remark n = 0 to 3

7.1.5 Control registers of timers 8 and 9

(1) 24-bit timer mode control register 80 (TMC80)

TMC80 enables and disables TM8 counting and specifies the count clock.

This register can be read and written in units of one or eight bits.

Figure 7-8. 24-Bit Timer Mode Control Register 80 (TMC80)

After reset: 01H R/W Address: FFFFF3A4H

	7	6	5	4	3	2	1	0
TMC80	CE8	OST8	0	0	PRM83	PRM82	PRM81	PRM80

CE8	Specifies Whether to Enable or Disable Timer Counting
0	Disables timer counting (stops with TM8 = 000000H)
1	Enables timer counting
When the ECLR8 bit of the TMC82 register is 1, the timer does not start counting up until TCLR8 is input. When the ECLR8 bit of the TMC82 register is 0, writing 1 to the CE8 bit is used as the counting start trigger. Therefore, the timer does not start even if ECLR8 = 0 after CE8 is set with ECLR8 = 1.	

OST8	Specifies the Operation Performed after Timer Overflows
0	The timer continues counting after it overflows
1	The timer retains 000000H and stops after it overflows
The timer resumes counting up by the following operation: When ECLR8 = 0: Write 1 to the CE8 bit When ECLR8 = 1: Trigger input to the timer clear pin (TCLR8)	

PRM83	PRM82	PRM81	PRM80	Count Clock
0	0	0	1	f_{XX}/2 (default value)
0	0	1	0	f_{XX}/4
0	0	1	1	f_{XX}/8
0	1	0	0	f_{XX}/16
0	1	0	1	f_{XX}/32
0	1	1	0	f_{XX}/64
0	1	1	1	f_{XX}/128
1	0	0	0	f_{XX}/256
1	1	1	1	TI8 input
Other than above				Setting prohibited

Caution Do not change the count clock while the timer is operating.

Remark f_{XX}: System clock

(2) 24-bit timer mode control register 81 (TMC81)

TMC81 selects the function of each capture/compare register and specifies whether to enable or disable the timer clear function.

Even if the contents of TMC81 are rewritten while timer 8 is operating, the contents of each register and the timer count operation are not affected.

This register can be read and written in units of one or eight bits.

Figure 7-9. 24-Bit Timer Mode Control Register 81 (TMC81)

After reset: 00H R/W Address: FFFFF3A6H

	7	6	5	4	3	2	1	0
TMC81	CMS83	CMS82	CMS81	CMS80	IMS83	IMS82	IMS81	IMS80

CMS8n	IMS8n	Selects CC8n Register Operation Mode and Interrupt Source
0	0	Operates as a capture register. Generates an interrupt request at the capture timing
0	1	Setting prohibited
1	0	Operates as a compare register. Generates an interrupt request when the coincidence signal is output from the compare register. Ignores the capture trigger from INTCP8n
1	1	Operates as a compare register. Generates an interrupt request when the INTCP8n signal is input

Remark n = 0 to 3

(3) 24-bit timer mode control register 82 (TMC82)

TMC82 selects the function of each capture/compare register and specifies whether to enable or disable the timer clear function.

This register can be read and written in units of one or eight bits.

Figure 7-10. 24-Bit Timer Mode Control Register 82 (TMC82)

After reset: 00H R/W Address: FFFFF3A8H

	7	6	5	4	3	2	1	0
TMC82	0	0	IMS85	IMS84	0	0	ECLR8	CCLR8

IMS85	IMS84	Selects CC8n Register Operation Mode and Interrupt Source
0	0	Generates an overflow interrupt request by TM8
0	1	Generates an interrupt request by INTTCLR8
1	0	Generates an interrupt request by INTTI8
1	1	Generates an interrupt request by ORing the value of INTTCLR8 and INTTI8

ECLR8	Specifies Whether to Clear and Start TM8 by an External Clear Input (TCLR8)
0	Does not clear TM8
1	Clears TM8 and causes TM8 do start counting up

CCLR8	Specifies Whether to Clear and Start TM8 by CC83 Coincidence
0	Does not clear TM8
1	Clears TM8 and causes TM8 do start counting up

(4) 24-bit timer mode control register 90 (TMC90)

TMC90 enables and disables TM9 counting and specifies the count clock.

This register can be read and written in units of one or eight bits.

Figure 7-11. 24-Bit Timer Mode Control Register 90 (TMC90)

After reset: 01H R/W Address: FFFFF3CCH

	7	6	5	4	3	2	1	0
TMC90	CE9	OST9	CS9	IMS9	PRM93	PRM92	PRM91	PRM90

CE9	Specifies Whether to Enable or Disable Timer Counting
0	Disables timer counting (stops with TM9 = 000000H)
1	Enables timer counting

OST9	Specifies Operation after Timer 9 Overflows
0	Timer 9 continues counting after it overflows
1	Timer 9 retains 000000H and stops after it overflows
The timer resumes counting up by the following operation: Write 1 to the CE9 bit Write 1 to the CS9 bit	

CS9	Controls Clearing and Starting of TM9 by Software
0	TM9 continues counting
1	Clears TM9 and makes it resume counting

Remark CS9 is always 0 when read.

IMS9	Selects Interrupt Source
0	Generates an interrupt request when TM9 overflows
1	Generates an interrupt request by the INTTI9 signal

PRM93	PRM92	PRM91	PRM90	Count Clock
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$ (default value)
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	1	1	1	TI9 input
Other than above				Setting prohibited

Caution Do not change the count clock while the timer is operating.

Remark f_{XX}: System clock

(5) 24-bit timer mode control register 91 (TMC91)

TMC91 selects the capture trigger for the capture register (CP91).

This register can be read and written in units of one or eight bits.

Figure 7-12. 24-Bit Timer Mode Control Register 91 (TMC91)

After reset: 00H R/W Address: FFFFF3CEH

	7	6	5	4	3	2	1	0
TMC91	0	0	0	0	0	0	0	TRSEL

TRSEL	Selects Capture Trigger of CP91
0	External capture signal (INTCP91)
1	Vsync signal

(6) Timer output control register 8 (TOC8)

TOC8 controls the timer output from the TO80 and TO81 pins.

This register can be read and written in units of one or eight bits.

Figure 7-13. Timer Output Control Register 8 (TOC8)

After reset: 00H R/W Address: FFFFF3AAH

	7	6	5	4	3	2	1	0
TOC8	0	0	0	0	ENTO81	ALV81	ENTO80	ALV80

ENTO8n	Enables Each Corresponding Timer Output (TO8n)
0	Disables timer output (The level in the reverse phase of the ALV8n bit (inactive level) is output from the corresponding TO8n pin. The TO8n pin level does not change even if the coincidence signal is generated from the corresponding compare register.)
1	Enables timer output (When the coincidence signal is generated from the corresponding compare register, the timer output changes. The level in the reverse phase of the ALV8n bit (inactive level) is output until the first coincidence signal is generated after timer output is enabled.)

ALV8n	Specifies Active Level of Timer Output (TO8n)
0	Low level
1	High level

Caution The TO80 and TO81 outputs are not changed by an external interrupt signal (INTCP80 to INTCP83). When using the TO80 or TO81 signal, specify the capture/compare registers as compare registers (set all of CMS80 to CMS83 to 1: Refer to 7.1.5 (2) 24-bit timer mode control register 81 (TMC81)).

Remarks 1. Reset of the TO80 and TO81 output F/Fs is given priority.

2. n = 0, 1

(7) Timer overflow status register (TOVS)

The TM8 and TM9 overflow flags are assigned to the TOVS register.

This register can be read and written in units of one or eight bits.

The TOVS register can be tested and reset by software to poll the occurrence of an overflow.

Figure 7-14. Timer Overflow Status Register (TOVS)

After reset: 00H R/W Address: FFFFF3ACH

	7	6	5	4	3	2	1	0
TOVS	0	0	0	0	0	0	OVF9	OVF8

Bit Position	TMn Overflow Flag
0	No overflow occurs
1	An overflow occurs

Interrupt request signal INTOVn, generated by TMn, is sent to the interrupt controller in synchronization with an overflow. Interrupt operation is completely independent of TOVS. The overflow flag (OVFn) received from TMn can be rewritten like other overflow flags. (At this time, the interrupt request flag (OVIFn) corresponding to INTOVn in the interrupt controller is not affected.)
Transfer to the TOVS register is not performed during access from the CPU. Therefore, even if an overflow occurs during read of the TOVS register, the flag values do not change and the occurrence of the overflow is reflected at the next read.

Remark n = 8, 9

(8) Rising edge specification register 2 (EGP2)

The EGP2 register can be read and written in units of one or eight bits.

Figure 7-15. Rising Edge Specification Register 2 (EGP2)

After reset: 00H R/W Address: FFFFF0C8H

	7	6	5	4	3	2	1	0
EGP2	0	0	EGP25	EGP24	EGP23	EGP22	EGP21	EGP20

EGP2n	Specifies Whether to Enable Rising Edge
0	Does not generate the interrupt request signal at the rising edge
1	Generates the interrupt request signal at the rising edge

n = 0: INTCP80
n = 1: INTCP81
n = 2: INTCP82
n = 3: INTCP83
n = 4: TCLR8/INTTCLR8
n = 5: TI8/INTTI8

(9) Falling edge specification register 2 (EGN2)

The EGN2 register can be read and written in units of one or eight bits.

Figure 7-16. Falling Edge Specification Register 2 (EGN2)

After reset: 00H R/W Address: FFFFF0CAH

	7	6	5	4	3	2	1	0
EGN2	0	0	EGN25	EGN24	EGN23	EGN22	EGN21	EGN20

EGN2n	Specifies Whether to Enable Falling Edge
0	Does not generate the interrupt request signal at the falling edge
1	Generates the interrupt request signal at the falling edge

n = 0: INTCP80
n = 1: INTCP81
n = 2: INTCP82
n = 3: INTCP83
n = 4: TCLR8/INTTCLR8
n = 5: TI8/INTTI8

(10) Rising edge specification register 3 (EGP3)

The EGP3 register can be read and written in units of one or eight bits.

Figure 7-17. Rising Edge Specification Register 3 (EGP3)

After reset: 00H R/W Address: FFFFF0CCH

	7	6	5	4	3	2	1	0
EGP3	0	0	0	EGP34	EGP33	EGP32	EGP31	EGP30

EGP3n	Specifies Whether to Enable Rising Edge
0	Does not generate the interrupt request signal at the rising edge
1	Generates the interrupt request signal at the rising edge

n = 0: INTCP90
n = 1: INTCP91
n = 2: INTCP92
n = 3: INTCP93
n = 4: TI9/INTTI9

(11) Falling edge specification register 3 (EGN3)

The EGN3 register can be read and written in units of one or eight bits.

Figure 7-18. Falling Edge Specification Register 3 (EGN3)

After reset: 00H R/W Address: FFFFF0CEH

	7	6	5	4	3	2	1	0
EGN3	0	0	0	EGN34	EGN33	EGN32	EGN31	EGN30

EGN3n	Specifies Whether to Enable Falling Edge
0	Does not generate the interrupt request signal at the falling edge
1	Generates the interrupt request signal at the falling edge

n = 0: INTCP90
n = 1: INTCP91
n = 2: INTCP92
n = 3: INTCP93
n = 4: TI9/INTTI9

7.1.6 Frequency divider

The V850/SV1 can internally divide the signal input to each of pins P141 to P143 (INTCP91 to INTCP93). The result of division is used as an external interrupt request signal or timer capture trigger.

The frequency dividing ratio is set in the event divide control register (EDVC). The frequency of the INTCP signal is divided by using the signal as an internal event signal when the event divide counter (EDV) value is compared with the frequency dividing ratio and a coincidence occurs.

The frequency of the INTCP91 and INTCP92 signals can be divided by 1 to 64. The frequency of the INTCP93 signal can be divided by 1 to 128. The frequency obtained by dividing the frequency of the INTCP92 signal by 1 to 64 can be divided by 1 to 128 again. When this function is used, the INTCP93 signal cannot be used.

Figure 7-19 shows inputs INTCP91 to INTCP93.

Figure 7-19. INTCP9n Inputs (n = 1 to 3)

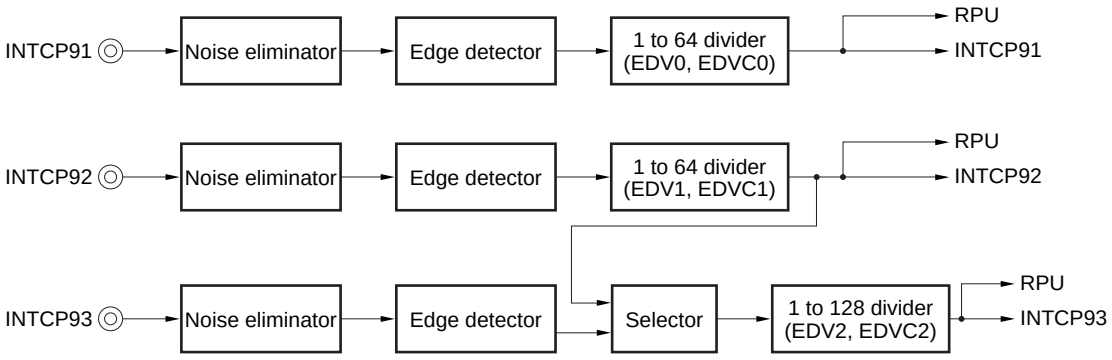
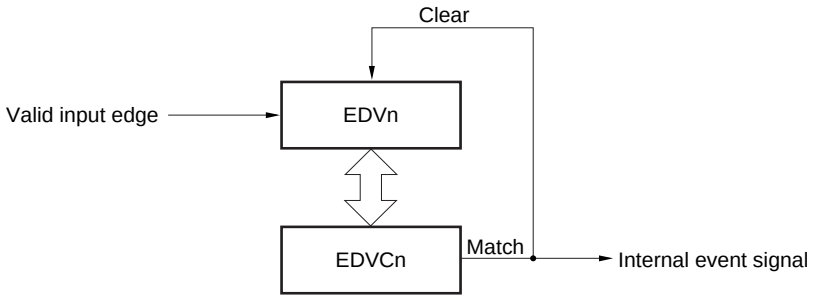


Figure 7-20. Frequency Divider



Remark n = 0 to 2

(1) Event divide counters 0 to 2 (EDV0 to EDV2)

Each of these counters counts the valid edges of the corresponding INTCP9n input signal. The EDV0 and EDV1 registers consist of a six-bit counter, while the EDV2 register consists of a seven-bit counter. (n = 0 to 3)

These registers are cleared in either of the following two cases:

- The value of event divide control register n (EDVCn) coincides with the count value. (n = 0 to 2)
- The EDVCn register is written. (n = 0 to 2)

These registers can only be read in units of one or eight bits. An input edge may not be counted while the valid edge specification in the EGP3/EGN3 register is being changed.

Figure 7-21. Event Divide Counters 0 to 2 (EDV0 to EDV2)

EDV0 After reset: 00H R Address: FFFFF3D2H
EDV1 After reset: 00H R Address: FFFFF3D4H
EDV2 After reset: 00H R Address: FFFFF3D6H

	7	6	5	4	3	2	1	0
EDV0	0	0	EDV05	EDV04	EDV03	EDV02	EDV01	EDV00
EDV1	0	0	EDV15	EDV14	EDV13	EDV12	EDV11	EDV10
EDV2	0	EDV26	EDV25	EDV24	EDV23	EDV22	EDV21	EDV20

(2) Event divide control registers 0 to 2 (EDVC0 to EDVC2)

Each of these registers is used to set the frequency dividing ratio for the INTCP9n (n = 0 to 3) input signal. The setting is used for the frequency dividing ratio as is. Specifying 0 sets the highest frequency dividing ratio. For EDVC0 and EDVC1, the frequency of the INTCP9n input signal is divided by 64. For EDVC2, the frequency of the INTCP9n input signal is divided by 128.

Bits 7 and 6 of EDVC0 and EDVC1 and bit 7 of EDVC2 are fixed to 0. Any writing of 1 is ignored.

These registers can be read and written in units of one or eight bits.

Figure 7-22. Event Divide Control Registers 0 to 2 (EDVC0 to EDVC2)

EDVC0 After reset: 00H R/W Address: FFFFF3D8H
EDVC1 After reset: 00H R/W Address: FFFFF3DAH
EDVC2 After reset: 00H R/W Address: FFFFF3DCH

	7	6	5	4	3	2	1	0
EDVC0	0	0	EDVC05	EDVC04	EDVC03	EDVC02	EDVC01	EDVC00
EDVC1	0	0	EDVC15	EDVC14	EDVC13	EDVC12	EDVC11	EDVC10
EDVC2	0	EDVC26	EDVC25	EDVC24	EDVC23	EDVC22	EDVC21	EDVC20

(3) Event select register (EVS)

The EVS register selects the signal to be input to the EDV2 register. This register can be read and written in units of one or eight bits.

Figure 7-23. Event Select Register (EVS)

After reset: 00H R/W Address: FFFFF3DEH

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

Bit Position	Bit Name	Specifies Whether to Enable Rising Edge
0	ESE	Event select Selects the signal input to the EDV2 register. 0: INTCP93 signal 1: Result of dividing the frequency of the INTCP92 signal by the EDVC1 register

7.2 24-Bit Timer (TM8) Operation

7.2.1 Count operation

Timer 8 functions as a 24-bit interval timer or event counter for the external signals, as specified by 24-bit timer mode control registers 80 to 82 (TMC80 to TMC82).

Timer 8 performs counting up by count clock. Start/stop of counting is controlled by the CE8 bit of timer control register 80 (TMC80).

(1) Start counting

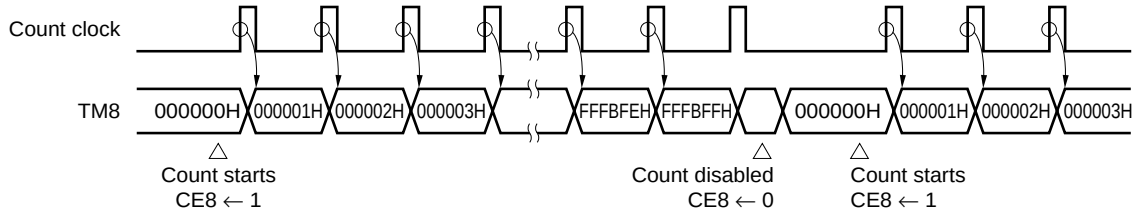
Timer 8 starts counting by setting the CE8 bit to 1 while the ECLR8 bit of the TMC82 register is 0. However, when the ECLR8 bit is 1, timer 8 does not start counting until the TCLR8 signal is input. Therefore, it does not start counting by setting ECLR8 = 0 after setting CE8 = 1 while ECLR8 = 1.

Writing 1 to TM8 during counting operations (CE8 = 1) does not clear the TM8 register, and timer 8 continues counting.

(2) Stop counting

Timer 8 stops counting by setting the CE8 bit to 0. If the OST8 bit of the TMC80 register is set to 1, timer 8 stops operation after occurrence of overflow. However, the value of the timer register can immediately be cleared by setting CE8 = 0.

Figure 7-24. Basic Operation of Timer 8



Remark ECLR8 = 0

7.2.2 Count clock selection

An internal or external count clock can be input to timer 8. The count clock to be used is selected by the PRM80 to PRM83 bits of the TMC80 register.

Caution Do not change the count clock while the timer is operating.

The PRM8n bits of the TMC80 register can be set to select the count clock, as follows:

- An internal count clock is selected from among the following eight clocks: $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, and $f_{XX}/256$.
- The signal input to the TI8 pin is counted (at this time, timer 8 operates as an event counter).

The PRM8n bits can be set as follows:

PRM83	PRM82	PRM81	PRM80	Count Clock
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	1	1	1	TI8 input
Other than above				Setting prohibited

The valid edge of TI8 is specified by EGP2 and EGN2. Refer to **7.1.5 (8) Rising edge specification register 2 (EGP2)** and **7.1.5 (9) Falling edge specification register 2 (EGN2)**.

Remark n = 0 to 3

7.2.3 Overflow

If the TM8 register overflows as a result of counting the count clock frequency to FFFFFFH, a flag is set to the OVF8 flag of the TOVS registers, and an overflow interrupt (INTOV8) is generated. The value of the OVF8 flag is retained until it is changed by user application.

The operation of the TM8 register after occurrence of overflow is determined by the OST8 bit.

(1) Operation after occurrence of overflow when OST8 = 0

The TM8 register continues counting.

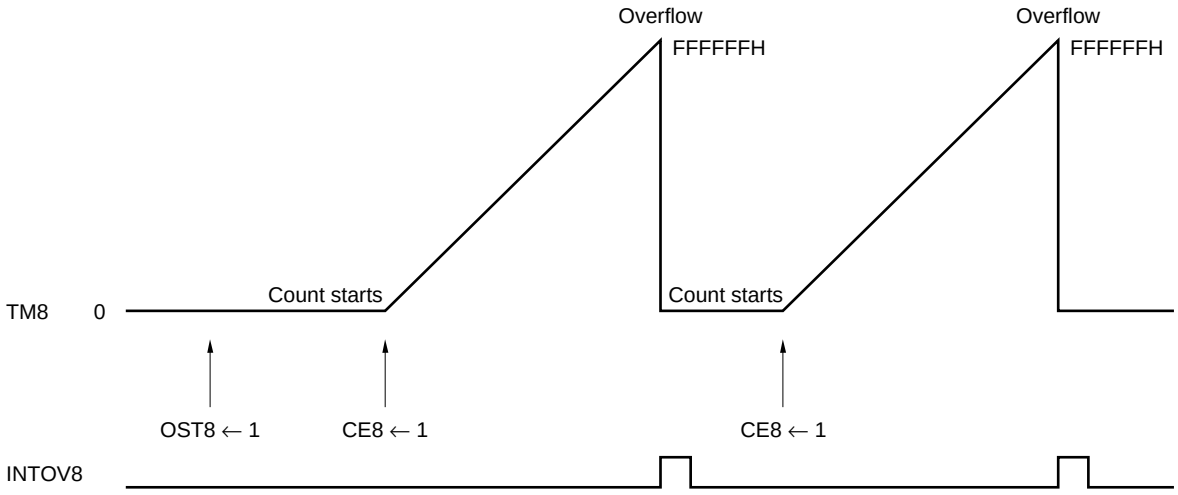
(2) Operation after occurrence of overflow when OST8 = 1

TM8 = 000000H is retained, and the TM8 register stops counting. At this time TM8 stops with CE8 = 1. Perform the following to resume counting.

- When ECLR8 = 0 : Write 1 to CE8 bit
- When ECLR8 = 1 : Trigger input to timer clear pin (TCLR8)

The operation is not affected even if the CE8 bit is set to 1 during count operation.

Figure 7-25. Operation after Occurrence of Overflow (When ECLR8 = 0, OST8 = 1)



Remark ECLR8 = 0

7.2.4 Clearing/starting timer

There are three methods of clearing/starting timer 8: by overflow, by TCLR8 signal input, and by CC83 coincidence.

(1) Clearing/starting by overflow

For the details of the operation, refer to **7.2.3 Overflow**.

(2) Clearing/starting by TCLR8 signal input

Timer 8 usually starts the count operation when the CE8 bit of the TMC80 register is set to 1. It is also possible to clear TM8 and start the count operation by using external input TCLR8.

When the valid edge is input to the TCLR8 signal after ECLR8 = 1, OST8 = 0, and the CE8 bit is set to 1, the count operation is started. If the valid edge is input to TCLR8 during operation, TM8 clears its value and then resume the count operation (refer to **Figure 7-26**).

When the valid edge is input to the TCLR8 signal after ECLR8 = 0, OST8 = 1, and the CE8 bit is set to 1, the count operation is started. When TM8 overflows, the count operation is stopped once and is not resumed until the valid edge is input to TCLR8. If the valid edge of TCLR8 is detected during count operation, TM8 is cleared and continues counting (refer to **Figure 7-27**). Timer 8 does not resume counting even if the CE8 bit is set to 1 after occurrence of overflow. When CE8 = 0, TCLR8 input is invalid.

Figure 7-26. Clearing/Starting Timer by TCLR8 Signal Input (When ECLR8 = 1, CCLR8 = 0, OST8 = 0)

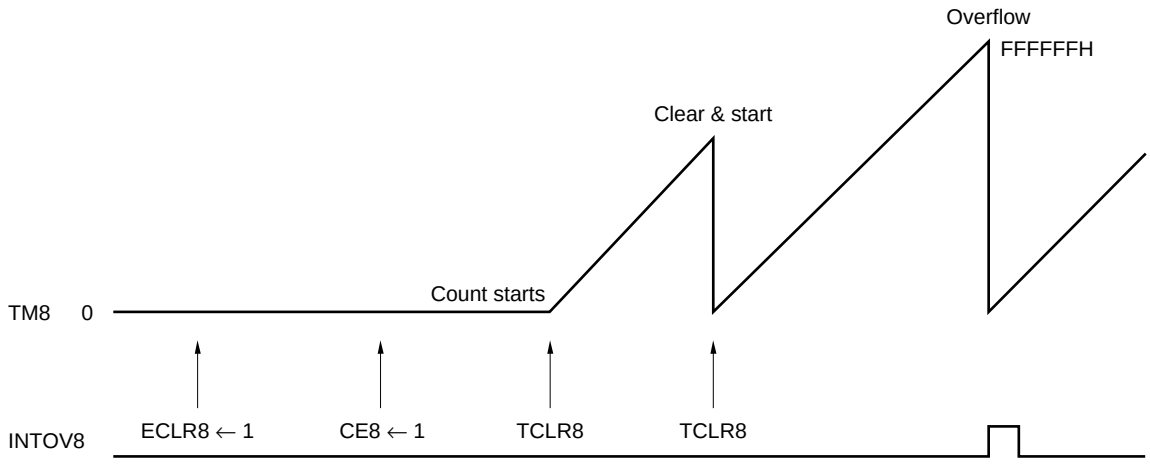
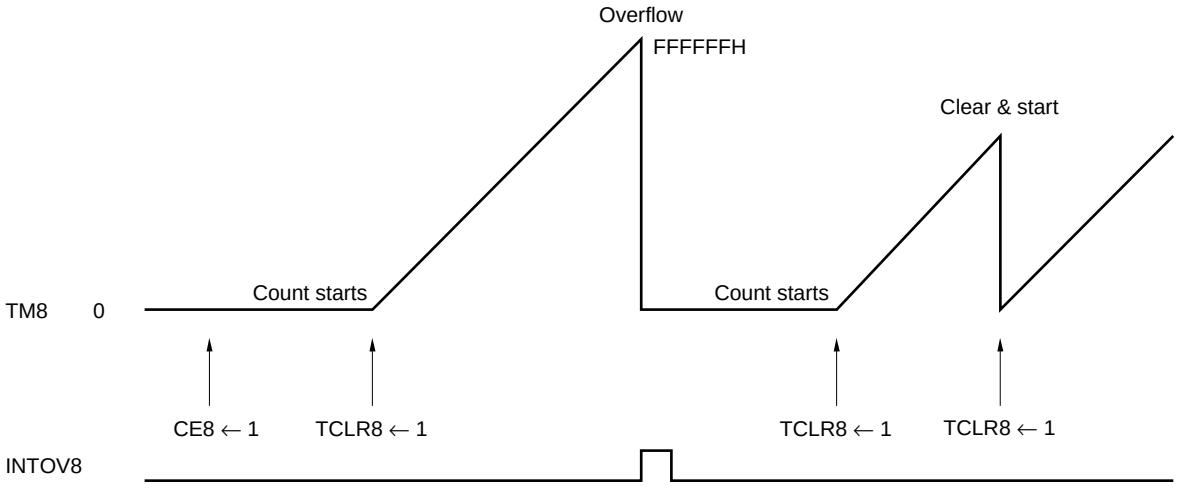


Figure 7-27. Relations between Clear/Start and Overflow of Timer by TCLR8 Signal Input (When ECLR8 = 1, OST8 = 1)



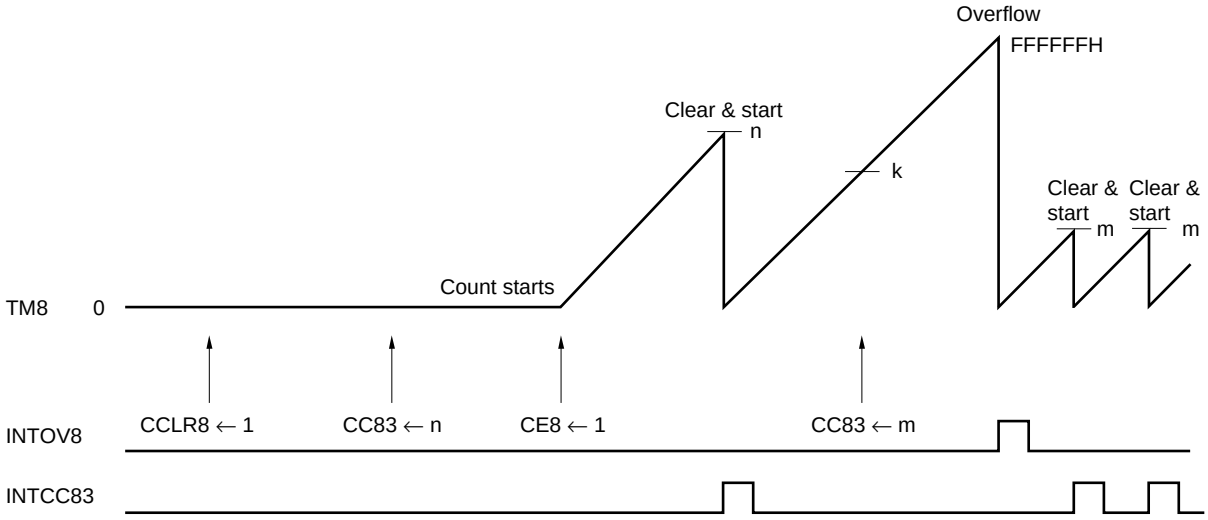
(3) Clearing/starting by CC83 match

Timer 8 usually starts the count operation when CCLR8 =1, CMS83 = 1, and the CE8 bits of the TMC80 registers are set to 1. It is also possible to clear TM8 and start the count operation by generation of CC83 match (INTCC83).

When CCLR8 = 1, CMS83 = 1, and the CE8 bit is set to 1, the count operation is started. If CC83 match is generated during operation, TM8 clears its value and then resumes the count operation (refer to **Figure 7-28**).

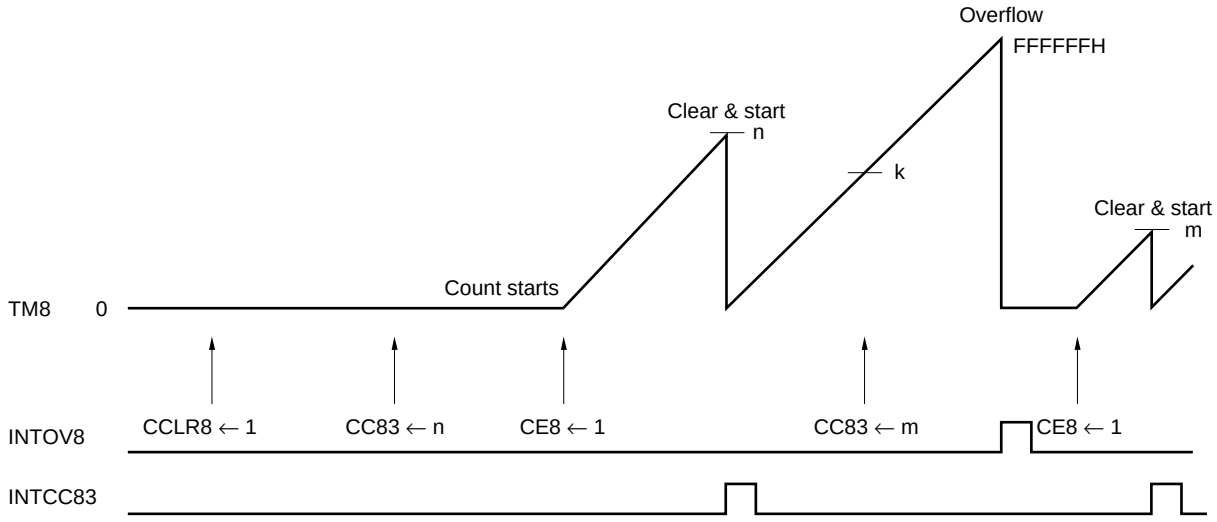
If a value smaller than the current count value of TM8 is set to CC83 during count operation, overflow of TM8 occurs (refer to **Figure 7-29**). For the operation after occurrence of overflow, refer to **7.2.3 Overflow**.

Figure 7-28. Clearing/Starting Timer by CC83 Coincidence (When CCLR8 =1, OST8 = 0)



Remark n > k > m

Figure 7-29. Relations between Clear/Start and Overflow of Timer by CC83 Coincidence (When CCLR8 = 1, OST8 = 1)



Remark n > k > m

7.2.5 Capture operation

When the TMC81 register is set to a capture register, the capture/compare registers (CC80 to CC83) are synchronized with an external trigger to perform capture operations that capture and hold the count values of TM8 to a capture register in asynchronization with a count clock. The valid edge from the external interrupt request input pin INTCP8n is used as the capture trigger. In synchronization with this capture trigger signal, the count values of TM8 during counting are captured and loaded to the capture register and the interrupt request INTCP8n is simultaneously issued. The value of the capture register is retained until the next capture trigger is generated.

When the capture timing to a capture register and write operation to a register by instruction are in contention, the latter is given priority and the capture operation is ignored.

Remark n = 0 to 3

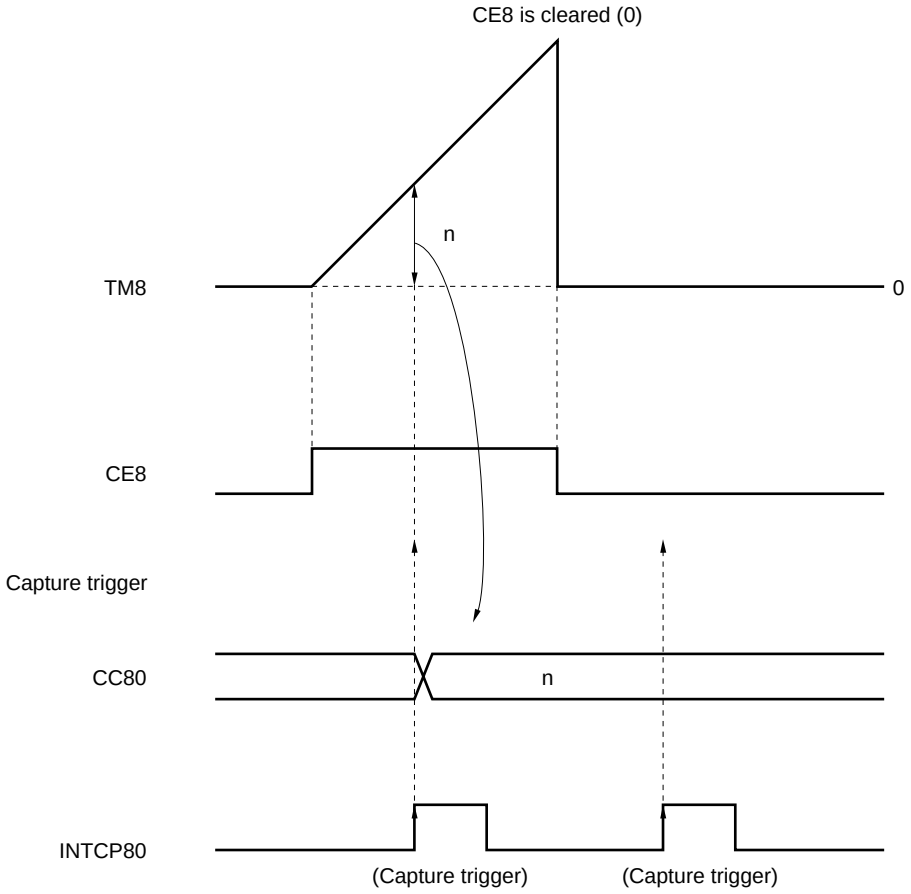
Table 7-4. Capture Trigger Signal to 24-Bit Capture Register (Timer 8)

Capture Trigger Signal	Capture Register	Interrupt Request
INTCP80	CC80	INTCP80
INTCP81	CC81	INTCP81
INTCP82	CC82	INTCP82
INTCP83	CC83	INTCP83

The valid edge of the capture trigger is set by the rising/falling edge specification register 2 (EGP2, EGN2).

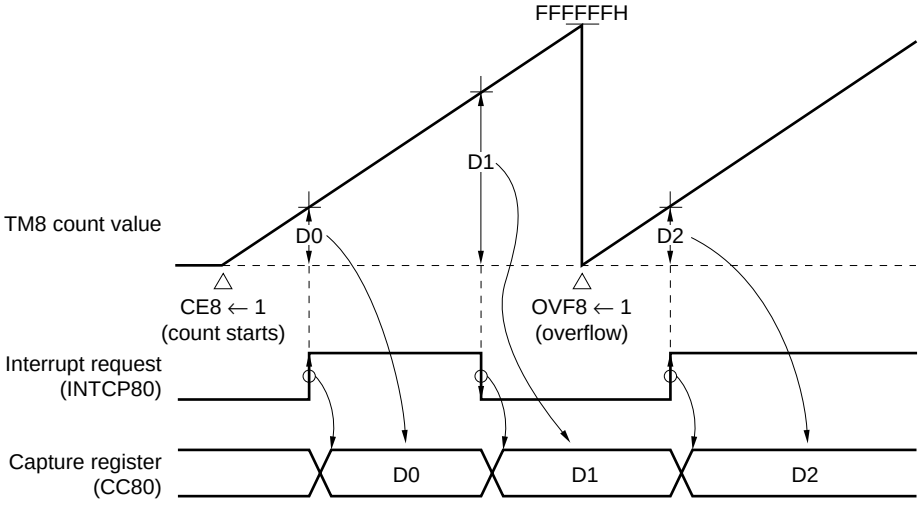
When both the rising and falling edges are specified as the capture trigger, the width of an externally input pulse can be measured. If either the rising or falling edge is specified as the capture trigger, the frequency of the input pulse can be measured.

Figure 7-30. Example of TM8 Capture Operation



Remark The capture operation is not performed even if the interrupt request (INTCP80) is input when CE8 is cleared to 0.

Figure 7-31. Example of TM8 Capture Operation (When Both Edges Are Specified)



Remark D0 to D2: Count value of TM8

7.2.6 Compare operation

When the TMC81 register is set as a compare register, the capture/compare registers (CC80 to CC83) perform a comparison between the value of the compare register with the count values of TM8.

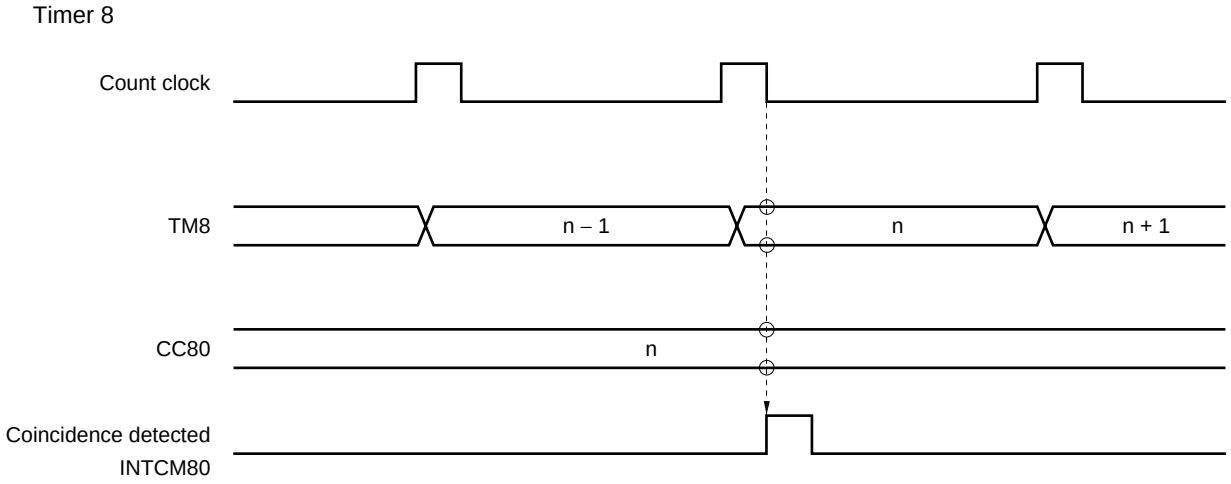
When the count values of TM8 coincide with the value of the compare register programmed in advance, a coincidence signal is sent to the output control circuit (refer to **Figure 7-32**). The levels of the timer output pins (TO80, TO81) can be changed by the coincidence signal, and an interrupt request signal (INTCM80 to INTCM83) can be generated at the same time.

Table 7-5. Interrupt Request Signal from 24-Bit Compare Register (Timer 8)

Compare Register	Interrupt Request	Compare Match Trigger
CC80	INTCM80	TO80 (S)
CC81	INTCM81	TO80 (R)
CC82	INTCM82	TO81 (S)
CC83	INTCM83	TO81 (R), A/D converter

Remark S/R: Set/reset

Figure 7-32. Example of TM8 Compare Operation



Remark Note that the coincidence detected signal INTCM80 is generated immediately after TM8 is incremented as shown above.

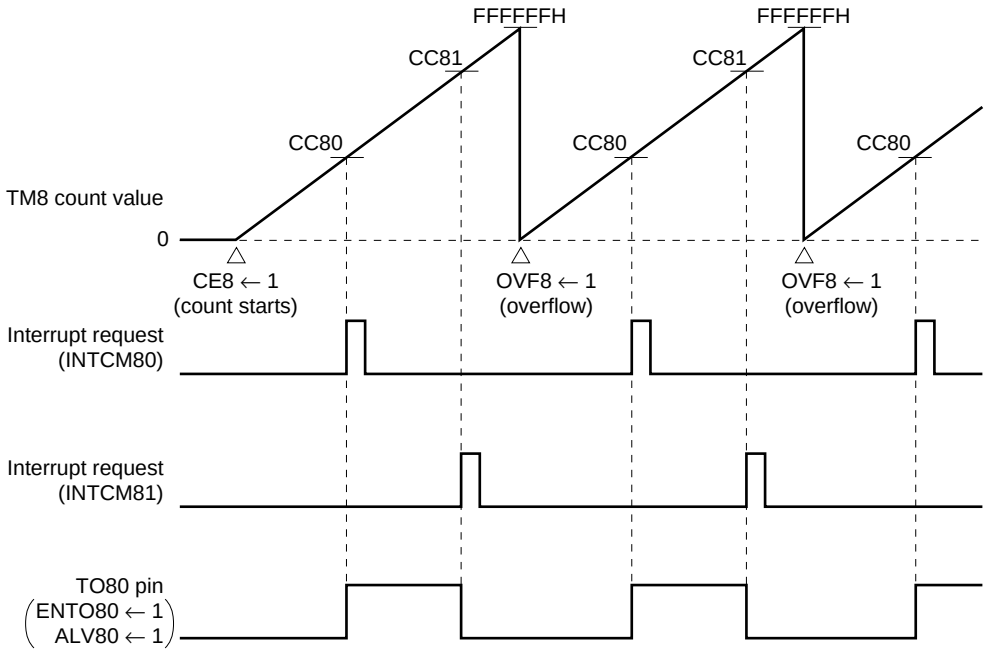
Timer 8 has two timer output pins: TO80 and TO81.

The count values of TM8 are compared with the values of CC82. When the two values coincide, the output level of the TO81 pin is set. The count values of TM8 are also compared with the values of CC83. When the two values coincide, the output levels of the TO81 pin are reset.

Similarly, the count values of TM8 are compared with the values of CC80. When the two values coincide, the output levels of the TO80 pin are set. The count values of TM8 are also compared with the values of CC81. When the two values coincide, the output levels of the TO80 pin are reset.

The output levels of the TO80 and TO81 pins can be specified by the TOC8 register.

Figure 7-33. Example of TM8 Compare Operation (Set/Reset Output Mode)



7.3 24-Bit Timer (TM9) Operation

7.3.1 Count operation

Timer 9 functions as a 24-bit free-running timer or event counter for the external signals, as specified by 24-bit timer mode control registers 90 and 91 (TMC90 and TMC91).

Timer 9 performs counting up by count clock. Start/stop of counting is controlled by the CE9 bit of timer control register 90 (TMC90).

(1) Start counting

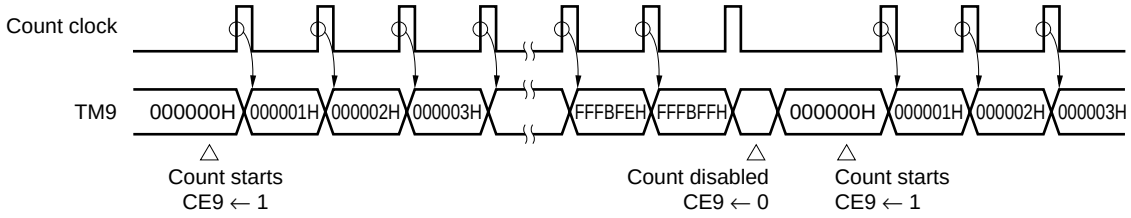
Timer 9 starts counting by setting the CE9 bit to 1.

Writing 1 to TM9 during counting operations (CE9 = 1) does not clear the TM9 register, and timer 9 continues counting.

(2) Stop counting

Timer 9 stops counting by setting the CE9 bit to 0. If the OST9 bit of the TMC90 register is set to 1, timer 9 stops operation after occurrence of overflow. However, the value of the timer register can immediately be cleared by setting CE9 = 0.

Figure 7-34. Basic Operation of Timer 9



7.3.2 Count clock selection

An internal or external count clock can be input to timer 9. The count clock to be used can be selected by the PRM90 to PRM93 bits of the TMC90 register.

Caution Do not change the count clock while the timer is operating.

The PRM9n bits of the TMC90 register can be set to select the count clock as follows:

- An internal count clock is selected from among the following nine clocks: f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, and $f_{XX}/256$.
- The signal input to the TI9 pin is counted (at this time, timer 9 operates as an event counter).

The PRM9n bits can be set as follows:

PRM93	PRM92	PRM91	PRM90	Count Clock
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	1	1	1	TI9 input
Other than above				Setting prohibited

The valid edge of TI9 is specified by EGP3 and EGN3. Refer to **7.1.5 (10) Rising edge specification register 3 (EGP3)** and **7.1.5 (11) Falling edge specification register 3 (EGN3)**.

7.3.3 Overflow

If overflow occurs as a result of counting the TM9 register count clock frequency to FFFFFFH, a flag is set to the OVF9 bits of the TOVS register, and an overflow interrupt (INTOV9) is generated.

The value of the OVF9 flag is retained until it is changed by user application.

The operation of the TM9 register after occurrence of overflow is determined by the OST9 bit.

(1) Operation after occurrence of overflow when OST9 = 0

The TM9 register continues counting.

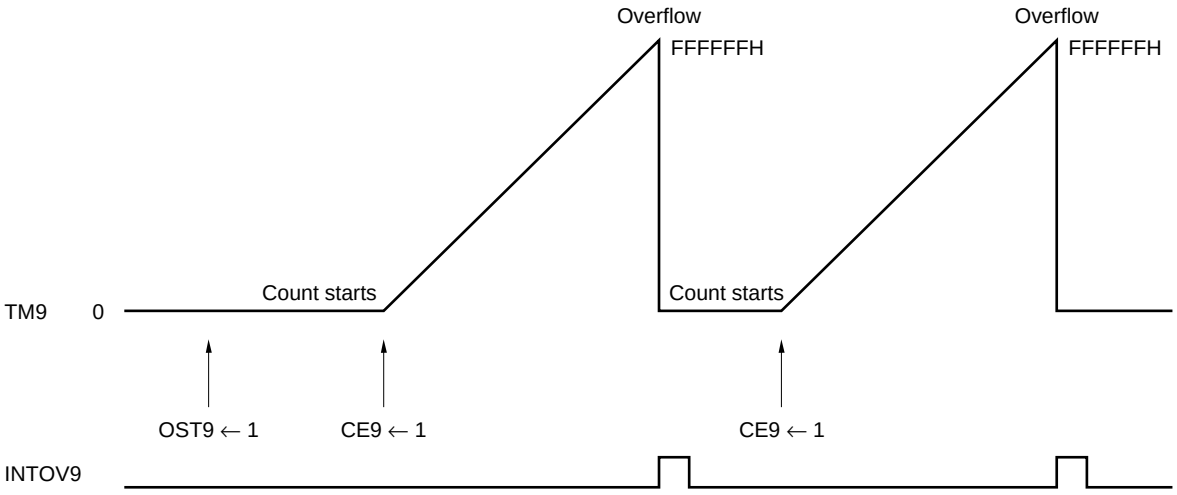
(2) Operation after occurrence of overflow when OST9 = 1

TM9 = 000000H is retained, and the TM9 register stops counting. At this time TM9 stops with CE9 = 1. Perform each of the following to resume counting.

- Write 1 to CE9 bit
- Write 1 to CS9 bit

The operation is not affected even if the CE9 bit is set to 1 during count operation.

Figure 7-35. Operation after Occurrence of Overflow (When OST9 = 1)



7.3.4 Clearing/starting timer

There are two methods of clearing/starting timer 9: by overflow and by software.

(1) Clearing/starting by overflow

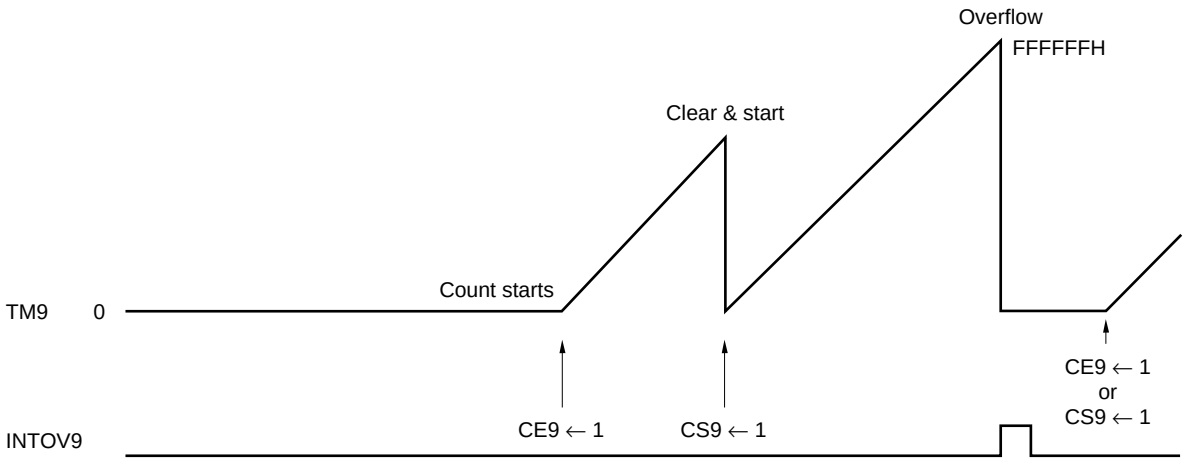
For the details of the operation, refer to **7.3.3 Overflow**.

(2) Clearing/starting by software

When the CS9 bit is set to 1 by software, the TM9 register clears its value and starts counting from 0.

However, this setting of the bit is valid only when the value of the CE9 bit is 1.

Figure 7-36. Clearing/Starting Timer by Software (When OST9 = 1)



7.3.5 Capture operation

A capture operation that is synchronized with an external trigger to capture and hold the count values of TM9 to a capture register in asynchronization with a count clock can be performed. The trigger divided by the valid edge from the external interrupt request input pin INTCP9n is used as the capture trigger. In synchronization with this capture trigger signal, the count values of TM9 during counting, are captured and loaded to the capture register and the interrupt request INTCP9n is simultaneously issued. The value of the capture register is retained until the next capture trigger is generated.

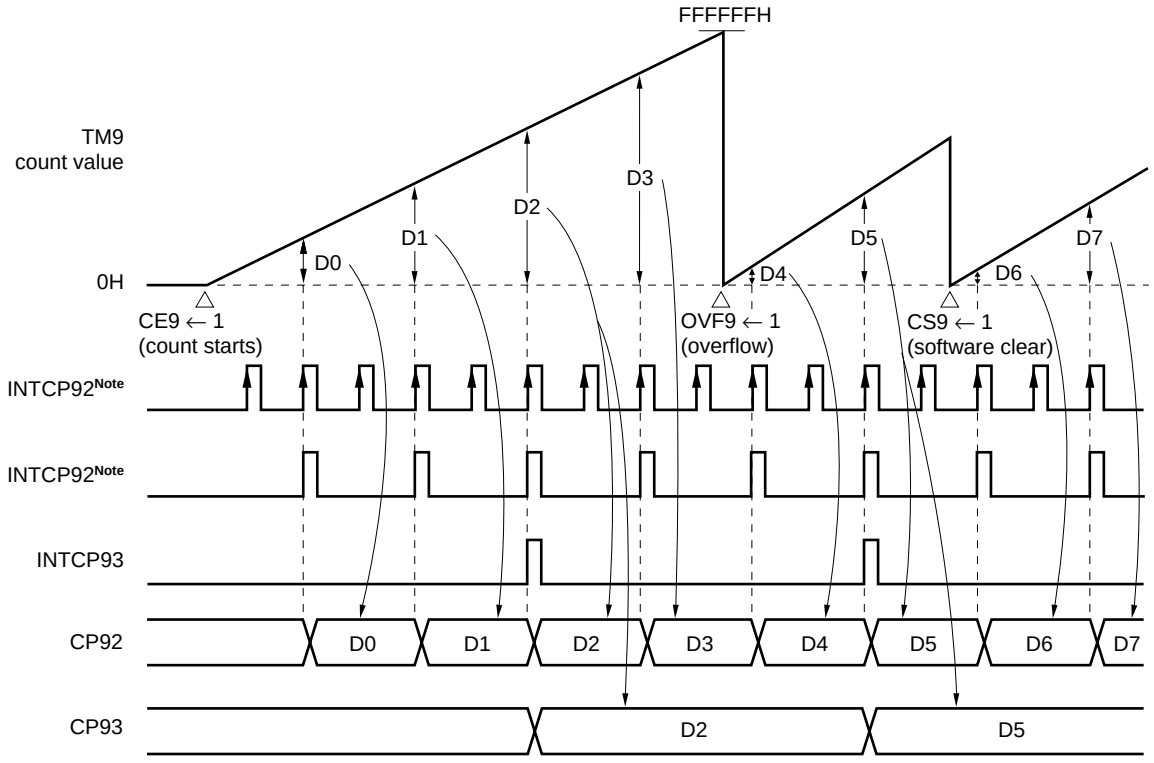
Table 7-6. Capture Trigger Signal to 24-Bit Capture Register (Timer 9)

Capture Trigger Signal	Capture Register	Interrupt Request
INTCP90	CP90	INTCP90
Divide of INTCP91	CP91	INTCP91
Divide of INTCP92	CP92	INTCP92
Divide of INTCP92/INTCP93	CP93	INTCP93

The valid edges of the INTCP9n input are set by EGP3 and EGN3. Refer to **7.1.5 (10) Rising edge specification register 3 (EGP3)** and **7.1.5 (11) Falling edge specification register 3 (EGN3)**. The frequency dividing ratio of the INTCP91 to INTCP93 triggers is set by the EDVC0 to EDVC2 registers and the EVS register. For the details, refer to **7.1.6 Frequency divider**.

Remark n = 0 to 3

Figure 7-37. Example of TM9 Capture Operation



Note The input pin name and interrupt pin name are the same.

7.3.6 Compare operation

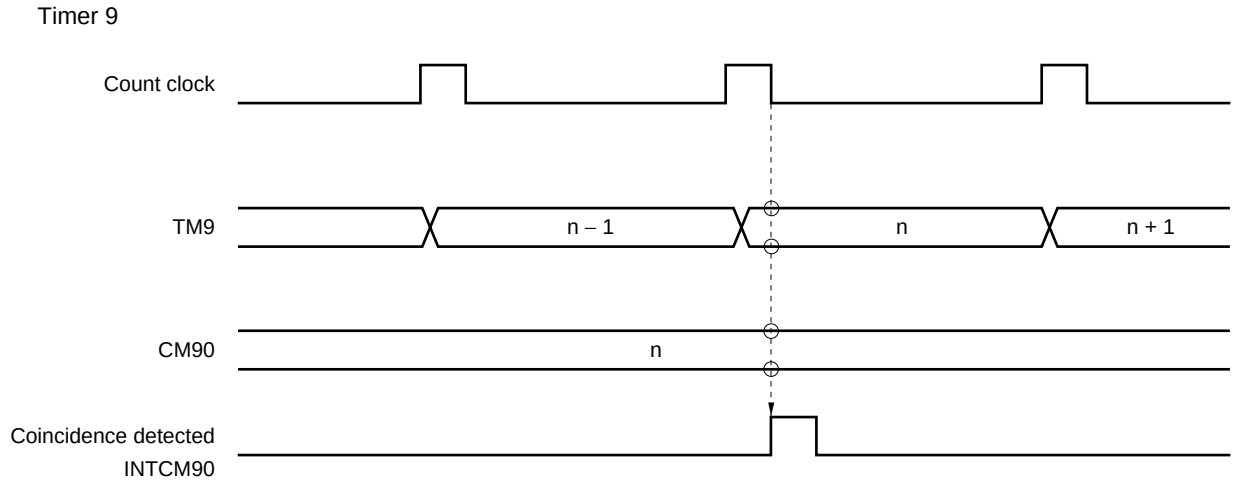
A comparison between the value in a compare register with the count values of TM9 can be performed.

When the count values of TM9 coincide with the value of the compare register programmed in advance, INTCM90 coinciding with CM90 is generated as a trigger of the real-time output port. An interrupt request signal INTCM90 can be generated at the same time.

Table 7-7. Interrupt Request Signal from 24-Bit Compare Register (Timer 9)

Compare Register	Interrupt Request	Compare Match Trigger
CM90	INTCM90	Real-time output port
CM91	INTCM91	–

Figure 7-38. Example of TM9 Compare Operation



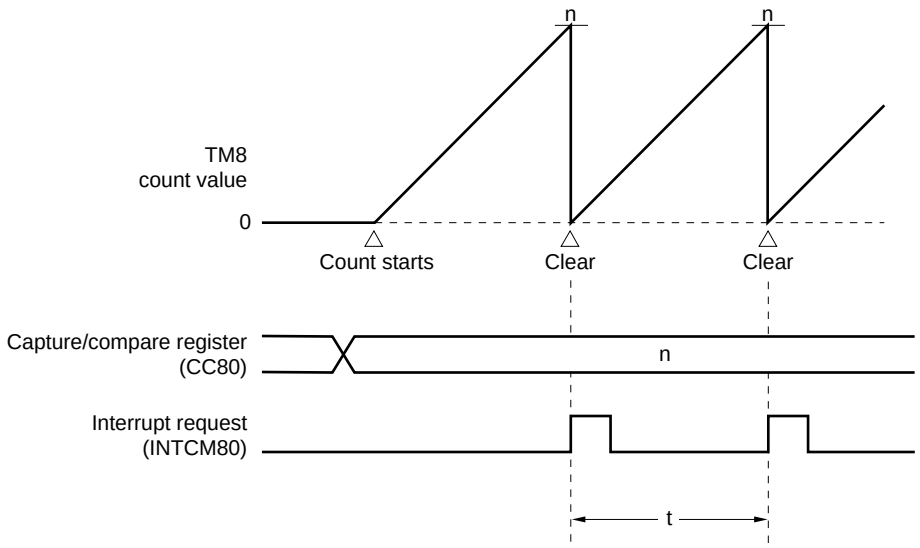
Remark Note that the coincidence detected signal INTCM90 is generated immediately after TM9 is incremented as shown above.

7.4 Application Examples of 24-Bit Timers

(1) Operation as interval timer (timer 8)

The following shows that timer 8 used as an interval timer that repeatedly generates an interrupt request at time intervals specified by the count value set in advance to capture/compare register 80 (CC80). Figure 7-39 shows the timing. Figure 7-40 illustrates the setting procedure.

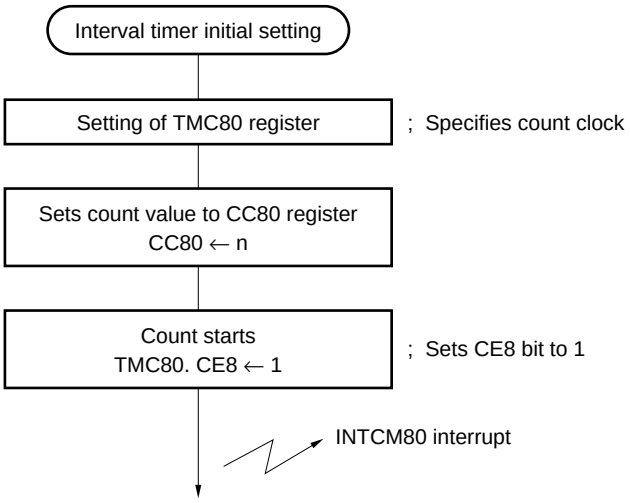
Figure 7-39. Example of Timing of Interval Timer Operation (Timer 8)



Remark n: Value of CC80 register

t: Interval time = (n + 1) × count clock cycle

Figure 7-40. Setting Procedure of Interval Timer Operation (Timer 8)



(2) Pulse width measurement (timer 8 and timer 9)

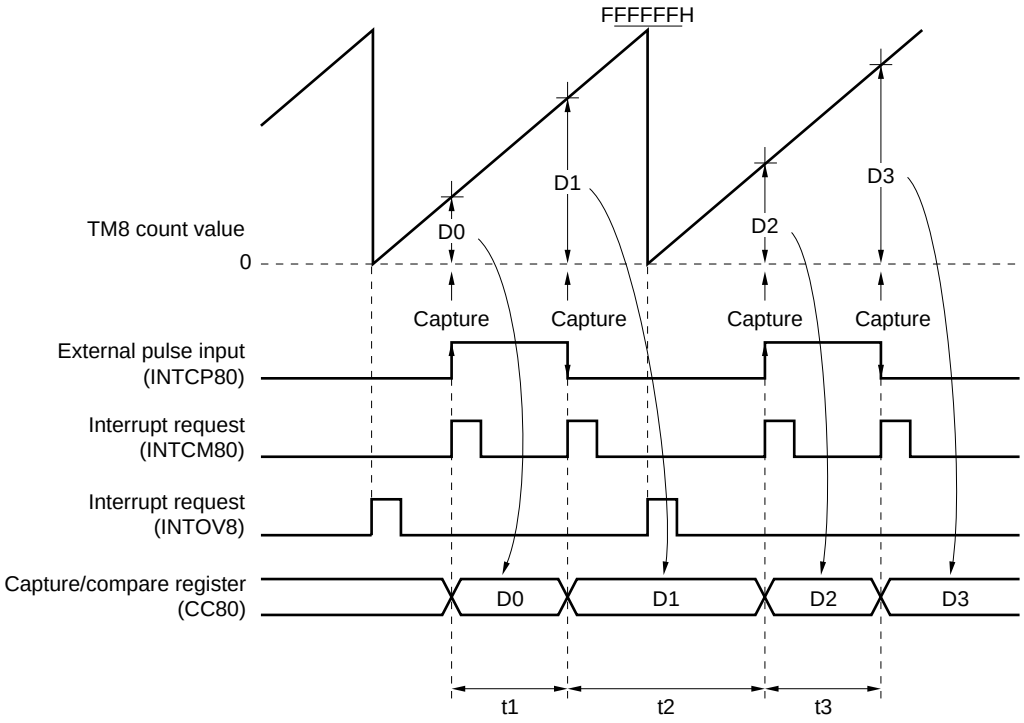
An example of pulse width measurement is shown below.

In this example, the width of the high or low level of an external pulse input to the INTCP80 pin is measured.

The value of timer 8 (TM8) is captured to a capture/compare register (CC80) in synchronization with the valid edge of the INTCP80 pin (both the rising and falling edges) and is pended, as shown in Figure 7-41.

To calculate the pulse width, the difference between the count value of TM8 captured to the CC80 register on detection of the nth valid edge (Dn), and the count value on detection of the (n – 1)th valid edge (Dn – 1) is calculated. This difference is multiplied by the count clock.

Figure 7-41. Pulse Width Measurement Timing (Timer 8)



t1 = (D1 – D0) × count clock cycle

t2 = {(1000000H – D1) + D2} × count clock cycle

t3 = (D3 – D2) × count clock cycle

Remark D0 to D3: Count value of TM8

Figure 7-42. Example of Setting Procedure for Pulse Width Measurement (Timer 8)

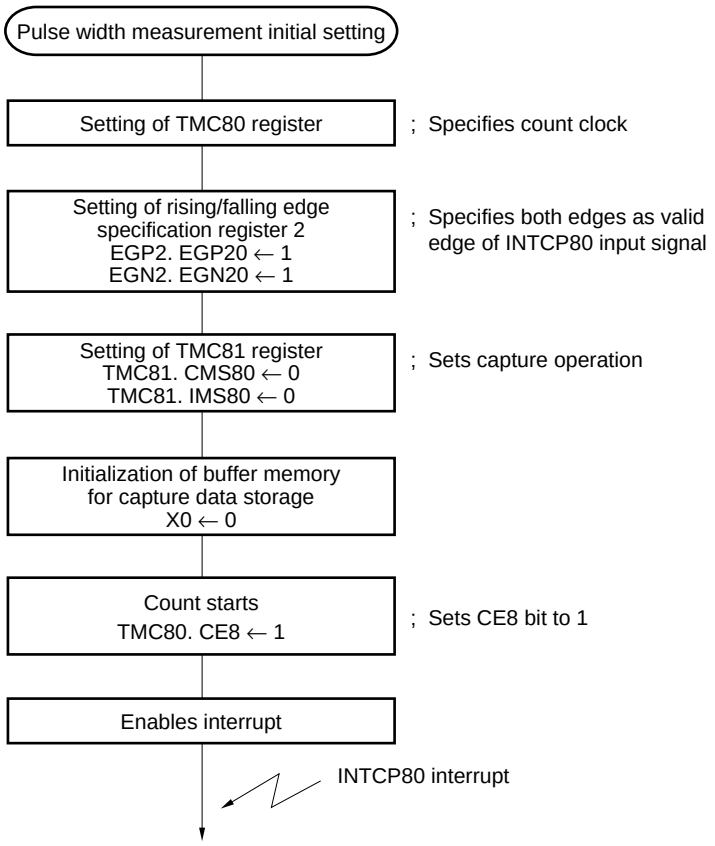
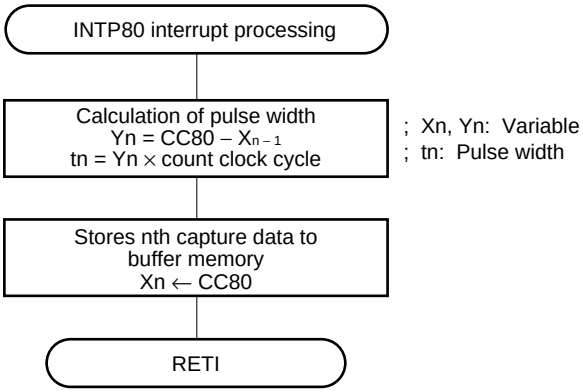


Figure 7-43. Example of Interrupt Request Processing Routine Calculating Pulse Width (Timer 8)



Caution If an overflow occurs two times or more between (n – 1)th capture and nth capture, the pulse width cannot be measured.

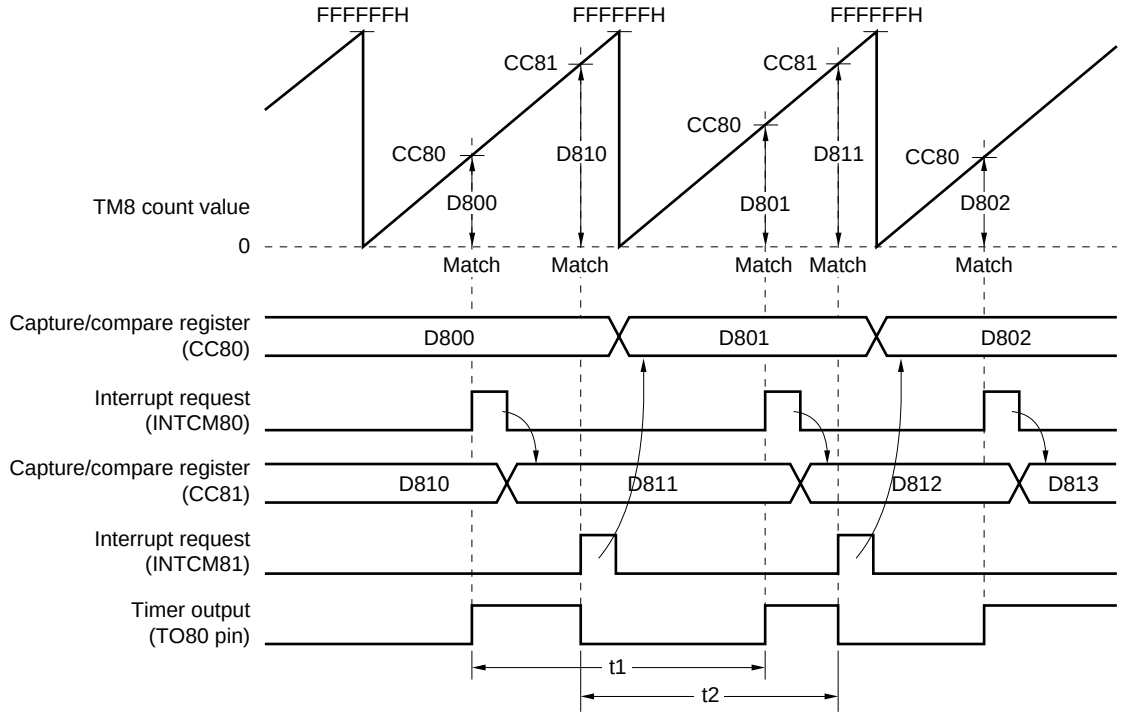
(3) PWM output (timer 8)

Any square wave can be output to timer output pin (TO80, TO81) by combining the use of timer 8 and the timer output function and can be used as a PWM output.

Shown below is an example of PWM output using two capture/compare registers, CC80 and CC81. In this case, a PWM signal with an accuracy of 24 bits can be output from the TO80 pin. Figure 7-44 shows the timing. When timer 8 is used as a 24-bit timer, the rising timing of the PWM output is determined by the value set to capture/compare register CC80, and the falling timing is determined by the value set to capture/compare register CC81.

The interval frequency of timer output can be changed freely by using compare coincidence of CC83 and by clearing and starting TM8.

Figure 7-44. Example of PWM Output Timing (Timer 8)

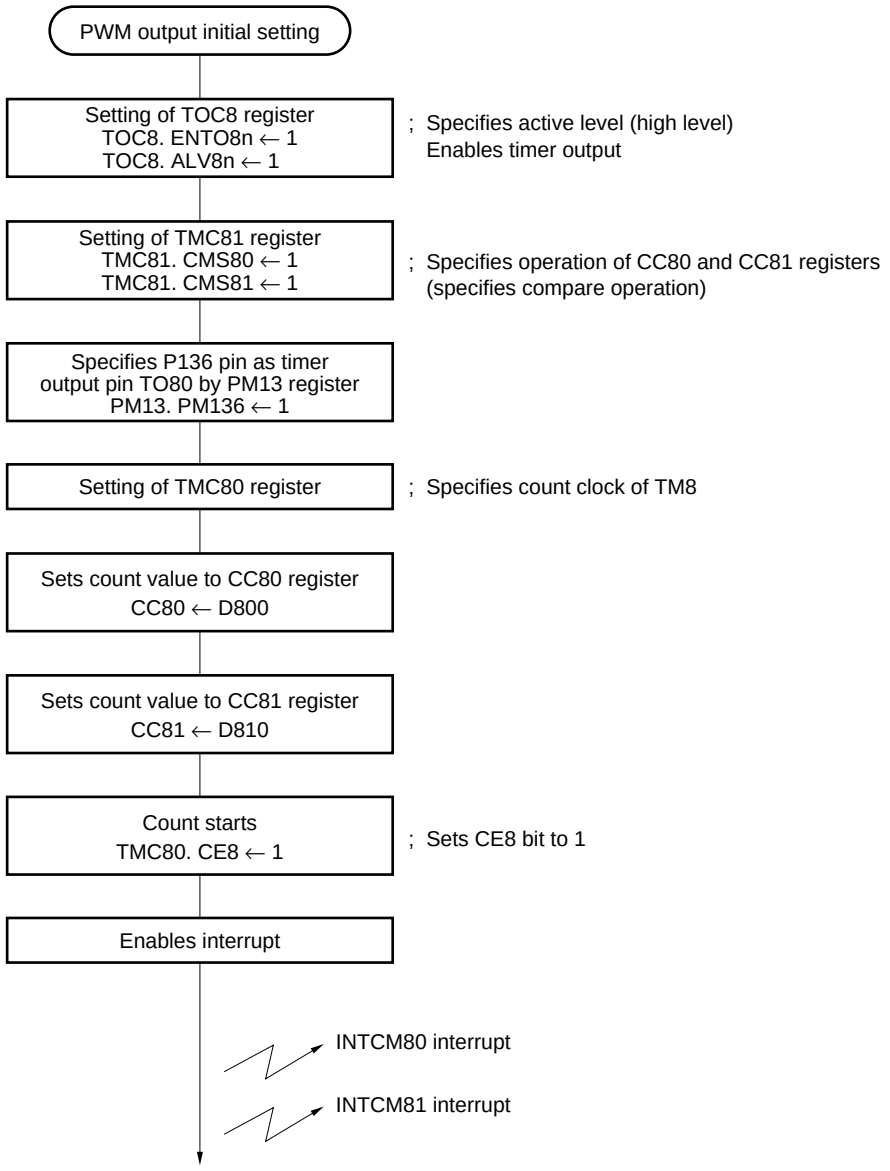


Remark D800 to D802, D810 to D813: Set value of compare register

t1 = {(1000000H – D800) + D801} × count clock cycle

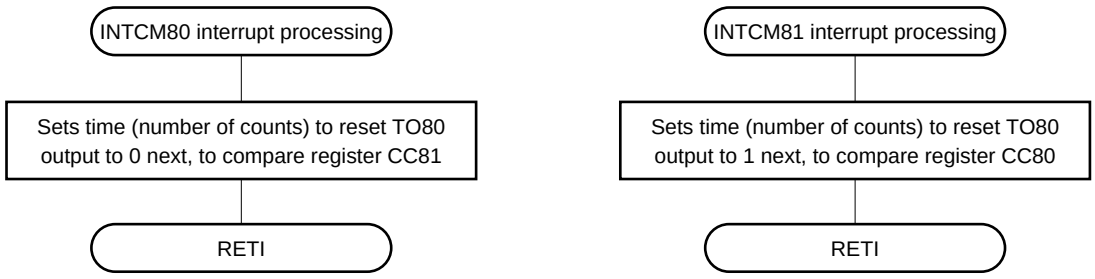
t2 = {(1000000H – D810) + D811} × count clock cycle

Figure 7-45. Example of Setting Procedure for PWM Output (Timer 8)



Remark n = 0, 1

Figure 7-46. Example of Interrupt Request Processing Routine Modifying Compare Value (Timer 8)



(4) Frequency measurement (timer 8 and timer 9)

Timer 8 and timer 9 can be used to measure the cycle or frequency of an external pulse input to the INTCPmn pin (m = 8, 9; n = 0 to 3).

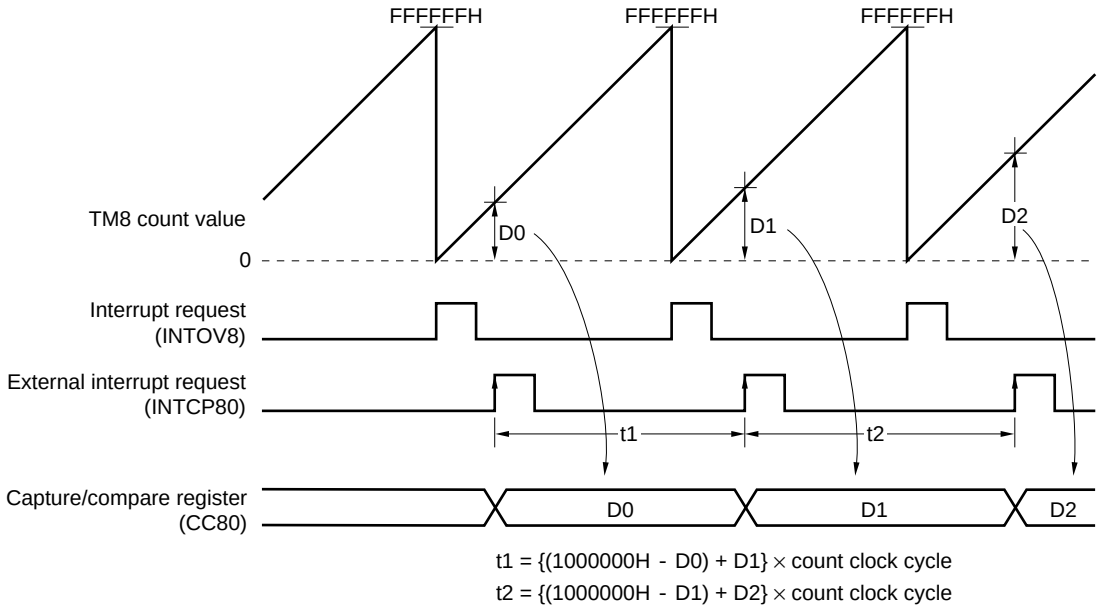
Shown below is an example where the frequency of the external pulse input to the INTCP80 pin is measured with an accuracy of 24 bits, by combining the use of timer 8 and the capture/compare register CC80.

The valid edge of the INTCP80 input signal is specified by the EGP2 register to be the rising edge.

To calculate the frequency, the difference between the count value of TM8 captured to the CC80 register at the nth rising edge (Dn), and the count value captured at the (n – 1)th rising edge (Dn – 1), is calculated, and the value multiplied by the count clock frequency.

The frequency measurement exceeding the maximum count value of TM8 is performed by counting the number of overflow with the INTOV8 overflow interrupt request.

Figure 7-47. Example of Frequency Measurement Timing (Timer 8)



t1 = {(1000000H - D0) + D1} × count clock cycle

t2 = {(1000000H - D1) + D2} × count clock cycle

Remark D0 to D2: Count value of TM8

Figure 7-48. Example of Setting Procedure for Frequency Measurement (Timer 8)

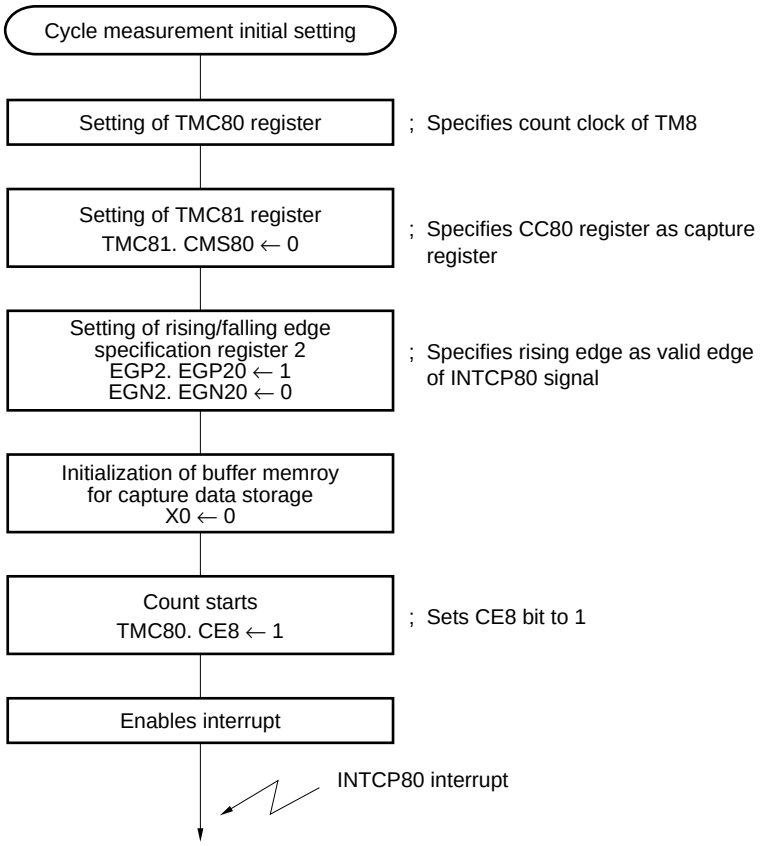
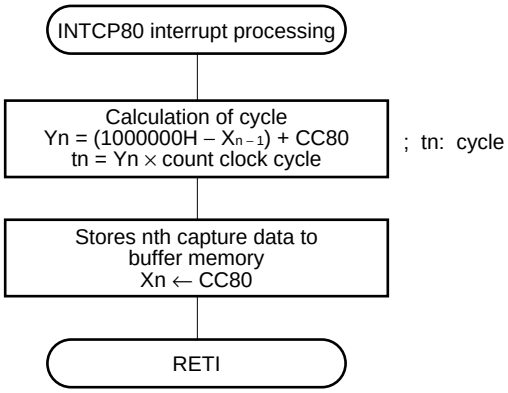


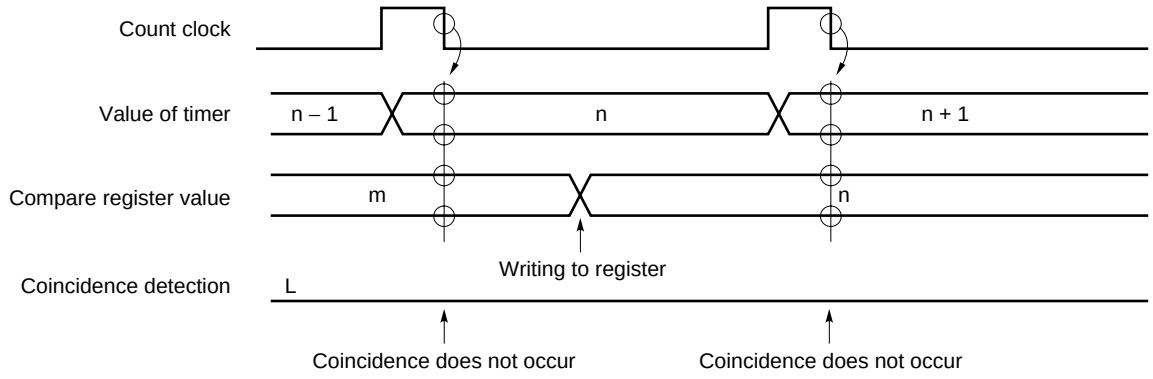
Figure 7-49. Example of Interrupt Request Processing Routine Calculating Cycle (Timer 8)



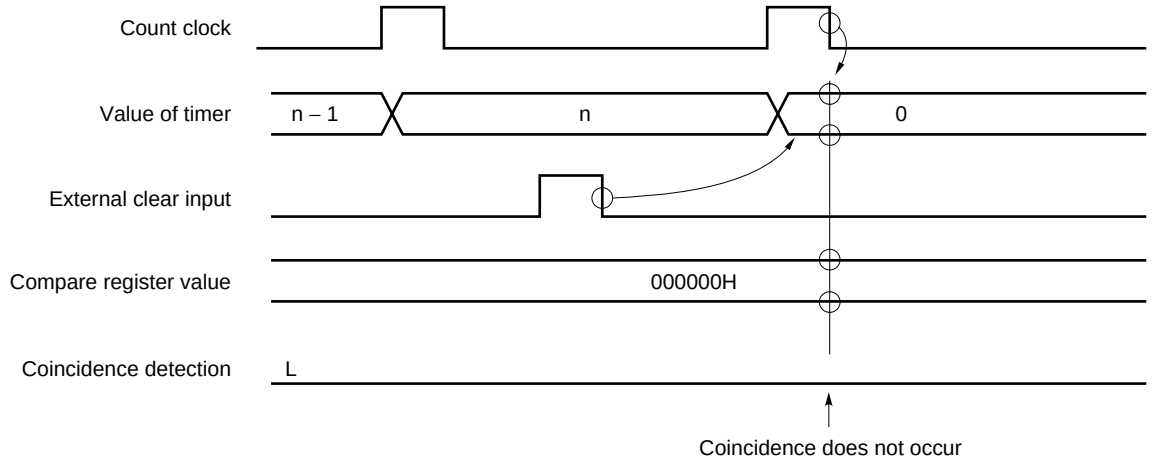
7.5 Cautions for 24-Bit Timers

Coincidence is detected by the compare register immediately after the timer value matches the compare register value, and does not take place in the following cases:

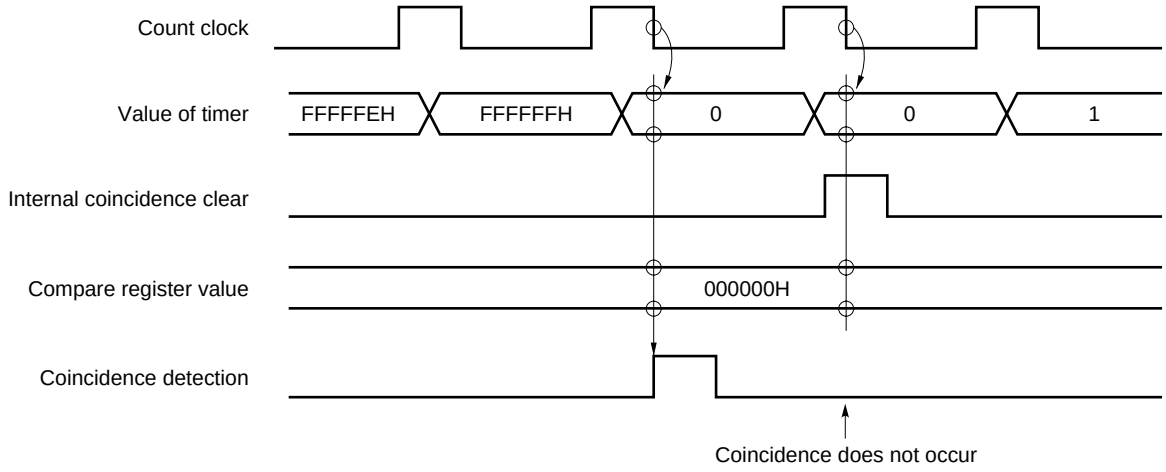
(1) When compare register is rewritten (timer 8 and timer 9)



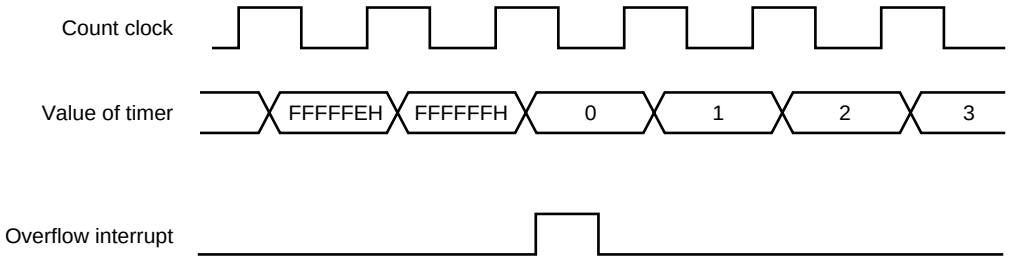
(2) When timer is cleared by external input (timer 8)



(3) When timer is cleared (timer 8)



Remark When timer 8 or timer 9 is operated as a free running timer, the timer value is cleared to 0 when the timer overflows.



7.6 16-Bit Timer (TM0, TM1)

7.6.1 Outline

- 16-bit capture/compare registers: 2 (CRn0, CRn1)
- Independent capture/trigger inputs: 2 (TI0n0, TI0n1)
- Support of output of capture/match interrupt request signals (INTTM0n0, INTTM0n1)
- Event input (shared with TI0n0) via digital noise elimination circuit and support of edge specifications
- Timer output operated by match detection: 1/each (TOn)
  When using P34/TO0 and P35/TO1 pins as TO0 and TO1 pins (timer output), set the value of port 3 (P3) to 0 (port mode output) and port 3 mode register (PM3) to 0. The ORed value of the output of a port and a timer is output.

Remark n = 0, 1

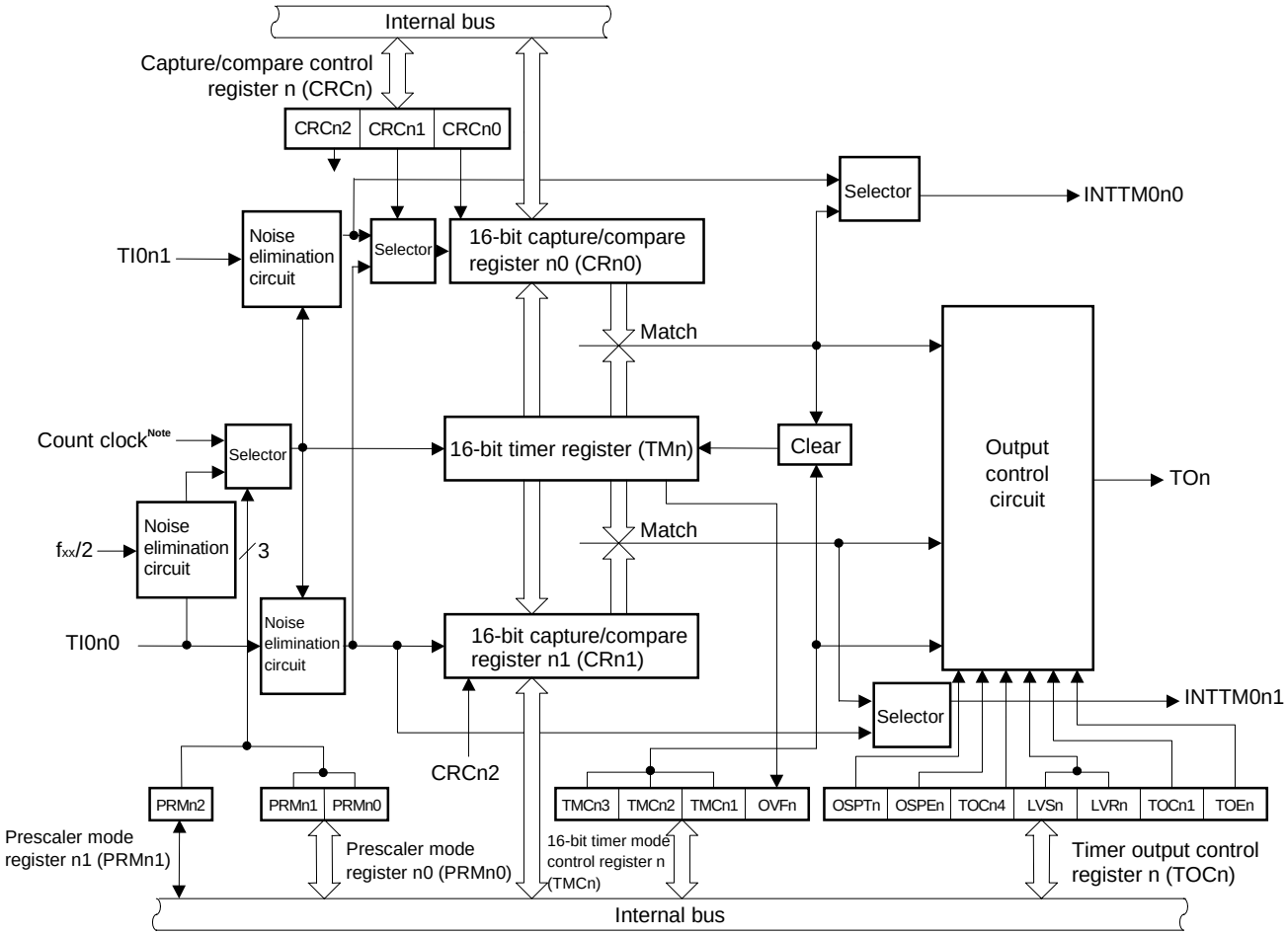
7.6.2 Function

TM0 and TM1 have the following functions:

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

Figure 7-50 shows the block diagram.

Figure 7-50. Block Diagram of TM0 and TM1



Note Count clock is set by the PRMn0 and PRMn1 registers.

Remark n = 0, 1

(1) Interval timer

Generates an interrupt at predetermined time intervals.

(2) PPG output

Can output the square wave, whose frequency and output-pulse width can be changed arbitrarily.

(3) Pulse width measurement

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

(4) External event counter

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

(5) Square wave output

Can output a square wave of any frequency.

(6) One-shot pulse output

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

7.6.3 Configuration

Timers 0 and 1 consist of the following hardware:

Table 7-8. Configuration of Timers 0 and 1

Item	Configuration
Timer registers	16 bits × 2 (TM0, TM1)
Registers	Capture/compare registers: 16 bits × 2 (CRn0, CRn1)
Timer outputs	2 (TO0, TO1)
Control registers	16-bit timer mode control registers 0, 1 (TMC0, TMC1) Capture/compare control registers 0, 1 (CRC0, CRC1) 16-bit timer output control registers 0, 1 (TOC0, TOC1) Prescaler mode registers n0, n1 (PRMn0, PRMn1)

Remark n = 0, 1

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

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

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

<1> At $\overline{\text{RESET}}$ input
<2> If TMCn3 and TMCn2 are cleared
<3> If valid edge of TI0n0 is input in the clear & start mode by inputting valid edge of TI0n0
<4> If TMn and CRn0 coincide with each other in the clear & start mode on coincidence between TMn and CRn0
<5> If OSPTn is set or if the valid edge of TI0n0 is input in the one-shot pulse output mode

(2) Capture/compare registers n0 (CR00, CR10)

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

(a) When using CRn0 as compare register

The value set to CRn0 is always compared with the count value of the TMn register. When the values of the two coincide, an interrupt request (INTTM0n0) is generated. When TMn is used as an interval timer, CRn0 can also be used as a register that holds the interval time.

(b) When using CRn0 as capture register

The valid edge of the TI0n0 or TI0n1 pin can be selected as a capture trigger. The valid edge for TI0n0 or TI0n1 is set by using the PRMn0 register.

When the valid edge for TI0n0 pin is specified as the capture trigger, refer to **Table 7-9.** When the valid edge for TI0n1 pin is specified as the capture trigger, refer to **Table 7-10.**

Table 7-9. Valid Edge of TI0n0 Pin and Capture Trigger of CRn0

ESn01	ESn00	Valid Edge of TI0n0 Pin	CRn0 Capture Trigger
0	0	Falling edge	Rising edge
0	1	Rising edge	Falling edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	No capture operation

Remark n = 0, 1

Table 7-10. Valid Edge of TI0n1 Pin and Capture Trigger of CRn0

ESn11	ESn10	Valid Edge of TI0n1 Pin	CRn0 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

Remark n = 0, 1

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

The value of this register is undefined after the $\overline{\text{RESET}}$ signal is input.

Caution Set a value other than 0000H to CRn0. Consequently, the count operation for one pulse is not enabled when CRn0 is operated as an event counter. In the free running mode or the TI0n0 valid edge clear mode, however, an interrupt request (INTTM0n0) is generated after an overflow (FFFFH) when 0000H is set to CRn0.

(3) Capture/compare register n1 (CRn1)

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

(a) When using CRn1 as compare register

The value set to CRn1 is always compared with the count value of TMn. When the values of the two coincide, an interrupt request (INTTM0n1) is generated.

(b) When using CRn1 as capture register

The valid edge of the TI0n0 pin can be selected as a capture trigger. The valid edge of TI0n0 is specified by using the PRMn0 register.

Table 7-11. Valid Edge of TI0n0 Pin and Capture Trigger of CRn1

ESn01	ESn00	Valid Edge of TI0n0 Pin	CRn1 Capture Trigger
0	0	Falling edge	Falling edge
0	1	Rising edge	Rising edge
1	0	Setting prohibited	Setting prohibited
1	1	Both rising and falling edges	Both rising and falling edges

Remark n = 0, 1

CRn1 is set by using a 16-bit memory manipulation instruction.
The value of this register is undefined after the $\overline{\text{RESET}}$ signal is input.

Caution Set a value other than 0000H to CRn1. Consequently, the count operation for one pulse is not enabled when CRn1 is operated as an event counter. In the free running mode or the TI0n1 valid edge clear mode, however, an interrupt request (INTTM0n1) is generated after an overflow (FFFFH) when 0000H is set to CRn1.

7.6.4 Timer 0, 1 control registers

The following four types of registers control timer 0, 1.

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

Remark n = 1, 0

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

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

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

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

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

Figure 7-51. 16-Bit Timer Mode Control Registers 0, 1 (TMC0, TMC1)

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

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

(n = 0, 1)

TMCn3	TMCn2	TMCn1	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	Coincidence between TMn and CRn0 or coincidence between TMn and CRn1	Generates on coincidence between TMn and CRn0 and coincidence between TMn and CRn1
0	1	1		Coincidence between TMn and CRn0, coincidence between TMn and CRn1, or valid edge of TI0n0	
1	0	0	Clears and starts at valid edge of TI0n0	Coincidence between TMn and CRn0 or coincidence between TMn and CRn1	
1	0	1		Coincidence between TMn and CRn0, coincidence between TMn and CRn1, or valid edge of TI0n0	
1	1	0	Clears and starts on coincidence between TMn and CRn0	Coincidence between TMn and CRn0 or coincidence between TMn and CRn1	
1	1	1		Coincidence between TMn and CRn0, coincidence between TMn and CRn1, or valid edge of TI0n0	

OVFn	Detection of Overflow of 16-Bit Timer Register n
0	Does not overflow
1	Overflows

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

2. The valid edge of the TI0n0 pin is set by using the prescaler mode register n0 (PRMn0).

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

Remark TOn: Output pin of timer n
TI0n0: Input pin of timer n
TMn: 16-bit timer register n
CRn0: Compare register n0
CRn1: Compare register n1

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

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

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

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

Figure 7-52. Capture/Compare Control Registers 0, 1 (CRC0, CRC1)

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

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

(n = 0, 1)

CRCn2	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 TI0n1
1	Captured in reverse phase of valid edge of TI0n0

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 coincidence between TMn and CRn0 is selected by the 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 TI0n0 valid edge, the capture operation does not work.

4. The capture trigger requires a pulse longer than two count clocks, selected by prescaler mode register n0 or n1 (PRMn0 or PRMn1), to ensure capture operation for the signal from TI0n1 or TI0n0.

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

TOCn controls the operation of the timer n output control circuit 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 a 1-bit or 8-bit memory manipulation instruction.

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

Figure 7-53. 16-Bit Timer Output Control Registers 0, 1 (TOC0, TOC1)

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

	7	6	5	4	3	2	1	0
TOCn	0	OSPTn	OSPEn	TOCn4	LVSn	LVRn	TOCn1	TOEn

(n = 0, 1)

OSPTn	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 Coincidence 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 0
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 Coincidence between CRn0 and TMn
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 in free-running mode or in a mode in which the timer is cleared and started at the valid edge of TI0n0.

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

2. LVSn and LVRn are 0 when read after data have 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 for other than one-shot pulse output.

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

PRM0n selects a count clock of the 16-bit timer (TM0) and the valid edge of TI00n input. PRM00 and PRM01 are set by an 8-bit memory manipulation instruction.

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

Remark n = 0, 1

Figure 7-54. Prescaler Mode Register 00 (PRM00)

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 TI001
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 TI000
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM02[Note 1]	PRM01	PRM00	Count Clock Selection			
			Count Clock	f_{XX}		
				20 MHz	10 MHz	2 MHz
0	0	0	$f_{XX}/2$	100 ns	200 ns	1 μs
0	0	1	$f_{XX}/16$	800 ns	1.6 μs	8 μs
0	1	0	INTWTN	–	–	–
0	1	1	TI000 valid edge[Note 2]	–	–	–
1	0	0	$f_{XX}/4$	200 ns	400 ns	2 μs
1	0	1	$f_{XX}/64$	3.2 μs	6.4 μs	32 μs
1	1	0	$f_{XX}/256$	12.8 μs	25.6 μs	128 μs
1	1	1	Setting prohibited	–	–	–

Notes 1. Bit 0 of the PRM01 register

2. The external clock requires a pulse longer than two internal clocks ($f_{XX}/2$).

Cautions 1. When selecting the valid edge of TI00n as the count clock, do not specify the valid edge of TI00n to clear and start the timer and as a capture trigger.

2. Before setting data to PRM0n, always stop the timer operation.

3. If the 16-bit timer (TM0) operation is enabled by specifying the rising edge or both edges for the valid edge of the TI00n pin while the TI00n 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 TI00n pin. However, the rising edge is not detected when operation is enabled after it has been stopped.

Figure 7-55. Prescaler Mode Register 01 (PRM01)

After reset: 00H R/W Address: FFFFF20EH

	7	6	5	4	3	2	1	0
PRM01	0	0	0	0	0	0	0	PRM02[Note]

Note Set together with bits 0 and 1 of the PRM00 register. (Refer to **Figure 7-54**)

(5) Prescaler mode registers 10, 11 (PRM10, PRM11)

PRM1n selects a count clock of the 16-bit timer (TM1) and the valid edge of TI01n input. PRM10 and PRM11 are set by an 8-bit memory manipulation instruction.

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

Remark n = 0, 1

Figure 7-56. Prescaler Mode Register 10 (PRM10)

After reset: 00H R/W Address: FFFFF216H

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

ES111	ES110	Selects Valid Edge of TI011
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES101	ES100	Selects Valid Edge of TI010
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM12[Note 1]	PRM11	PRM10	Count Clock Selection			
			Count Clock	f_{XX}		
				20 MHz	10 MHz	2 MHz
0	0	0	$f_{XX}/2$	100 ns	200 ns	1 μs
0	0	1	$f_{XX}/4$	200 ns	400 ns	2 μs
0	1	0	$f_{XX}/16$	800 ns	1.6 μs	8 μs
0	1	1	TI010 valid edge[Note 2]	–	–	–
1	0	0	$f_{XX}/32$	1.6 μs	3.2 μs	16 μs
1	0	1	$f_{XX}/128$	6.4 μs	12.8 μs	64 μs
1	1	0	$f_{XX}/256$	12.8 μs	25.6 μs	128 μs
1	1	1	Setting prohibited	–	–	–

Notes 1. Bit 0 of the PRM11 register

2. The external clock requires a pulse longer than two internal clocks ($f_{XX}/2$).

Cautions 1. When selecting the valid edge of TI01n as the count clock, do not specify the valid edge of TI01n to clear and start the timer and as a capture trigger.

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

3. If the 16-bit timer (TM1) operation is enabled by specifying the rising edge or both edges for the valid edge of the TI01n pin while the TI01n 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 TI01n pin. However, the rising edge is not detected when operation is enabled after it has been stopped.

Figure 7-57. Prescaler Mode Register 11 (PRM11)

After reset: 00H R/W Address: FFFFF21EH

	7	6	5	4	3	2	1	0
PRM11	0	0	0	0	0	0	0	PRM12[Note]

Note Set together with bits 0 and 1 of the PRM10 register. (Refer to **Figure 7-56**)

7.7 16-Bit Timer Operation

7.7.1 Operation as interval timer (16 bits)

The 16-bit timer register n (TMn) operates as an interval timer when the 16-bit timer mode control register n (TMCn) and capture/compare control register n (CRCn) are set as shown in Figure 7-58 (n = 0, 1).

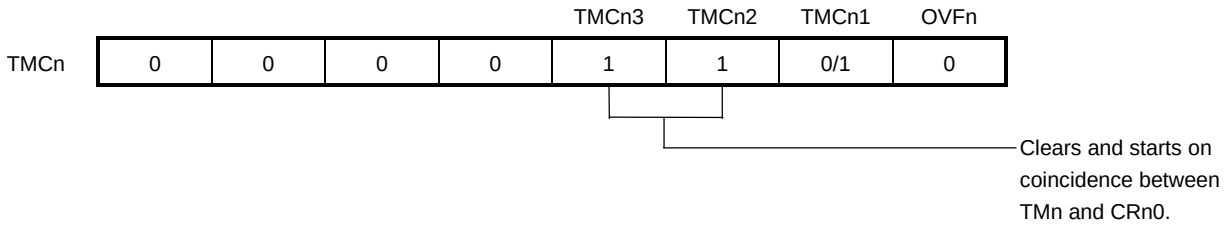
In this case, TMn repeatedly generates an interrupt at the time interval specified by the count value set in advance to the 16-bit capture/compare register n (CRn0, CRn1).

When the count value of TMn coincides with the set value of CRn0, the value of TMn is cleared to 0, and the timer continues counting. At the same time, an interrupt request signal (INTTM0n0) is generated.

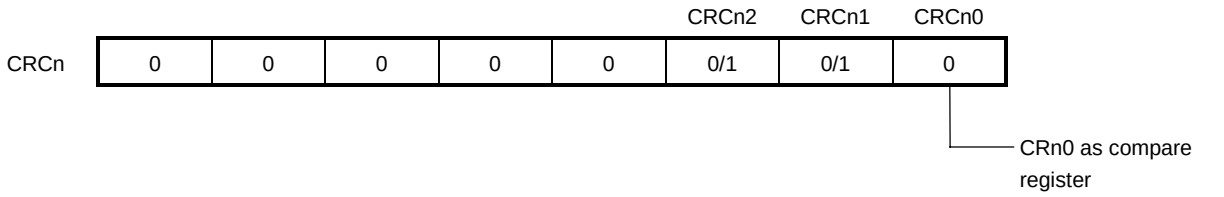
The count clock of the 16-bit timer/event counter can be selected by bits 0 and 1 (PRMn0 and PRMn1) of the prescaler mode register n0 (PRMn0) and by bits 0 (PRMn2) of prescaler mode register n1 (PRMn1).

Figure 7-58. Control Register Settings When TMn Operates as Interval Timer

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

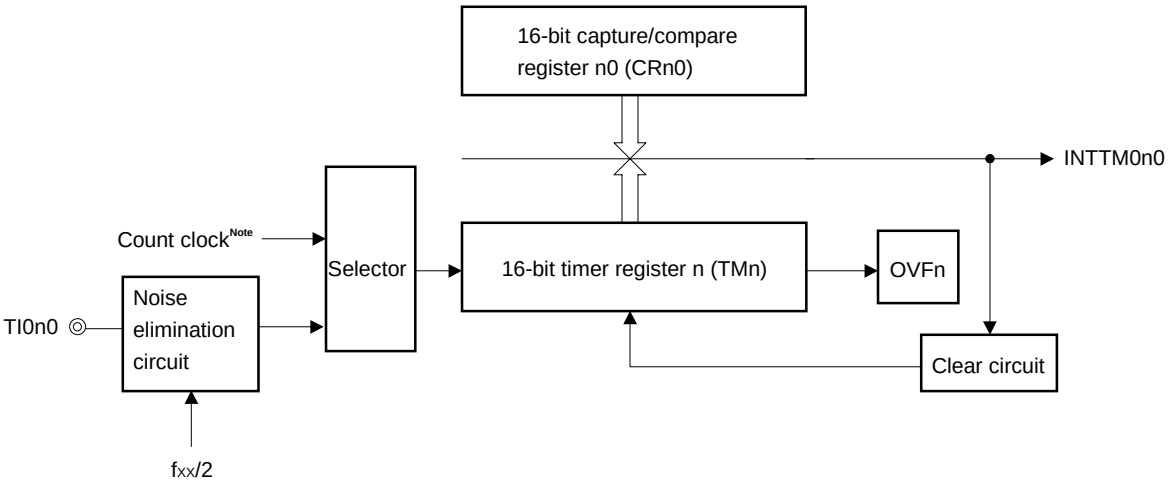


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the interval timer function. For details, refer to **Figures 7-51** and **7-52**.

Figure 7-59. Configuration of Interval Timer

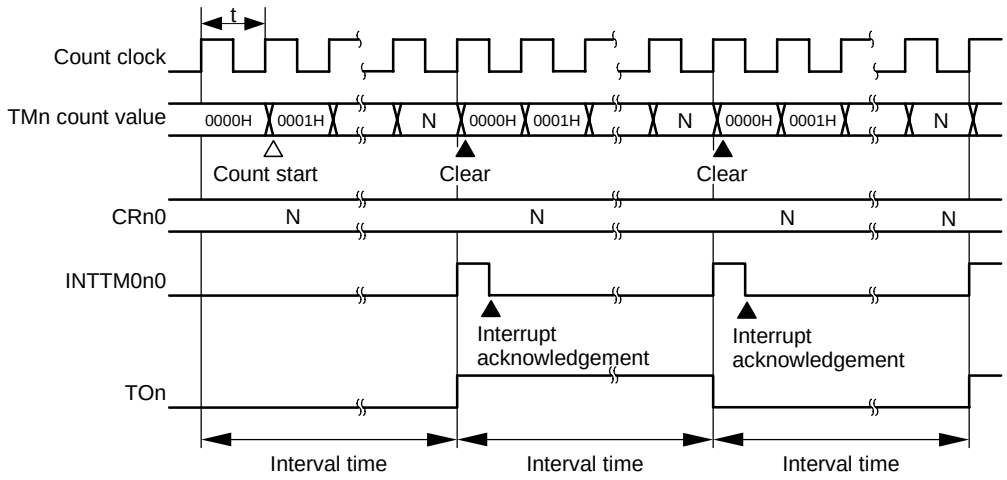


Note Count clock is set by the PRMn0, PRMn1 registers.

Remarks 1. " ◎—— " indicates the signal that can be directly connected ports.

2. n = 0, 1

Figure 7-60. Timing of Interval Timer Operation



Remarks 1. Interval time = (N + 1) × t: N = 0001H to FFFFH

2. n = 0, 1

7.7.2 PPG output operation

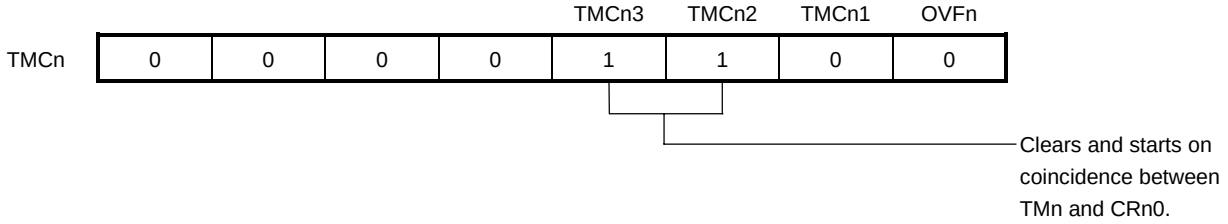
TMn can be used for PPG (Programmable Pulse Generator) output by setting the 16-bit timer mode control register n (TMCn) and capture/compare control register n (CRCn) as shown in Figure 7-61.

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

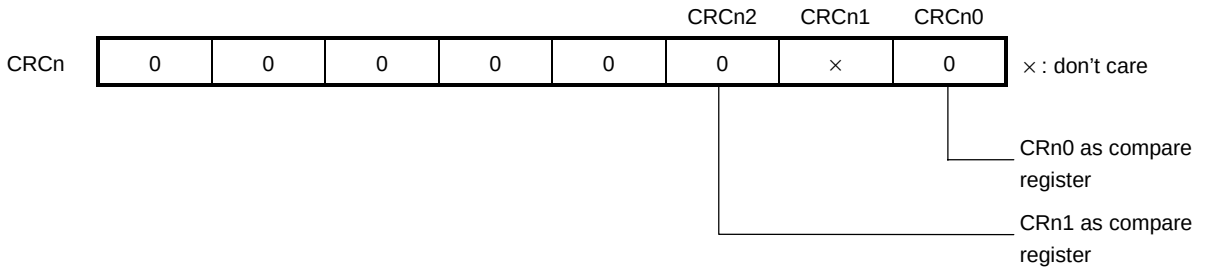
Remark n = 0, 1

Figure 7-61. Control Register Settings in PPG Output Operation

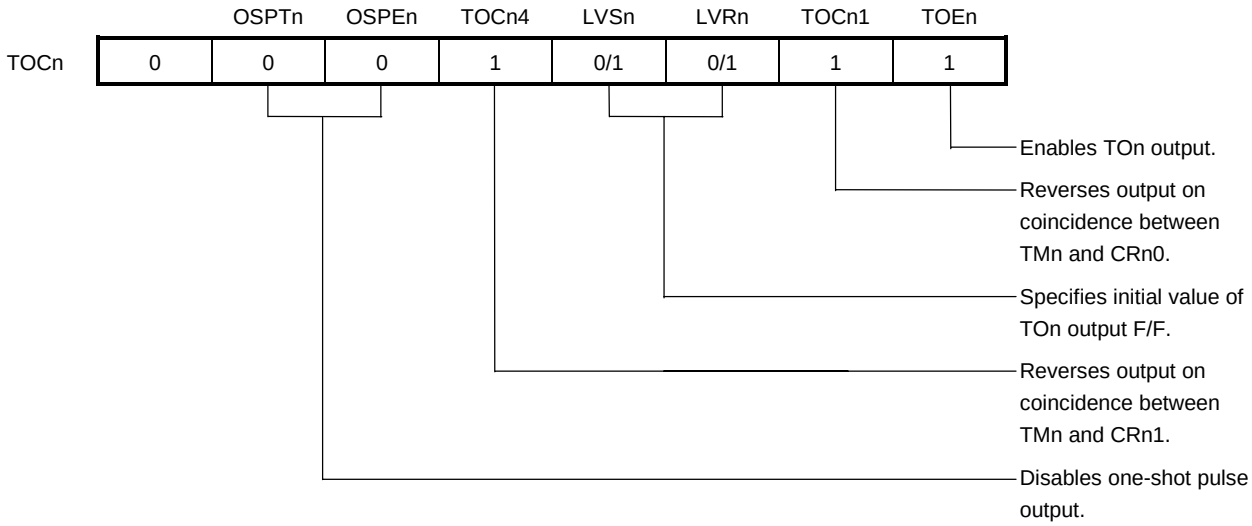
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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



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



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

2. PPG output set the pulse cycle to (CRn0 setup value + 1).
Duty factor is (CRn1 setup value + 1)/(CRn0 setup value + 1).

7.7.3 Pulse width measurement

The 16-bit timer register n (TMn) can be used to measure the pulse widths of the signals input to the TI0n0 and TI0n1 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 TI0n0 pin.

(1) Pulse width measurement with free running counter and one capture register

If the edge specified by the prescaler mode register n0 (PRMn0) is input to the TI0n0 pin when the 16-bit timer register n (TMn) is used as a free running counter (refer to **Figure 7-62**), the value of TMn is loaded to the 16-bit capture/compare register n1 (CRn1), and an external interrupt request signal (INTTM0n1) is set.

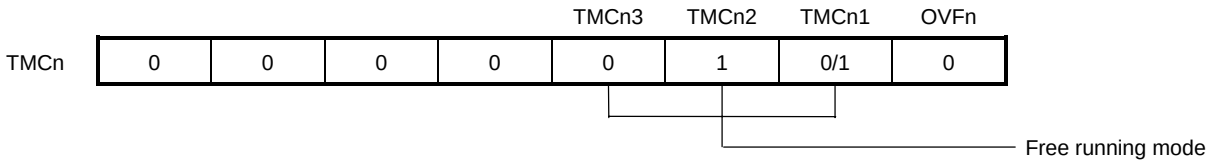
The edge is specified by using bits 6 and 7 (ESn10 and ESn11) of the prescaler mode register n0 (PRMn0). The rising edge, falling edge, or both the rising and falling edges can be selected.

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

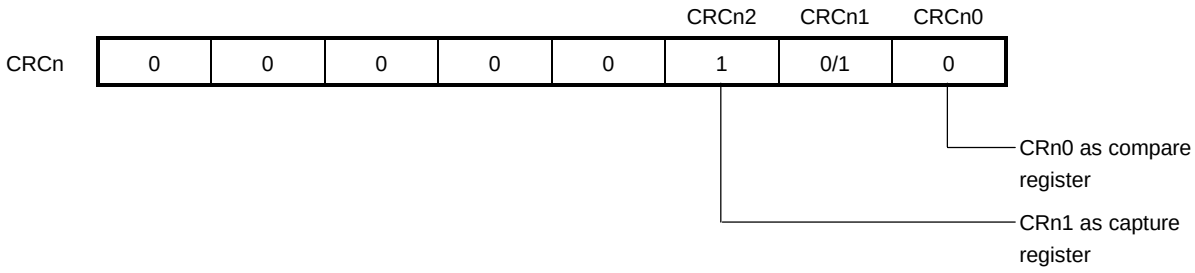
Remark n = 0, 1

Figure 7-62. Control Register Settings for Pulse Width Measurement with Free Running Counter and One Capture Register

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

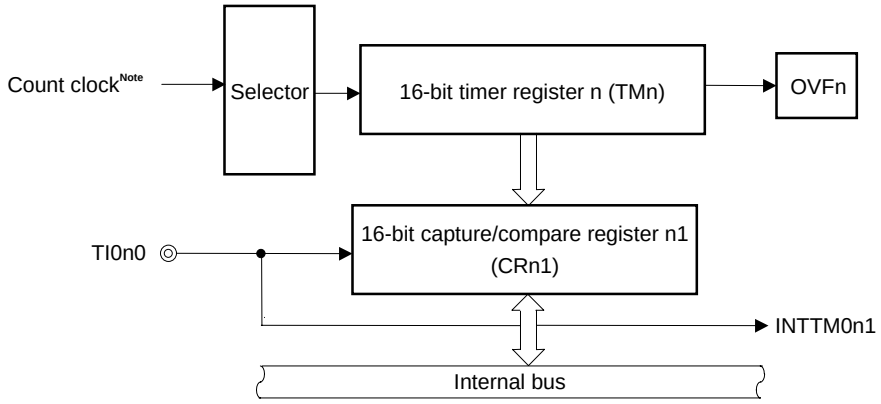


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, refer to **Figures 7-51** and **7-52**.

Figure 7-63. Configuration for Pulse Width Measurement with Free Running Counter

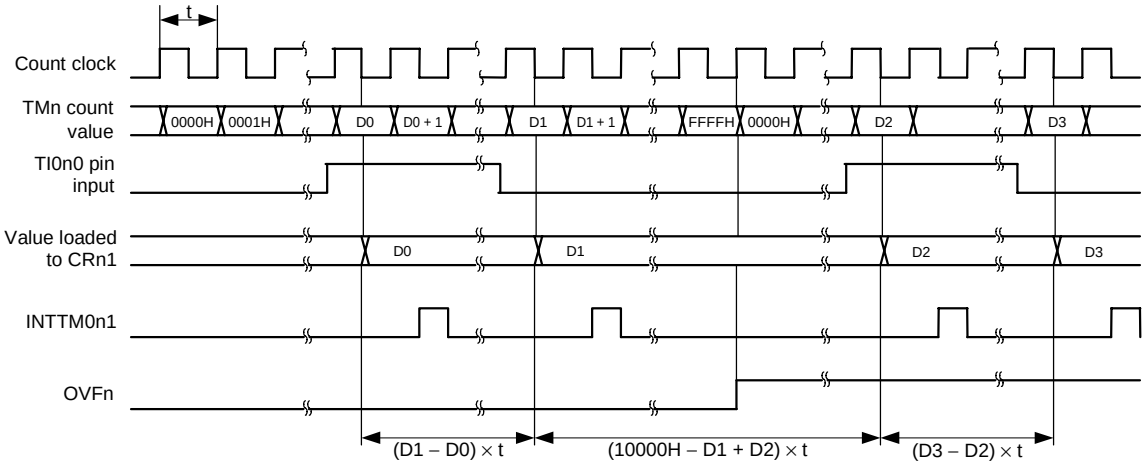


Note Count clock is set by the PRMn0 and PRMn1 registers.

Remarks 1. " ◎—— " indicates the signal that can be directly connected ports.

2. n = 0, 1

Figure 7-64. Timing of Pulse Width Measurement with Free Running Counter and One Capture Register (with Both Edges Specified)



Remark n = 0, 1

(2) Measurement of two pulse widths with free running counter

The pulse widths of the two signals respectively input to the TI0n0 and TI0n1 pins can be measured when the 16-bit timer register n (TMn) is used as a free running counter (refer to **Figure 7-65**).

When the edge specified by bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0) is input to the TI0n0 pin, the value of the TMn is loaded to the 16-bit capture/compare register n1 (CRn1) and an external interrupt request signal (INTTM0n1) is set.

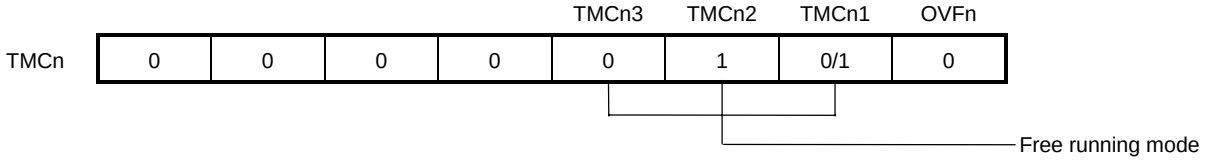
When the edge specified by bits 6 and 7 (ESn10 and ESn11) in PRMn0 is input to the TI0n1 pin, the value of TMn is loaded to the 16-bit capture/compare register n0 (CRn0), and an external interrupt request signal (INTTM0n0) is set.

The edges of the TI0n0 and TI0n1 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.

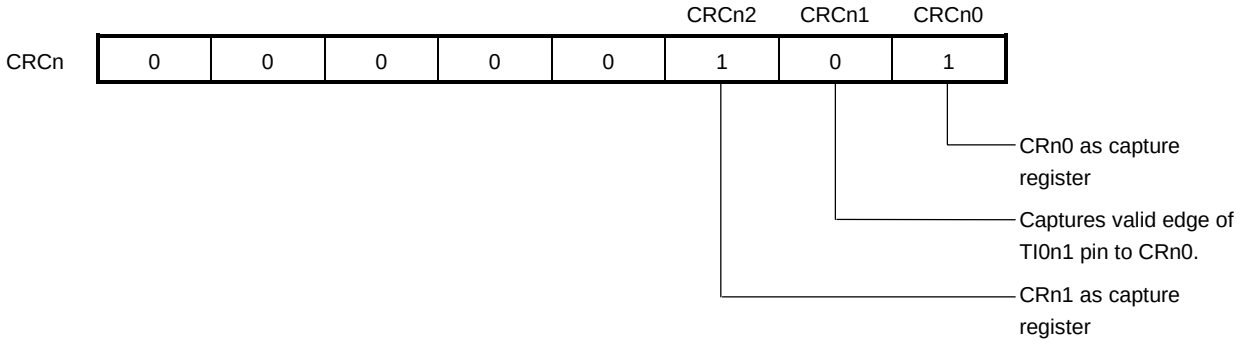
The valid edge of the TI0n0 pin is detected through sampling at a count clock cycle selected by the prescaler mode register n0, n1 (PRMn0, PRMn1), and the capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be removed.

Figure 7-65. Control Register Settings for Measurement of Two Pulse Widths with Free Running Counter

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



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

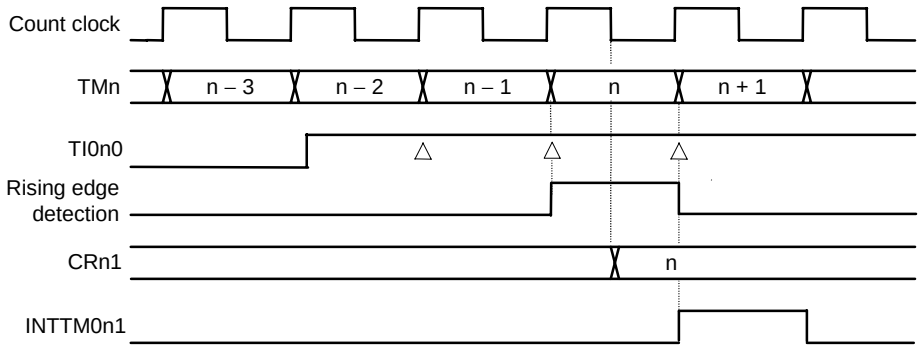


Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, refer to **Figures 7-51** and **7-52**.

- **Capture operation (free running mode)**

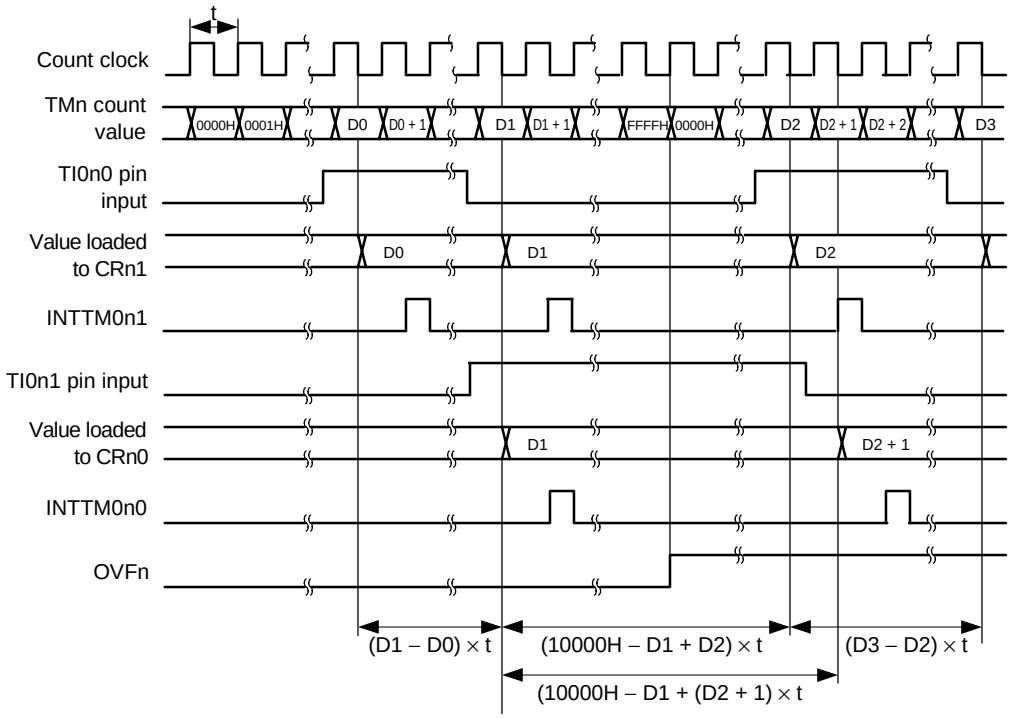
  The following figure illustrates the operation of the capture register when the capture trigger is input.

Figure 7-66. CRn1 Capture Operation with Rising Edge Specified



Remark n = 0, 1

Figure 7-67. Timing of Pulse Width Measurement with Free Running Counter (with Both Edges Specified)



Remark n = 0, 1

(3) Pulse width measurement with free running counter and two capture registers

When the 16-bit timer register n (TMn) is used as a free running counter (refer to **Figure 7-68**), the pulse width of the signal input to the TI0n0 pin can be measured.

When the edge specified by bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0) is input to the TI0n0 pin, the value of TMn is loaded to the 16-bit capture/compare register n1 (CRn1), and an external interrupt request signal (INTTM0n1) is set.

The value of TMn is also loaded to the 16-bit capture/compare register n0 (CRn0) when an edge reverse to the one that triggers capturing to CRn1 is input.

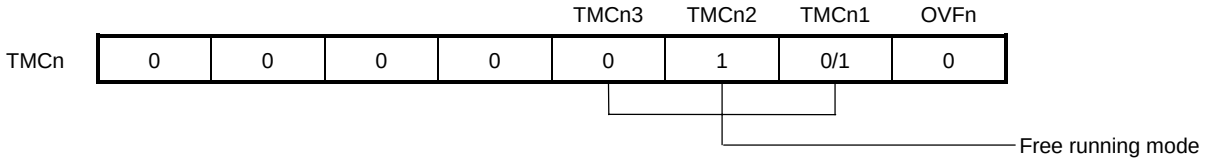
The edge of the TI0n0 pin is specified by bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0). The rising or falling edge can be specified.

The valid edge of TI0n0 is detected through sampling at a count clock cycle selected by the prescaler mode register n0, n1 (PRMn0, PRMn1), and the capture operation is not performed until the valid level is detected two times. Therefore, noise with a short pulse width can be removed.

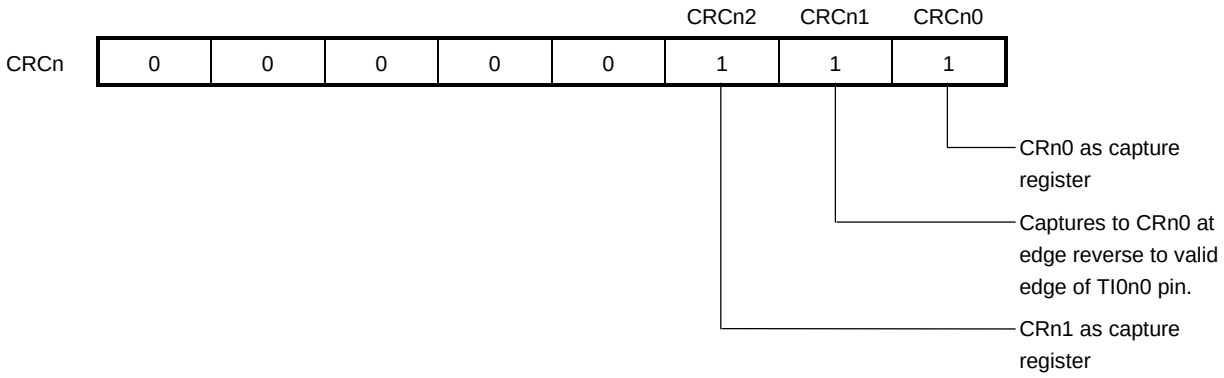
Caution If the valid edge of the TI0n0 pin is specified to be both the rising and falling edges, the capture/compare register n0 (CRn0) cannot perform its capture operation.

Figure 7-68. Control Register Settings for Pulse Width Measurement with Free Running Counter and Two Capture Registers

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

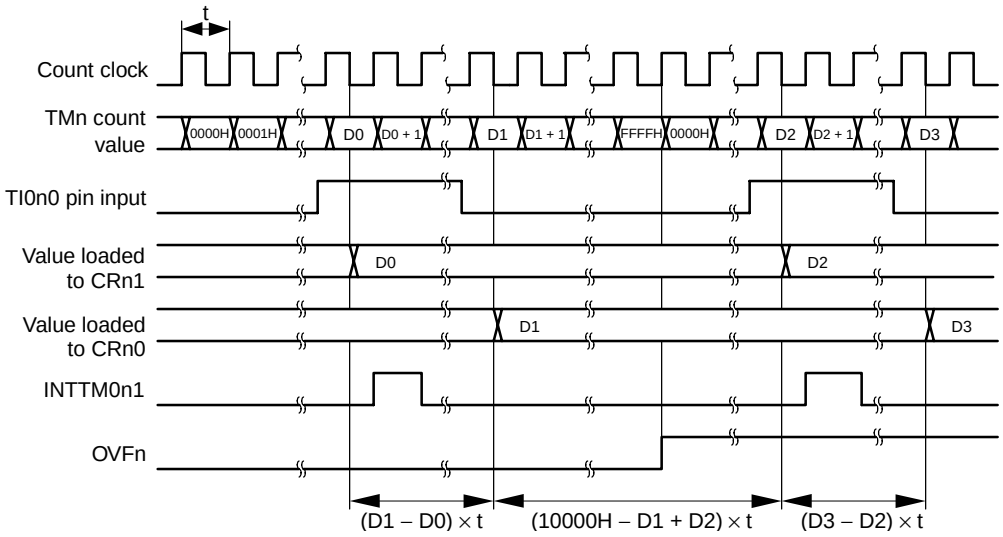


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, refer to **Figures 7-51** and **7-52**.

Figure 7-69. Timing of Pulse Width Measurement with Free Running Counter and Two Capture Registers (with Rising Edge Specified)



Remark n = 0, 1

(4) Pulse width measurement by restarting

When the valid edge of the TI0n0 pin is detected, the pulse width of the signal input to the TI0n0 pin can be measured by clearing the 16-bit timer register n (TMn) once and then resuming counting after loading the count value of TMn to the 16-bit capture/compare register n1 (CRn1). (Refer to **Figure 7-69.**)

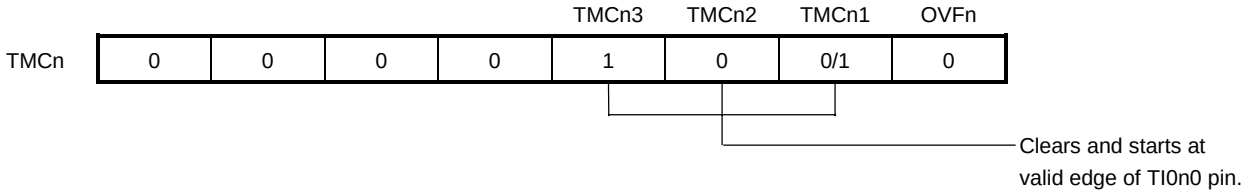
The edge is specified by bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0). The rising or falling edge can be specified.

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

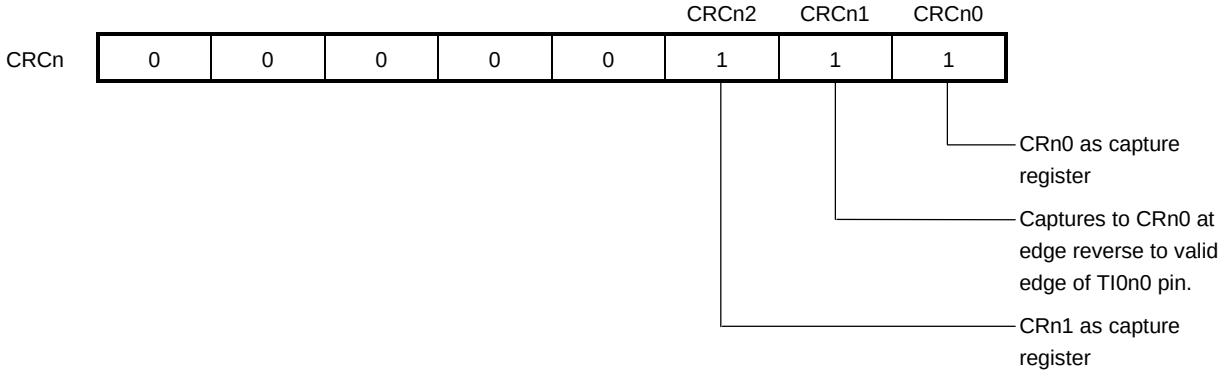
Caution If the valid edge of the TI0n0 pin is specified to be both the rising and falling edges, the capture/compare register n0 (CRn0) cannot perform its capture operation.

Figure 7-70. Control Register Settings for Pulse Width Measurement by Restarting

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

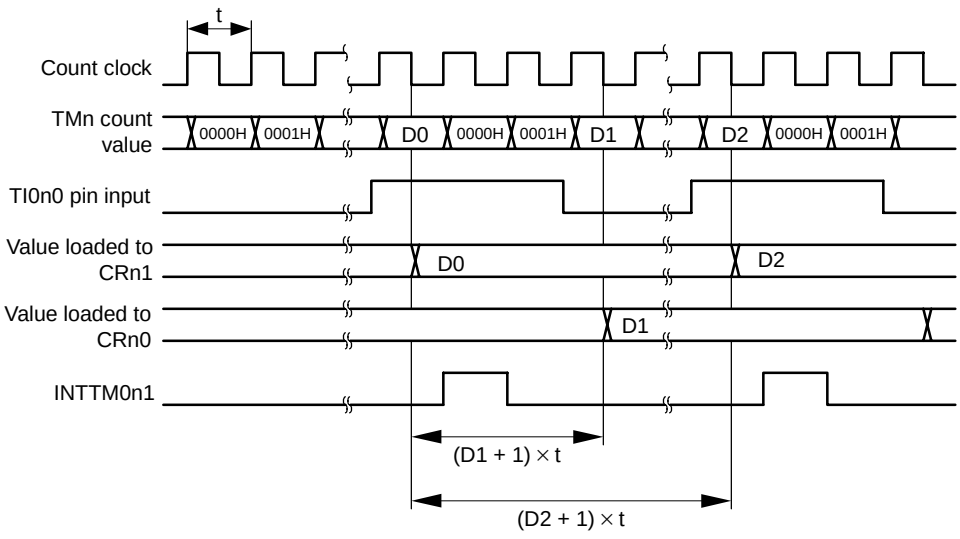


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the pulse width measurement function. For details, refer to **Figures 7-51** and **7-52**.

Figure 7-71. Timing of Pulse Width Measurement by Restarting (with Rising Edge Specified)



Remark n = 0, 1

7.7.4 Operation as external event counter

TMn can be used as an external event counter that counts the number of clock pulses input to the TI0n0 pin from an external source by using the 16-bit timer register n (TMn).

Each time the valid edge specified by the prescaler mode register n0 (PRMn0) has been input, TMn is incremented.

When the count value of TMn coincides with the value of the 16-bit capture/compare register n0 (CRn0), TMn is cleared to 0, and an interrupt request signal (INTTM0n0) is generated.

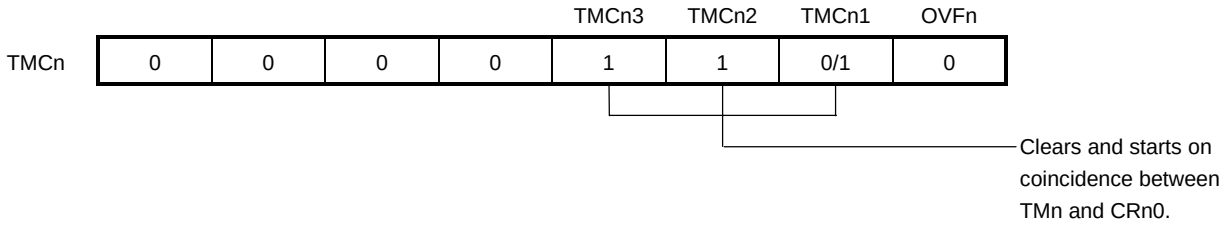
The edge is specified by bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0). The rising, falling, or both the rising and falling edges can be specified.

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

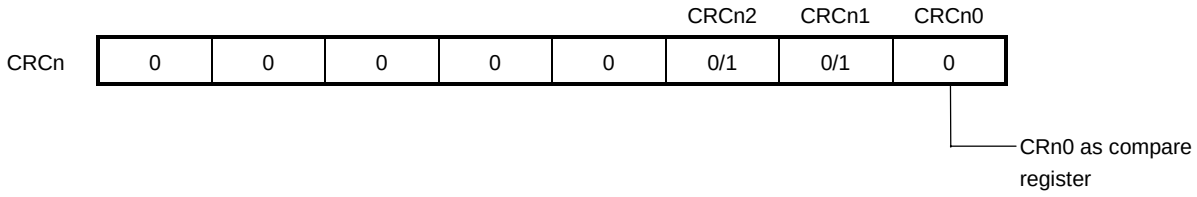
Remark n = 0, 1

Figure 7-72. Control Register Settings in External Event Counter Mode

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

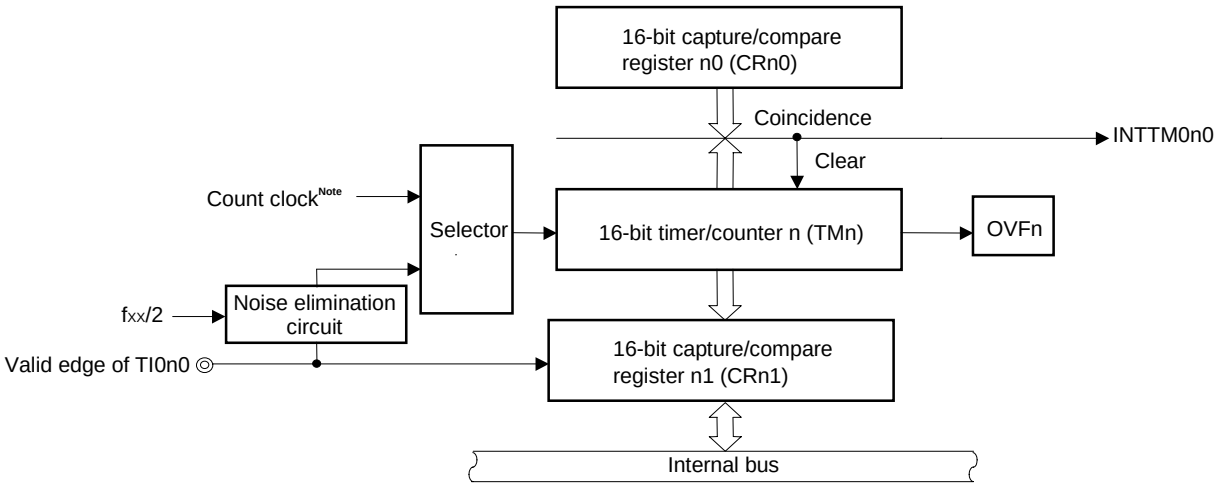


(b) Capture/compare control registers 0, 1 (CRC0, CRC1)



Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the external event counter function. For details, refer to **Figures 7-51** and **7-52**.

Figure 7-73. Configuration of External Event Counter

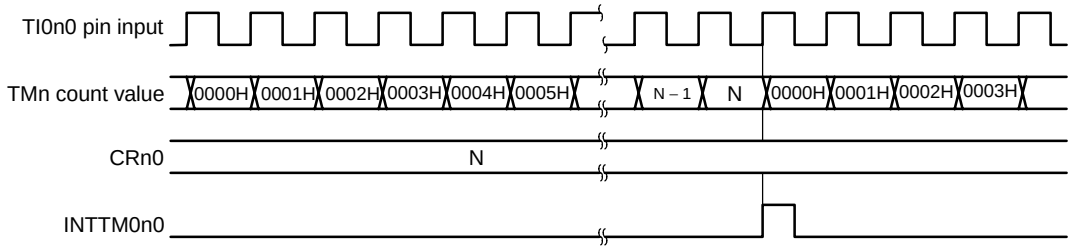


Note Count clock is set by the PRMn0 and PRMn1 registers.

Remarks 1. " ◎—— " indicates the signal that can be directly connected ports.

2. n = 0, 1

Figure 7-74. 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

7.7.5 Operation as square wave output

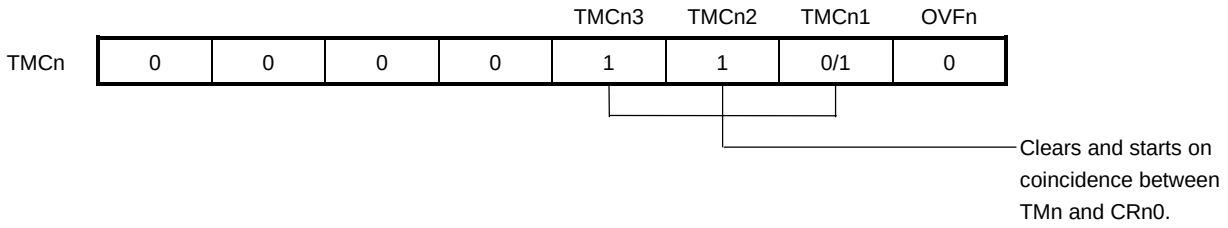
TMn can be used to output a square wave with any frequency at an interval specified by the count value set in advance to the 16-bit capture/compare register n0 (CRn0).

By setting bits 0 (TOEn) and 1 (TOCn1) of the 16-bit timer output control register n (TOCn) to 1, the output status of the TOn pin is reversed at an interval specified by the count value set in advance to CRn1. In this way, a square wave of any frequency can be output.

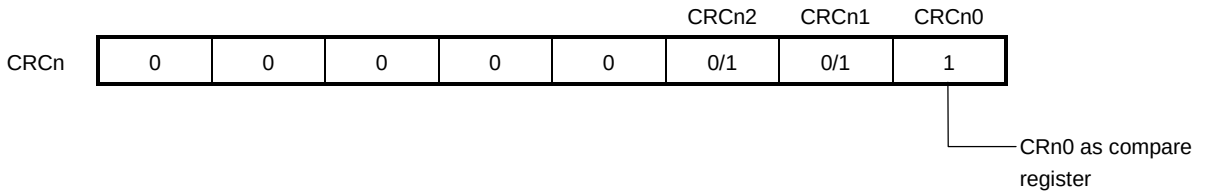
Remark n = 0, 1

Figure 7-75. Control Register Settings in Square Wave Output Mode

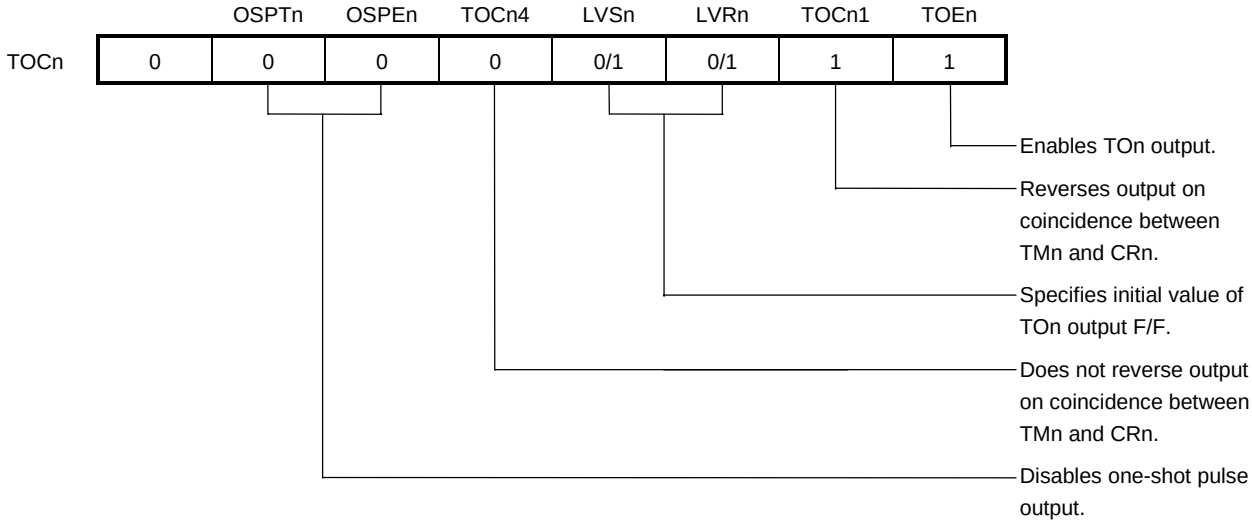
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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

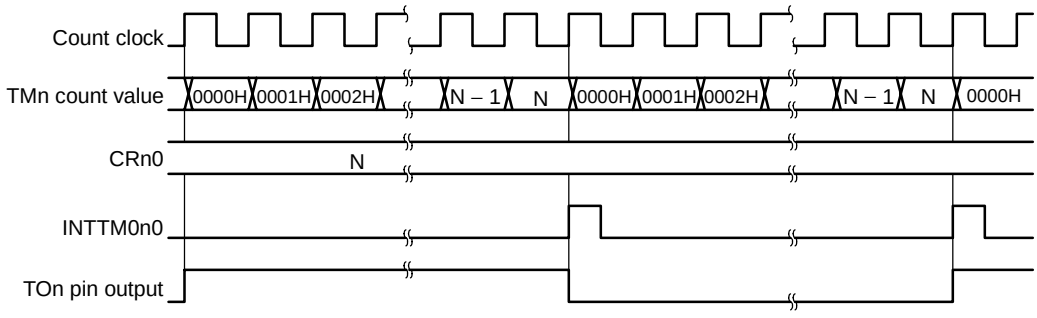


(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)



Remark 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 **Figures 7-51**, **7-52**, and **7-53**.

Figure 7-76. Timing of Square Wave Output Operation



Remark n = 0, 1

7.7.6 Operation as one-shot pulse output

TMn can output a one-shot pulse in synchronization with a software trigger and an external trigger (TI0n0 pin input).

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TOn pin by setting the 16-bit timer mode control register n (TMCn), capture/compare control register n (CRCn), and 16-bit timer output control register n (TOCn) as shown in Figure 7-77, and by setting bit 6 (OSPTn) of TOCn to 1 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 set in advance to the 16-bit capture/compare register n1 (CRn1). After that, the output is deasserted inactive at the count value set in advance to the 16-bit capture/compare register n0 (CRn0).

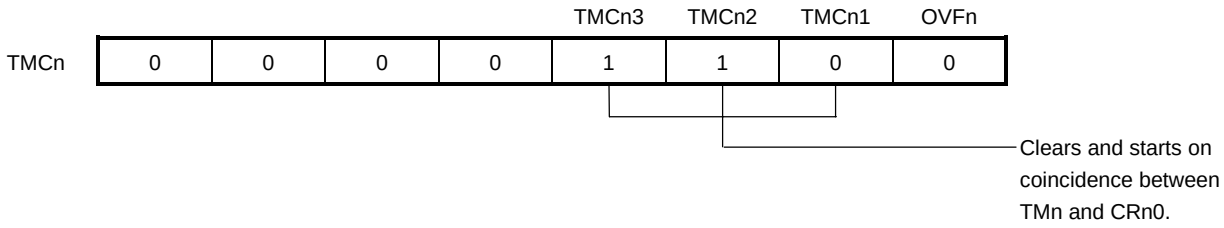
Even after the one-shot pulse has been output, TMn continues its operation. To stop TMn, TMCn must be reset to 00H.

Caution Do not set OSPTn to 1 while the one-shot pulse is being output. To output the one-shot pulse again, wait until INTTM0n0, which occurs on coincidence between TMn and CRn0, occurs.

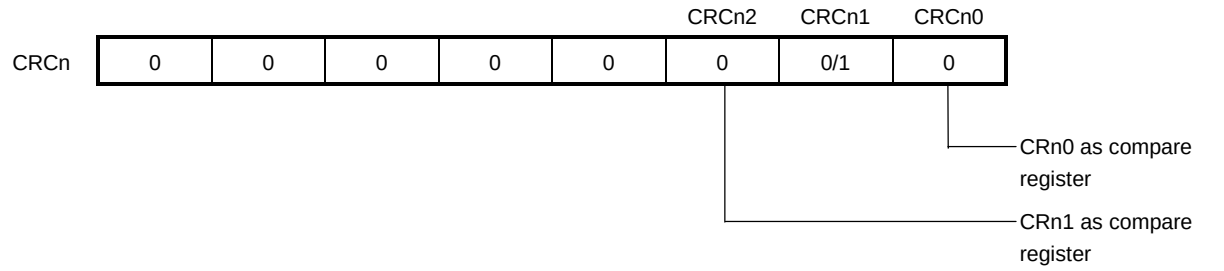
Remark n = 0, 1

Figure 7-77. Control Register Settings for One-Shot Pulse Output with Software Trigger

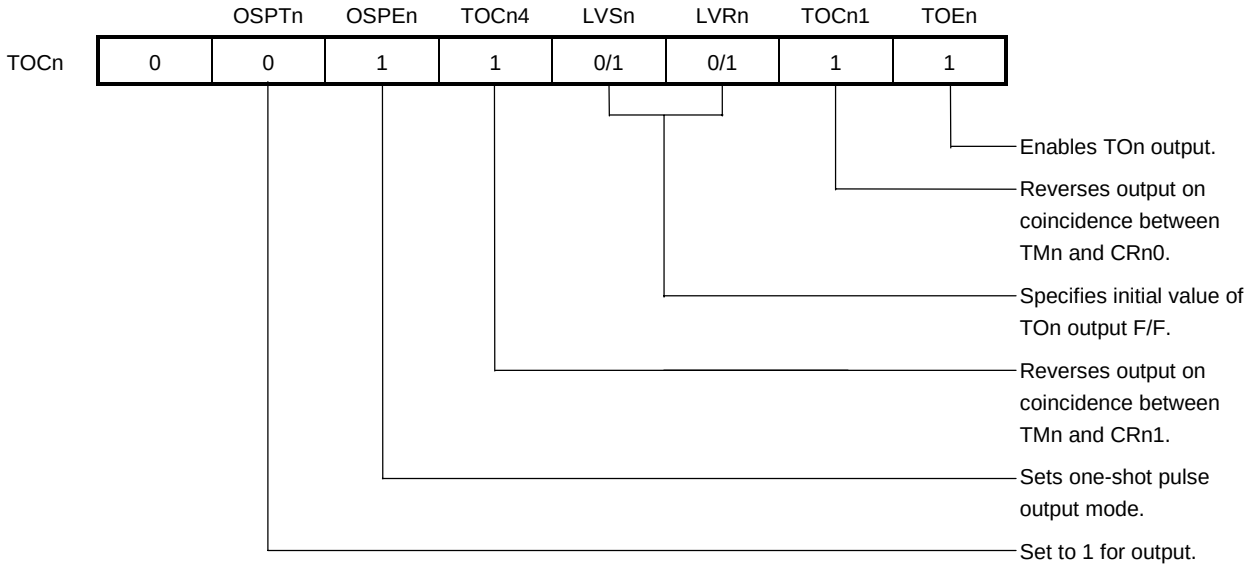
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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



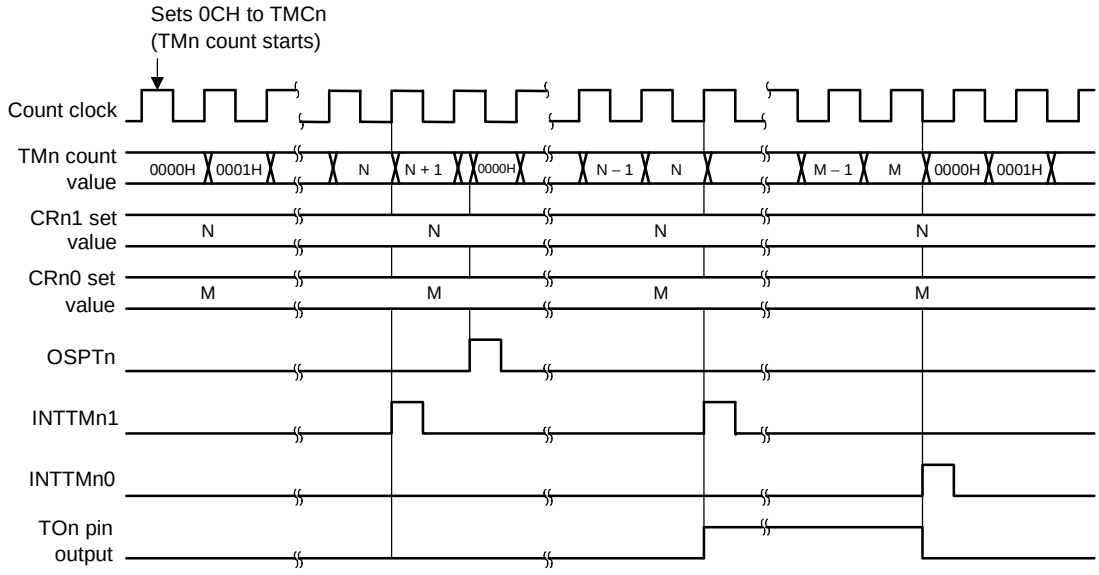
(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)



Caution Set a value in the following range to CRn0 and CRn1.
0000H < CRn1 < CRn0 ≤ FFFFH

Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the one-shot pulse output function. For details, refer to **Figures 7-51**, **7-52**, and **7-53**.

Figure 7-78. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution The 16-bit timer register n starts operating as soon as a value other than 0, 0 (operation stop mode) has been set to TMCn2 and TMCn3.

Remark n = 0, 1

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TOn0 pin by setting the 16-bit timer mode control register n (TMCn), capture/compare control register n (CRCn), and 16-bit timer output control register n (TOCn) as shown in Figure 7-79, and by using the valid edge of the TI0n0 pin as an external trigger.

The valid edge of the TI0n0 pin is specified by bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0). The rising, falling, or both the rising and falling edges can be specified.

When the valid edge of the TI0n0 pin is detected, the 16-bit timer/event counter is cleared and started, and the output is asserted active at the count value set in advance to the 16-bit capture/compare register n1 (CRn1).

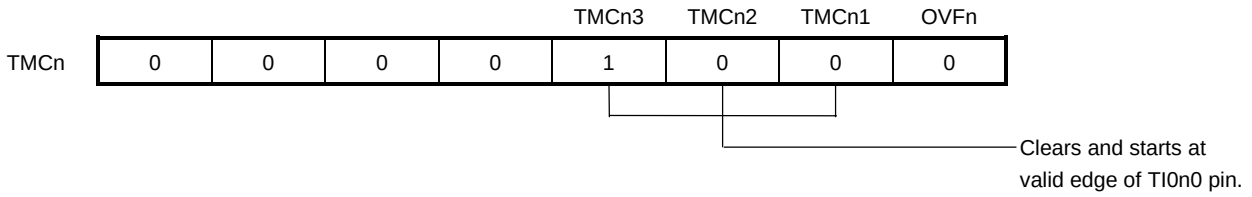
After that, the output is deasserted inactive at the count value set in advance to the 16-bit capture/compare register n0 (CRn0).

Caution Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.

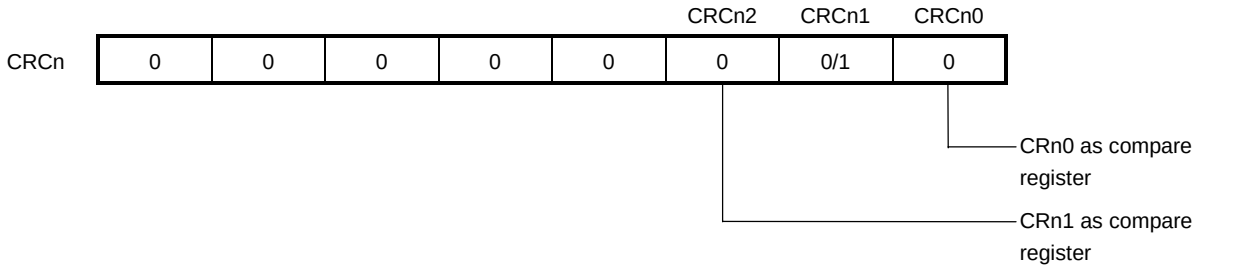
Remark n = 0, 1

Figure 7-79. Control Register Settings for One-Shot Pulse Output with External Trigger

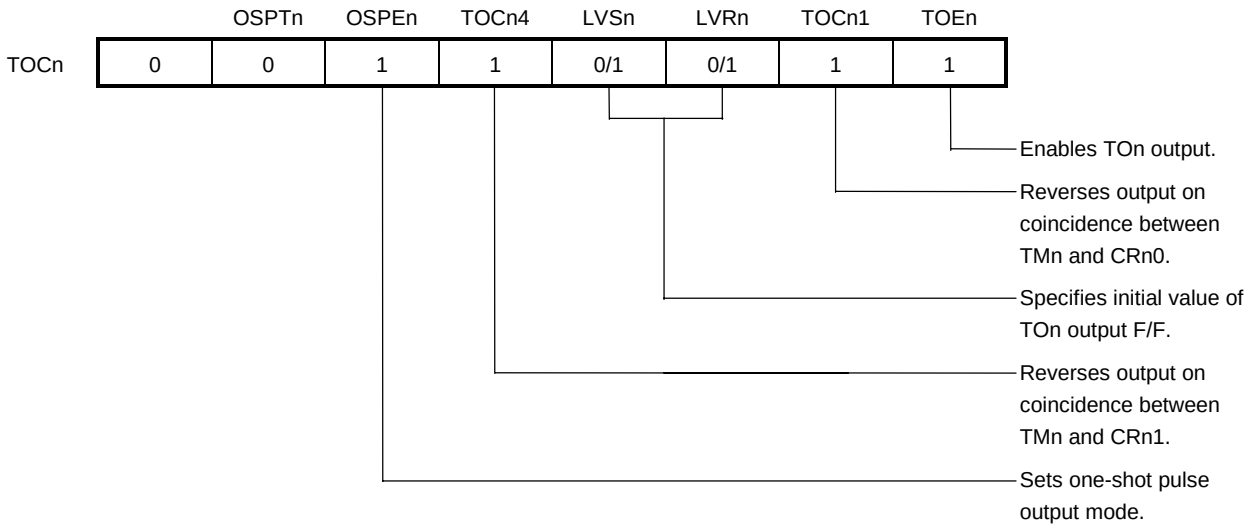
(a) 16-bit timer mode control registers 0, 1 (TMC0, TMC1)



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



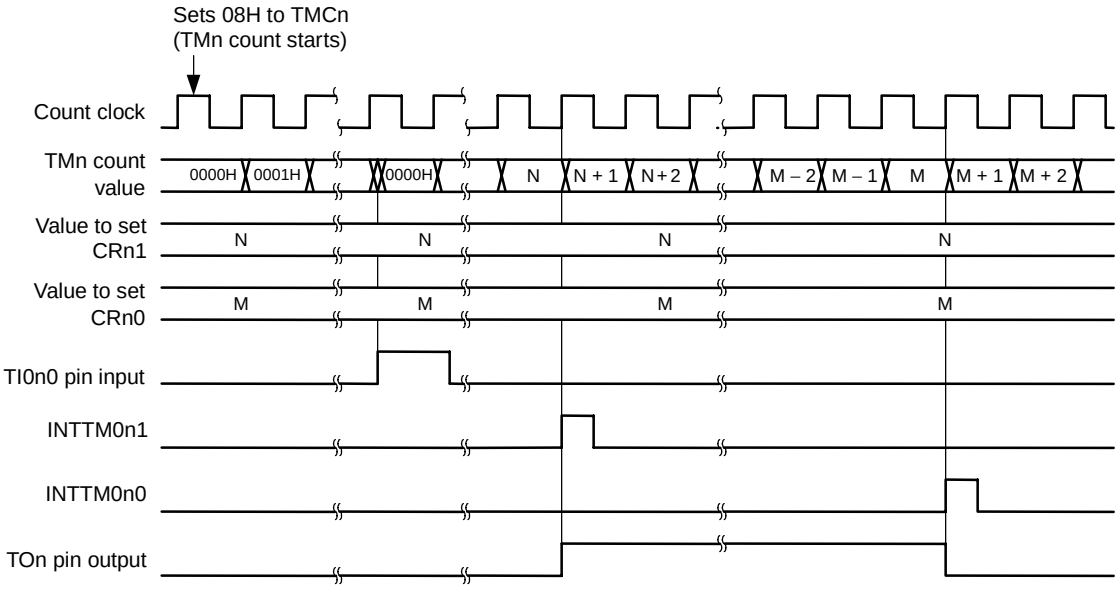
(c) 16-bit timer output control registers 0, 1 (TOC0, TOC1)



Caution Set a value in the following range to CRn0 and CRn1.
0000H < CRn1 < CRn0 ≤ FFFFH

Remark 0/1: When these bits are reset to 0 or set to 1, the other functions can be used along with the one-shot pulse output function. For details, refer to **Figures 7-51**, **7-52**, and **7-53**.

Figure 7-80. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution The 16-bit timer register n starts operating as soon as a value other than 0, 0 (operation stop mode) has been set to TMCn2 and TMCn3.

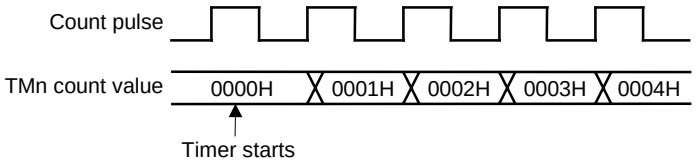
Remark n = 0, 1

7.7.7 Cautions

(1) Error on starting timer

An error of up to 1 clock occurs before the coincidence signal is generated after the timer has been started. This is because the 16-bit timer register n (TMn) is started asynchronously in respect to the count pulse.

Figure 7-81. Start Timing of 16-Bit Timer Register n



Remark n = 0, 1

(2) Setting the 16-bit capture/compare registers

Set a value other than 0000H in 16-bit capture/compare registers n0 and n1 (CRn0 and CRn1). The 16-bit capture/compare registers used as an event counter cannot perform a count operation for one pulse.

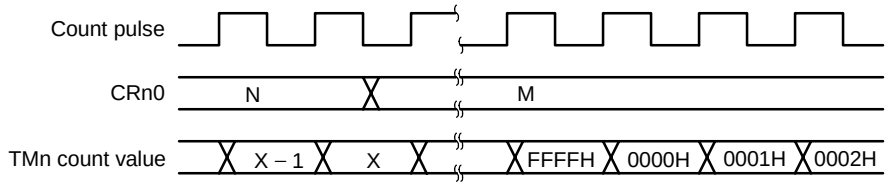
Remark n = 0, 1

(3) Setting compare register during timer count operation

If the value to which the current value of the 16-bit capture/compare register n0 (CRn0) has been changed is less than the value of the 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 restarted after the value of CRn0 has been changed.

Figure 7-82. Timing after Changing Compare Register during Timer Count Operation



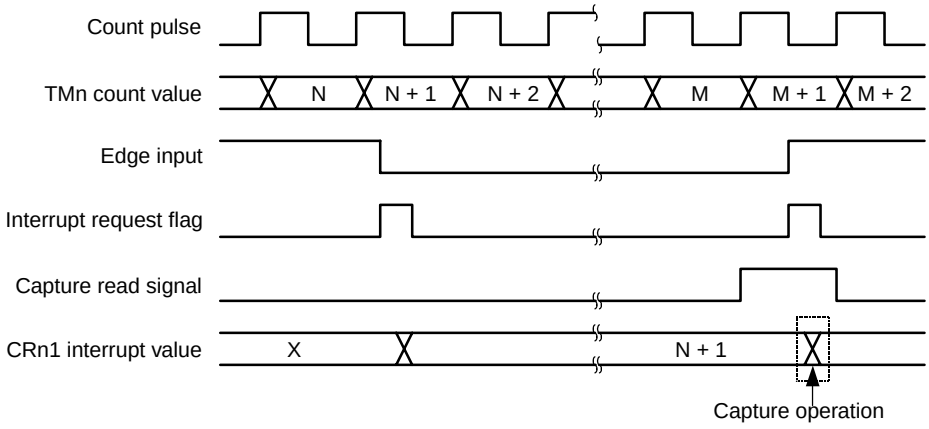
Remarks 1. N > X > M

2. n = 0, 1

(4) Data hold timing of capture register

If the valid edge is input to the TI0n0 pin while the 16-bit capture/compare register n1 (CRn1) is read, CRn1 performs the capture operation, but this capture value is not guaranteed. However, the interrupt request flag (INTTM0n1) is set as a result of detection of the valid edge.

Figure 7-83. Data Hold Timing of Capture Register



Remark n = 0, 1

(5) Setting valid edge

Before setting the valid edge of the TI0n0 pin, stop the timer operation by resetting bits 2 and 3 (TMCn2 and TMCn3) of the 16-bit timer mode control register n to 0, 0. Set the valid edge by using bits 4 and 5 (ESn00 and ESn01) of the prescaler mode register n0 (PRMn0).

Remark n = 0, 1

(6) Re-triggering one-shot pulse

<1> One-shot pulse output by software

When a one-shot pulse is output, do not set OSPTn to 1. Do not output the one-shot pulse again until INTTM0n0, which occurs on coincidence between TMn and CRn0, occurs.

Remark n = 0, 1

<2> One-shot pulse output with external trigger

If the external trigger occurs while a one-shot pulse is output, it is ignored.

(7) Operation of OVFn flag

<1> The OVFn flag is set to 1 in the following case:

Select mode in which TMn is cleared and started on coincidence between TMn and CRn0

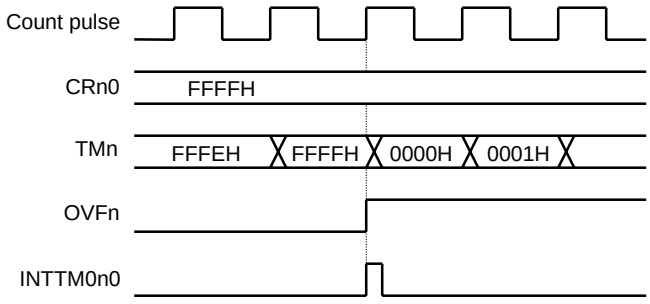
↓

Set CRn0 to FFFFH.

↓

When TMn counts up from FFFFH to 0000H

Figure 7-84. Operation Timing of OVFn Flag



Remark n = 0, 1

<2> If the OVFn flag is cleared before the next count clock (before TMn indicates 0001H) after TMn overflows, it is set again and the clear operation is ignored.

(8) Conflict operation

<1> If reading 16-bit capture/compare register n0 or n1 (CRn0 or CRn1) and capture trigger input (when CRn0 or CRn1 is used as a capture register) are in contention, the capture trigger input is given priority and CRn0 or CRn1 read data is undefined.

<2> If writing 16-bit capture/compare register n0 or n1 (CRn0 or CRn1) and coincidence with 16-bit timer register n (TMn) (when CRn0 or CRn1 is used as a compare register) are in contention, the coincidence is not determined normally. Do not write CRn0 or CRn1 around the coincidence timing.

Remark n = 0, 1

(9) Timer operation

<1> Reading 16-bit timer register n (TMn) does not cause capture operation for 16-bit capture/compare register n1 (CRn1).

<2> When the timer stops, any signals input to pins TI0n0 and TI0n1 are not accepted regardless of the CPU operation mode.

<3> The one-shot pulse output normally operates in free-running mode or in a mode in which the timer is cleared and started at the valid edge of the TI0n0 pin. The one-shot pulse output cannot be performed in a mode in which the timer is cleared and started upon the coincidence between TMn and CRn0 because no overflow occurs.

Remark n = 0, 1

(10) Capture operation

<1> If the valid edge of TI0n0 is specified as the count clock, a capture register with TI0n0 specified as the trigger cannot operate normally.

<2> If both rising and falling edges are selected as the valid edge of TI0n0, no capture operation is performed.

<3> The capture trigger requires a pulse longer than two count clocks selected using prescaler mode register n0 or n1 (PRMn0 or PRMn1) to ensure capture operation for the signal from TI0n1 or TI0n0.

<4> Capture operation is performed at the falling edge of the count clock. The interrupt request input (INTTM0n0 or INTTM0n1) is generated at the falling edge of the next count clock.

Remark n = 0, 1

(11) Compare operation

<1> 16-bit capture/compare register n0 or n1 (CRn0 or CRn1) may be rewritten while the timer is operating. In this case, if the value is near to and greater than the timer value, the coincidence interrupt request may not be generated normally or clear operation may not be performed normally.

<2> For CRn0 or CRn1 set in compare mode, no capture operation is performed even if the capture trigger is input.

Remark n = 0, 1

(12) Edge detection

<1> If 16-bit timer/counter n (TMn) operation is enabled by specifying the rising edge or both edges for the valid edge of the TI0n0 or TI0n1 pin while the TI0n0 or TI0n1 pin is high, immediately after a system reset, the rising edge is detected immediately after the rising edge or both edges is specified. Be careful when pulling up pin TI0n0 or TI0n1. However, the rising or falling edge is not detected when operation is enabled after it has been stopped.

<2> The sampling clock for noise elimination differs depending on whether to use the valid edge of TI0n0 as the count clock or capture trigger. The valid edge is sampled at $f_{XX}/2$ for the former or at the count clock selected by prescaler mode register n0 or n1 (PRMn0 or PRMn1) for the latter. A capture operation is not performed until the valid edge is sampled and detected twice. Therefore, noise with a short pulse width can be eliminated.

Remark n = 0, 1

7.8 8-Bit Timer (TM2 to TM7, TM10, and TM11)

7.8.1 Functions

8-bit timer n has the following two modes (n = 2 to 7, 10, 11).

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

Caution When a cascade connection is not used, access to TM23, TM45, TM67, TM1011, CR23, CR45, CR67, and CR1011 is prohibited.

These two modes are described next.

(1) Mode using timer alone (individual mode)

The timer operates as an 8-bit timer/event counter.
It can have the following functions.

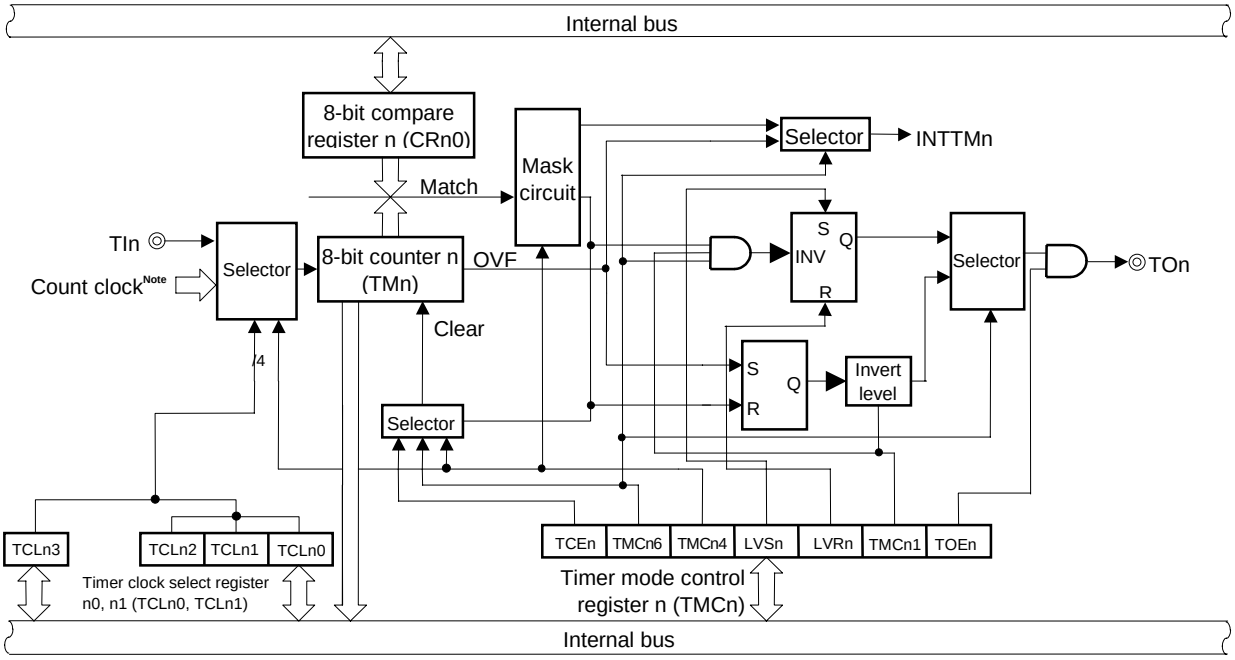
- Interval timer
- External event counter
- Square wave output
- PWM output

(2) Mode using the cascade connection (16-bit resolution: cascade connection mode)

The timer operates as a 16-bit timer/event counter by connecting TM2 and TM3, TM4 and TM5, TM6 and TM7, or TM10 and TM11 in cascade.
It can have the following functions.

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

Figure 7-85. Block Diagram of TM2 to TM7, TM10, and TM11



Note Count clock is set by the TCLn register.

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

2. n = 2 to 7, 10, 11

7.8.2 Configuration

Timer n is constructed from the following hardware.

Table 7-12. Configuration of Timers 2 to 7, 10, and 11

Item	Configuration
Timer registers	8-bit counter n (TMn) 16-bit counter m (TMm): Only when connecting in cascade
Registers	8-bit compare register n (CRn0) 16-bit compare register m (CRm): Only when connecting in cascade
Timer outputs	TOn
Control registers	Timer clock select registers n0 and n1 (TCLn0 and TCLn1) 8-bit timer mode control register n (TMCn)

Remark n = 2 to 7, 10, 11

m = 23, 45, 67, 1011

(1) 8-bit counters 2 to 7, 10, and 11 (TM2 to TM7, TM10, and TM11)

TMn is an 8-bit read-only register that counts the count pulses.

The counter is incremented synchronous to the rising edge of the count clock.

TM2 connected with TM3, TM4 connected with TM5, TM6 connected with TM7, or TM10 connected with TM11 in cascade can be used as a 16-bit timer.

When TMm connected with TMm + 1 in cascade is used as a 16-bit timer, it can be read using a 16-bit memory manipulation instruction. However, the 16-bit timer is read twice by dividing it into TMm and TMm + 1 because it is connected to an internal 8-bit bus. For this reason, read the 16-bit timer twice and compare the read results while considering a count change.

When the count is read out during operation, the count clock input temporarily stops and the count is read at that time. In the following cases, the count becomes 00H.

(1) $\overline{\text{RESET}}$ is input.
(2) TCEn is cleared.
(3) TMn and CRn0 match in the clear and start mode that occurs when TMn and CRn0 match.

Caution When connected in cascade, these registers become 00H even when TCEn in the lowest-order timer (TM2, TM4, TM6, or TM10) is cleared.

Remark n = 2 to 7, 10, 11
m = 2, 4, 6, 10

(2) 8-bit compare registers 2 to 7, 10, and 11 (CR20 to CR70, CR100, and CR110)

The value set in CRn0 is always compared to the count in the 8-bit counter n (TMn). If the two values match, an interrupt request (INTTMn) is generated (except in the PWM mode).

The value of CRn0 can be set in the range of 00H to FFH, and can be written during counting.

When TMm connected with TMm + 1 in cascade is used as a 16-bit timer, CRm0 and CR(m + 1)0 operate as a 16-bit compare register. The 16-bit counter value is compared with the 16-bit register value. When these values coincide, an interrupt request (INTTMm) is generated. At this time, the INTTMm + 1 interrupt request is also generated. When using TMm and TMm + 1, connected in cascade, mask the INTTMm + 1 interrupt request.

Caution If data is set in a cascade connection, always set after stopping the timer.

Remark n = 2 to 7, 10, 11
m = 2, 4, 6, 10

7.8.3 Timer n control register

The following two types of registers control timer n.

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

(1) Timer clock select registers 20 to 70, 100, 110, 21 to 71, 101, and 111 (TCL20 to TCL70, TCL100, TCL110, TCL21 to TCL71, TCL101, and TCL111)

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.

Remark n = 2 to 7, 10, 11

Figure 7-86. TM2, TM3 Timer Clock Select Registers 20, 21, 30, and 31 (TCL20, TCL21, TCL30, and TCL31)

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

	7	6	5	4	3	2	1	0
TCLn0	0	0	0	0	0	TCLn2	TCLn1	TCLn0

(n = 2, 3)

After reset: 00H R/W Address: FFFFF24EH, FFFFF25EH

	7	6	5	4	3	2	1	0
TCLn1	0	0	0	0	0	0	0	TCLn3

(n = 2, 3)

TCLn3	TCLn2	TCLn1	TCLn0	Count Clock Selection			
				Count Clock	f_{XX}		
					20 MHz	10 MHz	2 MHz
0	0	0	0	TIn falling edge	–	–	–
0	0	0	1	TIn rising edge	–	–	–
0	0	1	0	$f_{XX}/4$	200 ns	400 ns	2 μs
0	0	1	1	$f_{XX}/8$	400 ns	800 ns	4 μs
0	1	0	0	$f_{XX}/16$	800 ns	1.6 μs	8 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	3.2 μs	16 μs
0	1	1	0	$f_{XX}/128$	6.4 μs	12.8 μs	64 μs
0	1	1	1	$f_{XX}/512$	25.6 μs	51.2 μs	256 μs
1	0	0	0	Setting prohibited	–	–	–
1	0	0	1	Setting prohibited	–	–	–
1	0	1	0	$f_{XX}/64$	3.2 μs	6.4 μs	32 μs
1	0	1	1	$f_{XX}/256$	12.8 μs	25.6 μs	128 μs
1	1	0	0	Setting prohibited	–	–	–
1	1	0	1	Setting prohibited	–	–	–
1	1	1	0	Setting prohibited	–	–	–
1	1	1	1	Setting prohibited	–	–	–

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

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

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

Figure 7-87. TM4, TM5 Timer Clock Select Registers 40, 41, 50, and 51 (TCL40, TCL41, TCL50, and TCL51)

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

	7	6	5	4	3	2	1	0
TCLn0	0	0	0	0	0	TCLn2	TCLn1	TCLn0

(n = 4, 5)

After reset: 00H R/W Address: FFFFF26EH, FFFFF27EH

	7	6	5	4	3	2	1	0
TCLn1	0	0	0	0	0	0	0	TCLn3

(n = 4, 5)

TCLn3	TCLn2	TCLn1	TCLn0	Count Clock Selection			
				Count Clock	f_{XX}		
					20 MHz	10 MHz	2 MHz
0	0	0	0	TIn falling edge	–	–	–
0	0	0	1	TIn rising edge	–	–	–
0	0	1	0	$f_{XX}/4$	200 ns	400 ns	2 μs
0	0	1	1	$f_{XX}/8$	400 ns	800 ns	4 μs
0	1	0	0	$f_{XX}/16$	800 ns	1.6 μs	8 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	3.2 μs	16 μs
0	1	1	0	$f_{XX}/128$	6.4 μs	12.8 μs	64 μs
0	1	1	1	f_{XT} (Sub clock)	30.5 μs	30.5 μs	30.5 μs
1	0	0	0	Setting prohibited	–	–	–
1	0	0	1	Setting prohibited	–	–	–
1	0	1	0	$f_{XX}/64$	3.2 μs	6.4 μs	32 μs
1	0	1	1	$f_{XX}/256$	12.8 μs	25.6 μs	128 μs
1	1	0	0	Setting prohibited	–	–	–
1	1	0	1	Setting prohibited	–	–	–
1	1	1	0	Setting prohibited	–	–	–
1	1	1	1	Setting prohibited	–	–	–

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

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

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

Figure 7-88. TM6, TM7 Timer Clock Select Registers 60, 61, 70, and 71 (TCL60, TCL61, TCL70, and TCL71)

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

	7	6	5	4	3	2	1	0
TCLn0	0	0	0	0	0	TCLn2	TCLn1	TCLn0

(n = 6, 7)

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

	7	6	5	4	3	2	1	0
TCLn1	0	0	0	0	0	0	0	TCLn3

(n = 6, 7)

TCLn3	TCLn2	TCLn1	TCLn0	Count Clock Selection			
				Count Clock	f_{XX}		
					20 MHz	10 MHz	2 MHz
0	0	0	0	Setting prohibited	–	–	–
0	0	0	1	Setting prohibited	–	–	–
0	0	1	0	$f_{XX}/4$	200 ns	400 ns	2 μs
0	0	1	1	$f_{XX}/8$	400 ns	800 ns	4 μs
0	1	0	0	$f_{XX}/16$	800 ns	1.6 μs	8 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	3.2 μs	16 μs
0	1	1	0	$f_{XX}/64$	6.4 μs	12.8 μs	64 μs
0	1	1	1	$f_{XX}/128$	–	–	–
1	0	0	0	Setting prohibited	–	–	–
1	0	0	1	Setting prohibited	–	–	–
1	0	1	0	$f_{XX}/256$	12.8 μs	25.6 μs	128 μs
1	0	1	1	$f_{XX}/512$	25.6 μs	51.2 μs	256 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. When TCLn0 and TCLn1 are overwritten by different data, write after temporarily stopping the timer.

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

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

Figure 7-89. TM10, TM11 Timer Clock Select Registers 100, 101, 110, and 111 (TCL100, TCL101, TCL110, and TCL111)

After reset: 00H R/W Address: FFFFF3E4H, FFFFF3F4H

	7	6	5	4	3	2	1	0
TCLn0	0	0	0	0	0	TCLn2	TCLn1	TCLn0

(n = 10, 11)

After reset: 00H R/W Address: FFFFF3EEH, FFFFF3FEH

	7	6	5	4	3	2	1	0
TCLn1	0	0	0	0	0	0	0	TCLn3

(n = 10, 11)

TCLn3	TCLn2	TCLn1	TCLn0	Count Clock Selection			
				Count Clock	f_{XX}		
					20 MHz	10 MHz	2 MHz
0	0	0	0	Setting prohibited	–	–	–
0	0	0	1	Setting prohibited	–	–	–
0	0	1	0	$f_{XX}/4$	200 ns	400 ns	2 μs
0	0	1	1	$f_{XX}/8$	400 ns	800 ns	4 μs
0	1	0	0	$f_{XX}/16$	800 ns	1.6 μs	8 μs
0	1	0	1	$f_{XX}/32$	1.6 μs	3.2 μs	16 μs
0	1	1	0	$f_{XX}/64$	3.2 μs	6.4 μs	32 μs
0	1	1	1	$f_{XX}/128$	6.4 μs	12.8 μs	64 μs
1	0	0	0	Setting prohibited	–	–	–
1	0	0	1	Setting prohibited	–	–	–
1	0	1	0	$f_{XX}/256$	12.8 μs	25.6 μs	128 μs
1	0	1	1	$f_{XX}/512$	25.6 μs	51.2 μs	256 μs
1	1	0	0	Setting prohibited	–	–	–
1	1	0	1	Setting prohibited	–	–	–
1	1	1	0	Setting prohibited	–	–	–
1	1	1	1	Hsync (HSOUT0) input for compensation	–	–	–

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

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

Remark When connected in cascade, the settings of TCL113 to TCL110 in TM11 are invalid.

(2) 8-bit timer mode control registers 2 to 7, 10, and 11 (TMC2 to TMC7, TMC10, and TMC11)

The TMCn register makes the following six settings.

(1) Controls the counting by the 8-bit counter n (TMn)
(2) Selects the operating mode of the 8-bit counter n (TMn)
(3) Selects the individual mode or cascade connection mode
(4) Sets the state of the timer output flip-flop
(5) Controls the timer flip-flop or selects the active level in the PWM (free running) mode
(6) Controls timer output

TMCn is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets these registers to 04H.

Figure 7-90. 8-Bit Timer Mode Control Registers 2 to 7, 10, and 11 (TMC2 to TMC7, TMC10, and TMC11)

After reset: 04H R/W Address: TMC2 FFFFF246H TMC6 FFFFF286H
TMC3 FFFFF256H TMC7 FFFFF296H
TMC4 FFFFF266H TMC10 FFFFF3E6H
TMC5 FFFFF276H TMC11 FFFFF3F6H

	7	6	5	4	3	2	1	0
TMCn	TCEn	TMCn6	0	TMCn4	LVSn	LVRn	TMCn1	TOEn

(n = 2 to 7, 10, 11)

TCEn	TMn Count Operation Control
0	Counting is disabled after the counter is cleared to 0 (prescaler disabled)
1	Start count operation

TMCn6	TMn Operating Mode Selection
0	Clear & Start mode when TMn and CRn0 match
1	PWM (free running) mode

TMCn4	Individual Mode or Cascade Connection Mode Selection
0	Individual mode (fixed to 0 when n = 2, 4, 6, 10)
1	Cascade connection mode (connection to low-order timer)

LVSn	LVRn	Setting State of Timer Output Flip-Flop
0	0	Not change
0	1	Reset timer output flip-flop to 0
1	0	Set timer output flip-flop to 1
1	1	Setting prohibited

TMCn1	Other than PWM (Free Running) Mode (TMCn6 = 0)	PWM (Free Running) Mode (TMCn6 = 1)
	Controls Timer F/F	Selects Active Level
0	Disable inversion operation	Active high
1	Enable inversion operation	Active low

TOEn	Timer Output Control
0	Disable output (port mode)
1	Enable output

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

2. Since TOn and TIn share the same pin, either function can only be used.

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

2. If LVSn and LVRn are read after setting data, 0 is read.

7.9 8-Bit Timer Operation

7.9.1 Operation as interval timer (8 bits)

The timer operates as an interval timer that repeatedly generates interrupts at the interval of the preset count in the 8-bit compare register n (CRn0).

If the count in the 8-bit counter n (TMn) matches the value set in CRn0, simultaneous to clearing the value of TMn to 0 and continuing the count, the interrupt request signal (INTTMn) is generated.

The TMn count clock can be selected by bits 0 to 2 (TCLn0 to TCLn2) in the timer clock select register n0 (TCLn0) and by bit 0 (TCLn3) in timer clock select register n1 (TCLn1) (n = 2 to 7, 10, 11).

Setting method

(1) Set each register.
- TCLn0, TCLn1: Selects the count clock.
- CRn0: Compare value
- TMCn: Selects the clear and start mode when TMn and CRn0 match.
  (TMCn = 0000xxx0B, x is don't care)

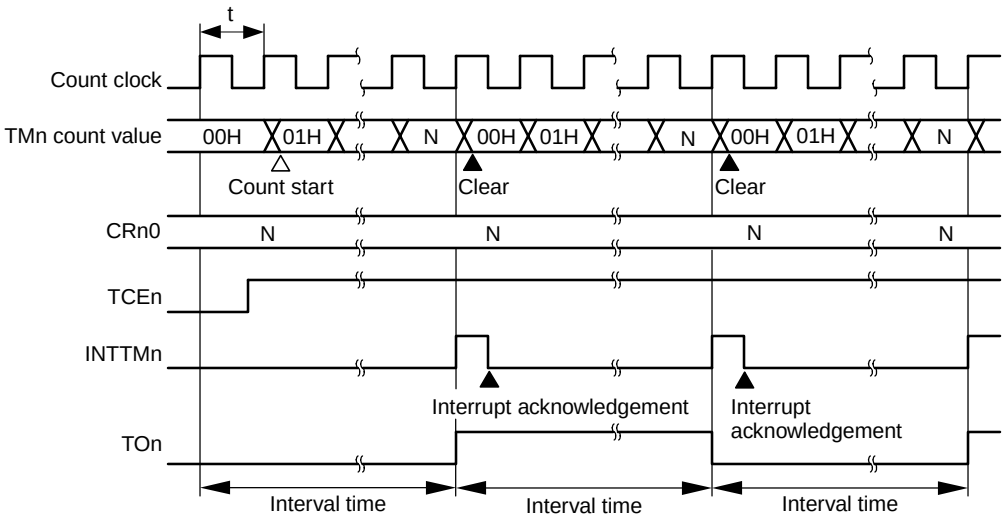
(2) When TCEn = 1 is set, counting starts.

(3) When the values of TMn and CRn0 match, INTTMn is generated (TMn is cleared to 00H).

(4) Then, INTTMn is repeatedly generated during the same interval. When counting stops, set TCEn = 0.

Figure 7-91. Timing of Interval Timer Operation (1/3)

Basic operation

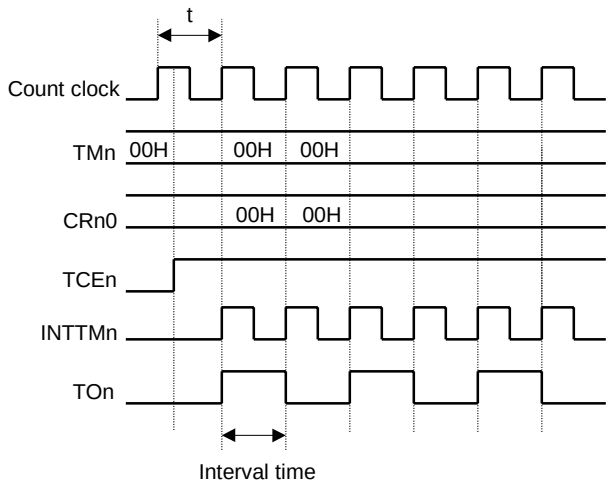


Remarks 1. Interval time = (N + 1) × t; N = 00H to FFH

2. n = 2 to 7, 10, 11

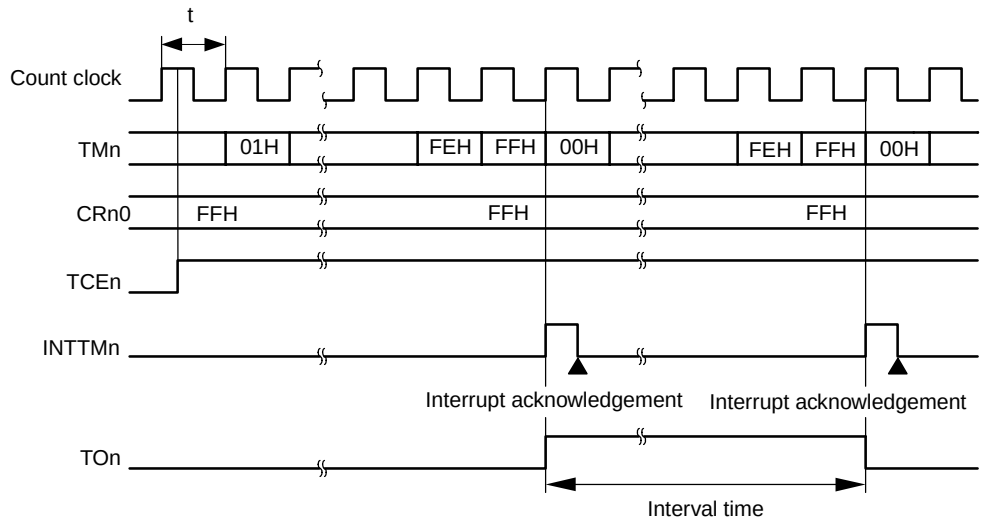
Figure 7-91. Timing of Interval Timer Operation (2/3)

When CRn0 = 00H



Remark n = 2 to 7, 10, 11

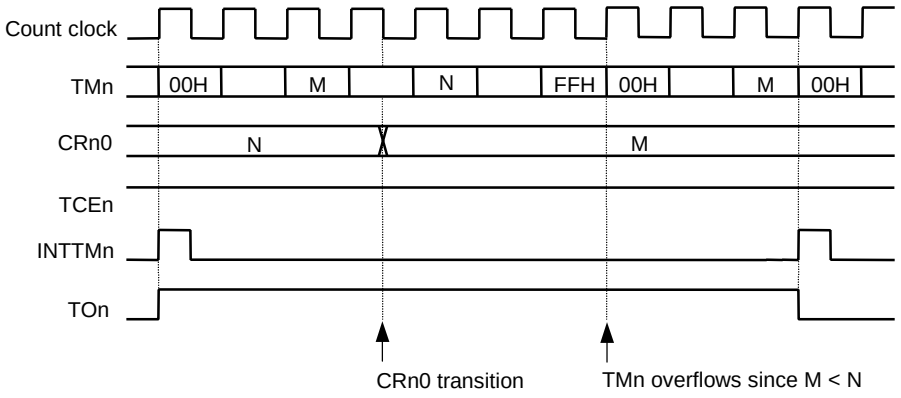
When CRn0 = FFH



Remark n = 2 to 7, 10, 11

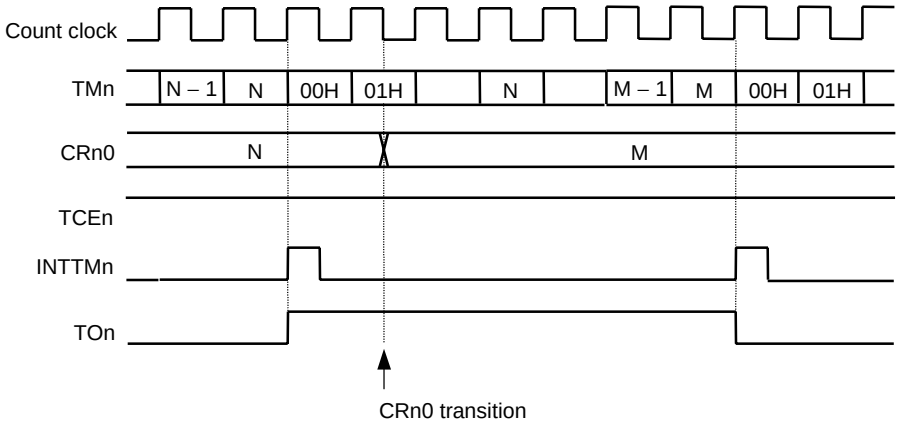
Figure 7-91. Timing of Interval Timer Operation (3/3)

Operated by CRn0 transition (M < N)



Remark n = 2 to 7, 10, 11

Operated by CRn0 transition (M > N)



Remark n = 2 to 7, 10, 11

7.9.2 Operation as external event counter

The external event counter counts the number of external clock pulses that are input to TIn.

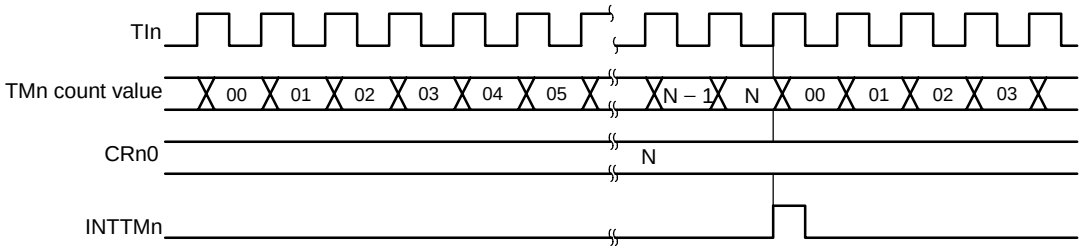
Each time a valid edge specified with the timer clock select register n0, n1 (TCLn0, TCLn1) is input, TMn is incremented. The edge setting can be selected to be either a rising or falling edge.

If the total of TMn and the value of the 8-bit compare register n (CRn0) match, TMn is cleared to 0 and the interrupt request signal (INTTMn) is generated.

INTTMn is generated each time the TMn value matches the CRn0 value.

Remark n = 2 to 7, 10, 11

Figure 7-92. Timing of External Event Counter Operation (When Rising Edge Is Set)



Remark n = 2 to 7, 10, 11

7.9.3 Operation as square wave output (8-bit resolution)

A square wave having any frequency is output at the interval preset in the 8-bit compare register n (CRn0).

By setting bit 0 (TOEn) of the 8-bit timer mode control register n (TMCn) to 1, the output state of TOn is inverted with the count preset in CRn0 as the interval. Therefore, a square wave output having any frequency (duty factor = 50%) is possible.

Setting method

(1) Set the registers.
- Sets the port latch and port mode register to 0
- TCLn0, TCLn1: Selects the count clock
- CRn0: Compare value
- TMCn: Clear and start mode when TMn and CRn0 match

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

Inversion of timer output flip-flop enabled
Timer output enabled → TOEn = 1

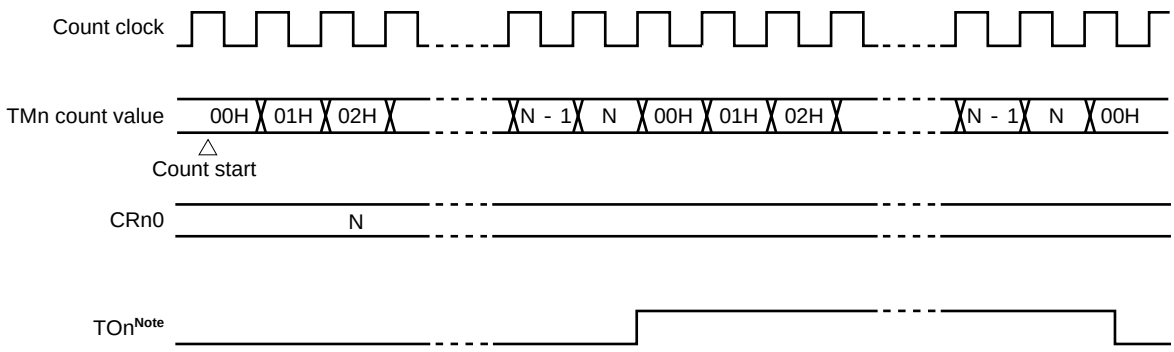
(2) When TCEn = 1 is set, the counter starts operating.

(3) If the values of TMn and CRn0 match, the timer output flip-flop inverts. Also, INTTMn is generated and TMn is cleared to 00H.

(4) Then, the timer output flip-flop is inverted for the same interval to output a square wave from TOn.

Remark n = 2 to 7, 10, 11

Figure 7-93. Square Wave Output Operation Timing



Note TOn output initial value can be set by bits 2 and 3 (LVRn and LVSn) of the 8-bit timer mode control register n (TMCn).

Remark n = 2 to 7, 10, 11

7.9.4 Operation as 8-bit PWM output

By setting bit 6 (TMCn6) of the 8-bit timer mode control register n (TMCn) to 1, the timer operates as a PWM output.

Pulses with the duty factor determined by the value set in the 8-bit compare register n (CRn0) is output from TOn.

Set the width of the active level of the PWM pulse in CRn0. The active level can be selected by bit 1 (TMCn1) in TMCn.

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 (TOEn) of TMCn.

Caution CRn0 can be rewritten only once in one period while in the PWM mode.

Remark n = 2 to 7, 10, 11

(1) Basic operation of the PWM output

Setting method

(1) Set the port latch and port mode register n to 0.
(2) Set the active level width in the 8-bit compare register n (CRn0).
(3) Select the count clock with the timer clock select register n0, n1 (TCLn0, TCLn1).
(4) Set the active level in bit 1 (TMCn1) of TMCn.
(5) If bit 7 (TCEn) of TMCn is set to 1, counting starts. When counting stops, set TCEn to 0.

PWM output operation

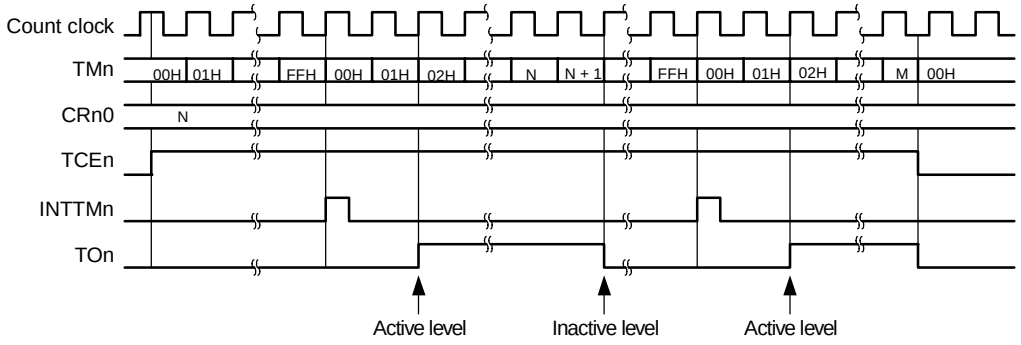
(1) When counting starts, the PWM output (output from TOn) outputs the inactive level until an overflow occurs.
(2) When the overflow occurs, the active level specified in step (1) in the setting method is output. The active level is output until CRn0 and the count of the 8-bit counter n (TMn) match.
(3) The PWM output after CRn0 and the count match is the inactive level until an overflow occurs again.
(4) Steps (2) and (3) repeat until counting stops.
(5) If counting is stopped by TCEn = 0, the PWM output goes to the inactive level.

Remark n = 2 to 7, 10, 11

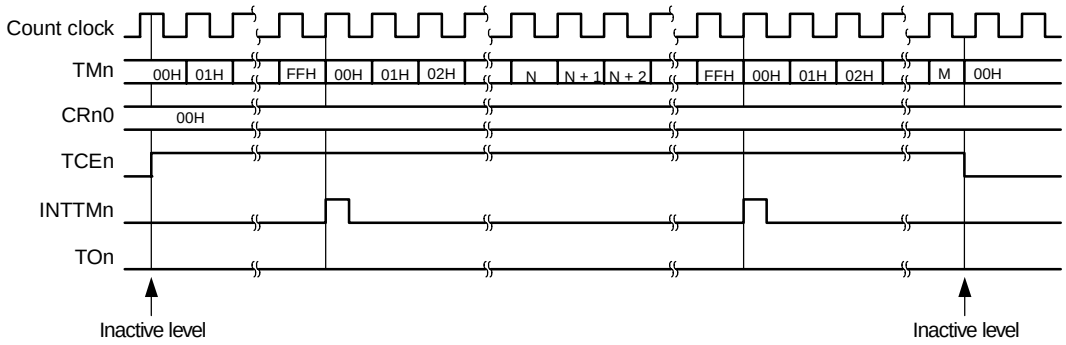
(a) Basic operation of PWM output

Figure 7-94. Timing of PWM Output

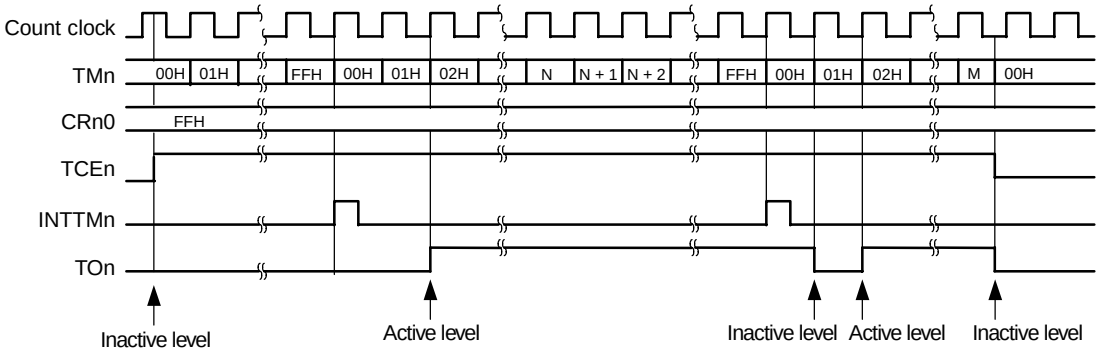
Basic operation (active level = H)



When CRn0 = 0



When CRn0 = FFH

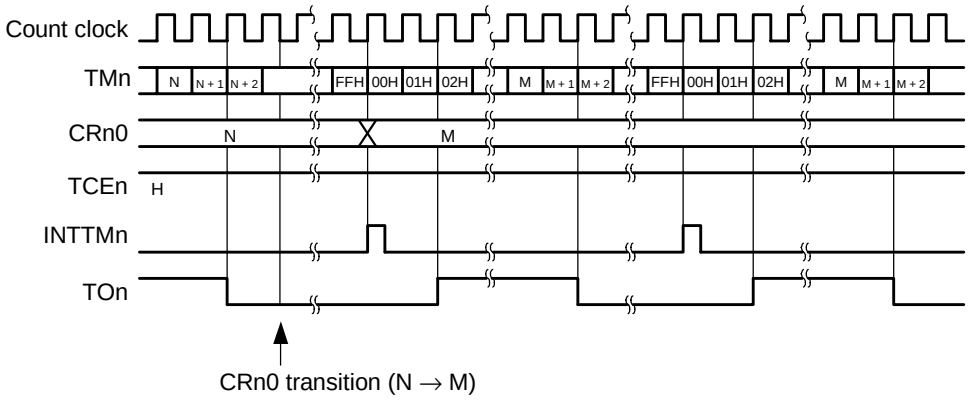


Remark n = 2 to 7, 10, 11

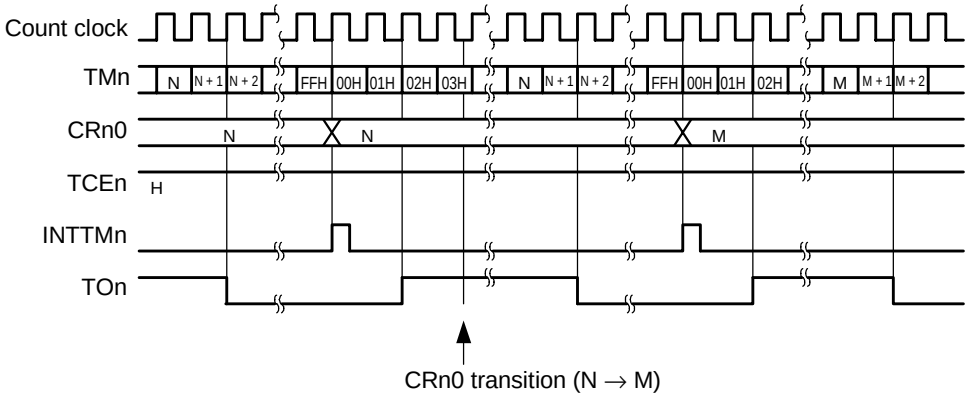
(b) Operation based on CRn0 transitions

Figure 7-95. Timing of Operation Based on CRn0 Transitions

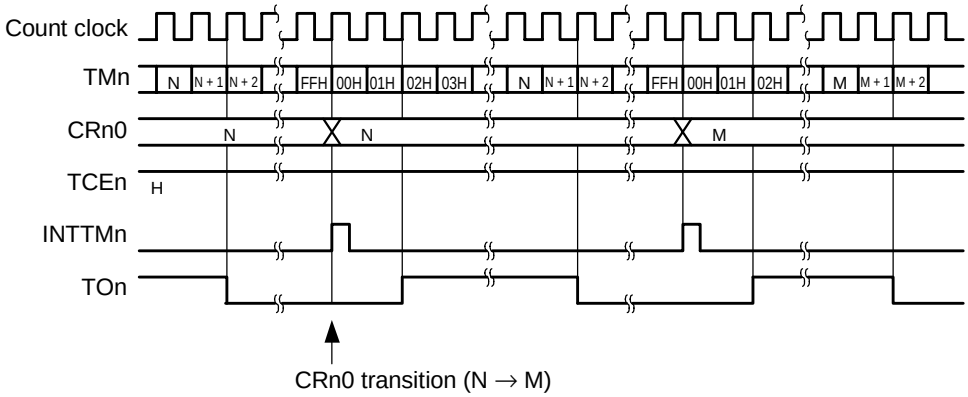
When the CRn0 value changes from N to M before TMn overflows



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



When the CRn0 value changes from N to M during two clocks (00H, 01H) immediately after TMn overflows



Remark n = 2 to 7, 10, 11

7.9.5 Operation as interval timer (16 bits)

(1) Cascade connection (16-bit timer) mode

The V850/SV1 provides a 16-bit register that can be used when connecting in cascade.
Available registers are as follows.

TM2 to TM3 cascade connection:	16-bit counter TM23 (Address: FFFFF24AH)
	16-bit compare register CR23 (Address: FFFFF24CH)
TM4 to TM5 cascade connection:	16-bit counter TM45 (Address: FFFFF26AH)
	16-bit compare register CR45 (Address: FFFFF26CH)
TM6 to TM7 cascade connection:	16-bit counter TM67 (Address: FFFFF28AH)
	16-bit compare register CR67 (Address: FFFFF28CH)
TM10 to TM11 cascade connection:	16-bit counter TM1011 (Address: FFFFF3EAH)
	16-bit compare register CR1011 (Address: FFFFF3ECH)

By setting bit 4 (TMCn4) of the 8-bit timer mode control register n (TMCn) to 1, the timer enters the timer/counter mode with 16-bit resolution (n = 2 to 7, 10, 11).
With the count preset in the 8-bit compare register n (CRn0) as the interval, the timer operates as an interval timer by repeatedly generating interrupts (n = 2 to 7, 10, 11).

The following explains the method of TM2 to TM3 cascade connection. For TM4 to TM5, TM6 to TM7, or TM10 to TM11 cascade connection, read TM2 and TM3 as referring to the target timers.

Example of setting method (TM2 to TM3 cascade connection)

(1) Set each register.
- TCL20, TCL21: TM2 selects the count clock.
  TM3 connected in cascade is not used in setting.
- CR20, CR30: Compare values (Each compare value can be set from 00H to FFH.)
- TMC2: Select the clear and start mode when TM2 and CR20 match. (x: don't care)

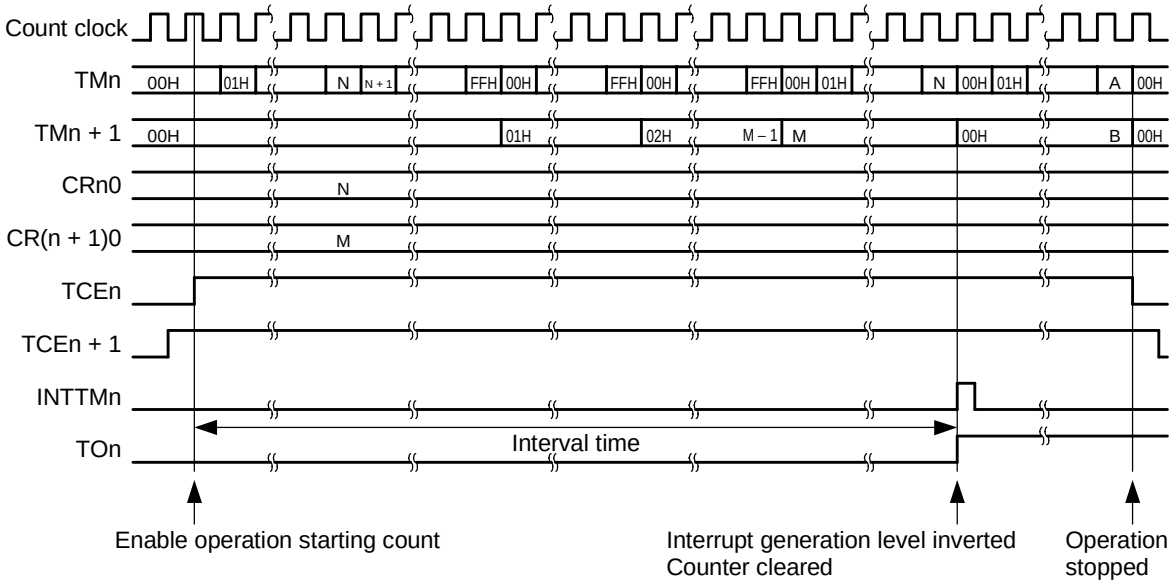
$$\begin{pmatrix} \text{TM2} \rightarrow \text{TMC2} = 0000\text{xxx}0\text{B} \\ \text{TM3} \rightarrow \text{TMC3} = 0001\text{xxx}0\text{B} \end{pmatrix}$$

(2) Setting TCE3 = 1 in TMC3 and then setting TCE2 = 1 in TMC2 starts the count operation.
(3) If the values of TM23 of the timers connected in cascade and CR23 match, INTTM2 in TM2 is generated (TM2 and TM3 are cleared to 00H).
(4) INTTM2 is repeatedly generated at the same interval.

Cautions 1. Before changing the setting of the compare register (CR23) when using 8-bit timers (TM2 and TM3), connected in cascade as a 16-bit timer (TM23), stop the count operation of each 8-bit timer connected in cascade. If the CR23 value is changed without stopping the timers, the values of the eight high-order bits (TM3) will be undefined.
2. If the count of the high-order timer (TM3) matches CR30 even when used in a cascade connection, the interrupt request (INTTM3) of the high-order timer (TM3) is generated. Always mask TM3 in order to disable interrupts.
3. Set the TCE3 bit of TMC3 first, then set the TCE2 bit of TMC2.
4. Restarting and stopping the count is possible by setting only 1 or 0 in TCE2 of TM2 to start and stop it.

Figure 7-96 shows a timing example of the cascade connection mode with 16-bit resolution.

Figure 7-96. Cascade Connection Mode with 16-Bit Resolution



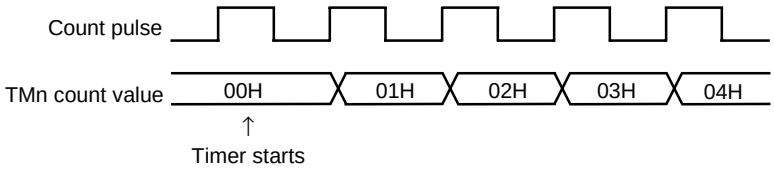
Remark n = 2, 4, 6, 10

7.9.6 Cautions

(1) Error when the timer starts

The time until the coincidence signal is generated after the timer starts has a maximum error of one clock. The reason is the starting of the 8-bit counter n (TMn) is asynchronous with respect to the count pulse.

Figure 7-97. Start Timing of Timer n

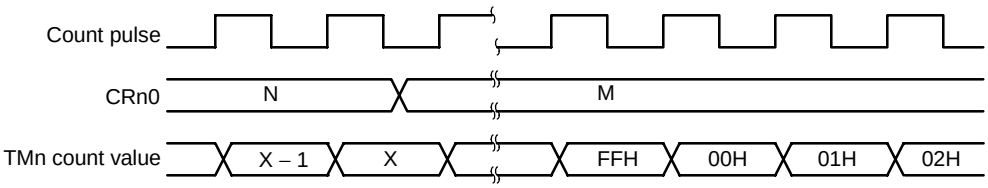


Remark n = 2 to 7, 10, 11

(2) Operation after the compare register is changed while the timer is counting

If the value after the 8-bit compare register n (CRn0) changes is less than the value of the 8-bit timer register (TMn), counting continues, overflows, and counting starts again from 0. Consequently, when the value (M) after CRn0 changes is less than the value (N) before the change, the timer must restart after CRn0 changes (n = 2 to 7, 10, 11).

Figure 7-98. Timing after Compare Register Changes during Timer Count Operation



Remarks 1. N > X > M

2. n = 2 to 7, 10, 11

Caution Except when the TIn input is selected, always set TCEn = 0 before setting the stop state.

(3) TMn read out during timer operation

Since reading out TMn during operation occurs while the selected clock is temporarily stopped, select some high or low level waveform that is longer than the selected clock (n = 2 to 7, 10, 11).

CHAPTER 8 WATCH TIMER

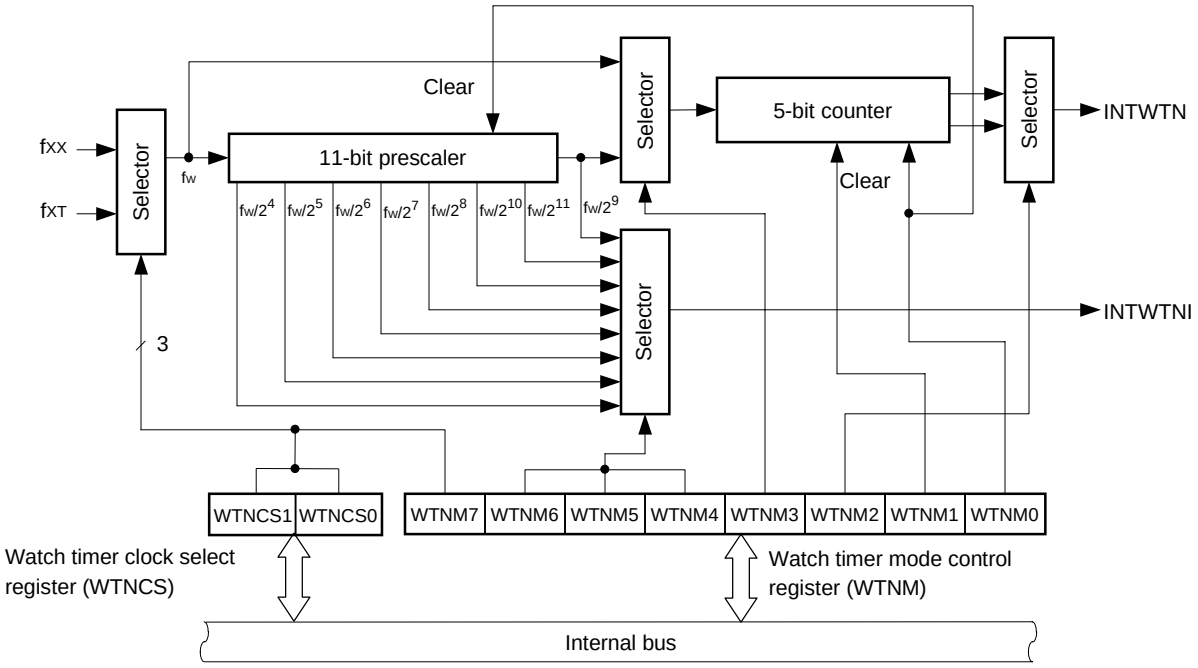
8.1 Function

The watch timer has the following functions:

- Watch timer
- Interval timer

The watch timer and interval timer functions can be used at the same time.
Figure 8-1 shows the block diagram of the watch timer.

Figure 8-1. Block Diagram of Watch Timer



Remarks 1. f_{XX}: Main system clock frequency
2. f_{XT}: Subsystem clock frequency
3. f_W: Watch timer clock frequency

(1) Watch timer

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

(2) Interval timer

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

Table 8-1. Interval Time of Interval Timer

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

Remark f_W: Watch timer clock frequency

8.2 Configuration

The watch timer consists of the following hardware:

Table 8-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control registers	Watch timer mode control register (WTNM) Watch timer clock select register (WTNCS)

8.3 Watch Timer Control Register

The watch timer mode control register (WTNM) and watch timer clock select register (WTNCS) control the watch timer. Before operating the watch timer, set the corresponding count clock.

(1) Watch timer mode control register (WTNM)

This register enables or disables the count clock and operation of the watch timer, sets the interval time of the prescaler, controls the operation of the 5-bit counter, and sets the set time of the watch flag.

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

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

Figure 8-2. Watch Timer Mode Control Register (WTNM)

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	$2^4/f_W$ (488 μs)
0	0	1	$2^5/f_W$ (977 μs)
0	1	0	$2^6/f_W$ (1.95 ms)
0	1	1	$2^7/f_W$ (3.91 ms)
1	0	0	$2^8/f_W$ (7.81 ms)
1	0	1	$2^9/f_W$ (15.6 ms)
1	1	0	$2^{10}/f_W$ (31.2 ms)
1	1	1	$2^{11}/f_W$ (62.4 ms)

WTNM3	WTNM2	Selects Set Time of Watch Flag
0	0	$2^{14}/f_W$ (0.5 s)
0	1	$2^{13}/f_W$ (0.25 s)
1	0	$2^5/f_W$ (977 μs)
1	1	$2^4/f_W$ (488 μs)

WTNM1	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. f_W: Watch timer clock frequency

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

3. For the settings of WTNM7, refer to **Figure 8-3**.

(2) 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 count clock while the watch timer is operating.

Figure 8-3. Watch Timer Clock Select Register (WTNCS)

After reset: 00H R/W Address: FFFFF364H

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

WTNCS1	WTNCS0	WTNM7	Selection of Count Clock	Main Clock Frequency
0	0	0	$f_{XX}/2^7$	4.194 MHz
0	0	1	f_{XT} (sub clock)	–
0	1	0	$f_{XX}/3 \times 2^6$	6.291 MHz
0	1	1	$f_{XX}/2^8$	8.388 MHz
1	0	0	Setting prohibited	–
1	0	1	Setting prohibited	–
1	1	0	$f_{XX}/3 \times 2^7$	12.582 MHz
1	1	1	$f_{XX}/2^9$	16.776 MHz

Remark WTNM7 is bit 7 of the WTNM register

8.4 Operation

8.4.1 Operation as watch timer

The watch timer operates with time intervals of 0.5 seconds with the subsystem clock (32.768 kHz).

The watch timer generates an interrupt request at fixed time intervals.

The count operation of the watch timer is started when bits 0 (WTNM0) and 1 (WTNM1) of the watch timer mode control register (WTNM) are set to 1. When these bits are cleared to 0, the 11-bit prescaler and 5-bit counter are cleared, and the watch timer stops the count operation.

The watch timer can be cleared by setting WTNM1 to 0. This, however, may incur an error of up to 15.6 ms.

The interval timer can be cleared by setting WTNM0 to 0. Since the 5-bit counter is cleared at the same time, an error of up to 0.5 seconds may arise if the watch timer overflows (INTWTN).

8.4.2 Operation as interval timer

The watch timer can also be used as an interval timer that repeatedly generates an interrupt at intervals specified by a count value set in advance.

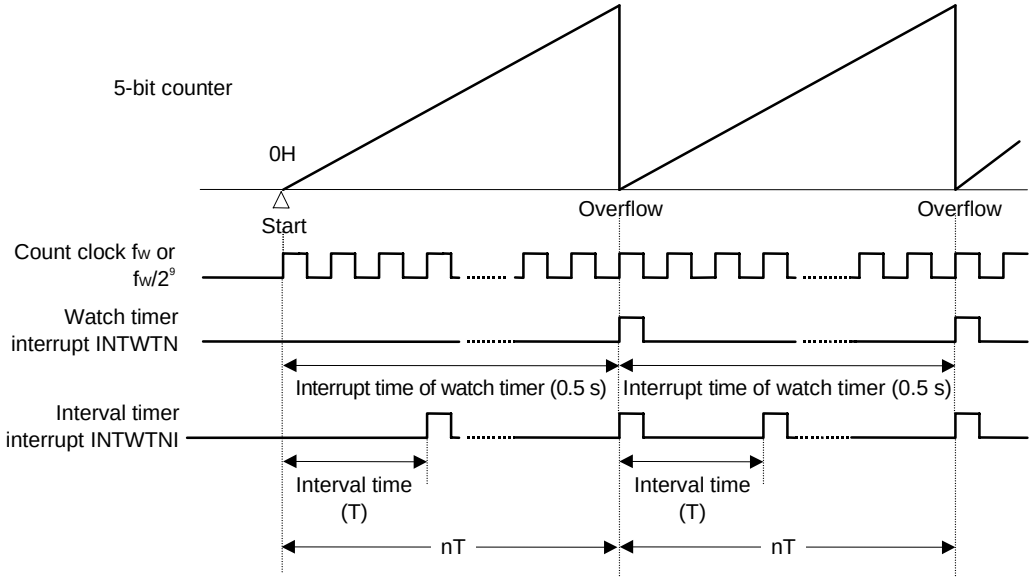
The interval time can be selected by bits 4 through 6 (WTNM4 through WTNM6) of the watch timer mode control register (WTNM).

Table 8-3. Interval Time of Interval Timer

WTNM6	WTNM5	WTNM4	Interval Time	f_W = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	488 μs
0	0	1	$2^5 \times 1/f_W$	977 μs
0	1	0	$2^6 \times 1/f_W$	1.95 ms
0	1	1	$2^7 \times 1/f_W$	3.91 ms
1	0	0	$2^8 \times 1/f_W$	7.81 ms
1	0	1	$2^9 \times 1/f_W$	15.6 ms
1	1	0	$2^{10} \times 1/f_W$	31.2 ms
1	1	1	$2^{11} \times 1/f_W$	62.4 ms

Remark f_W : Watch timer clock frequency

Figure 8-4. Operation Timing of Watch Timer/Interval Timer



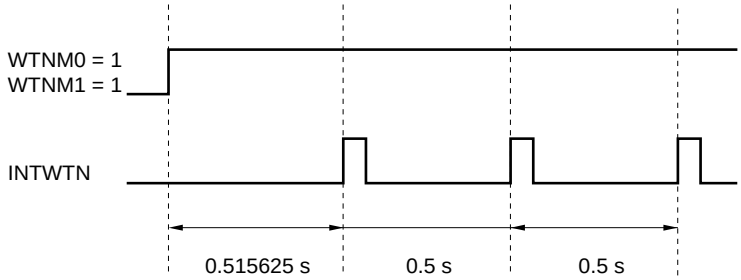
Remark fw: Watch timer clock frequency
( ): fw = 32.768 kHz
n: Interval timer operation numbers

8.4.3 Cautions

After the operation is enabled (the WTNM1 and WTNM0 bits of the watch timer mode control register (WTNM) are both set to 1), it will take a little time before the first watch timer interrupt request (INTWTN) is issued.

Example) If the interrupt cycle is set to 0.5 seconds, the first watch timer interrupt request (INTWTN) will occur 0.515625 seconds later. (The time is longer than the cycle by $29 \times 1/32.768 = 0.015625$ seconds.) Then, the watch timer interrupt request (INTWTN) occurs at 0.5-second intervals.

Figure 8-5. Sample Interrupt Request (0.5-Second Intervals)



CHAPTER 9 WATCHDOG TIMER

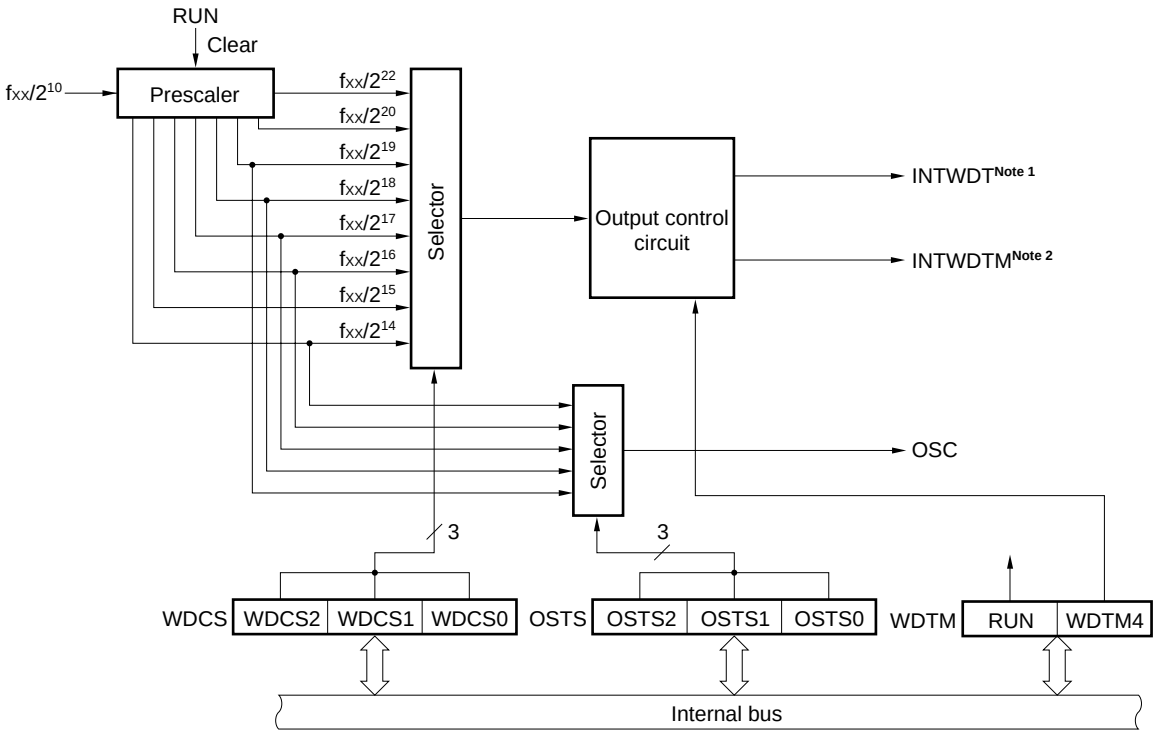
9.1 Functions

The watchdog timer has the following functions. Figure 9-1 shows a block diagram of the watchdog timer.

- Watchdog timer
- Interval timer
- Selecting the oscillation stabilization time

Caution Use the watchdog timer mode register (WDTM) to select the watchdog timer mode or the interval timer mode.

Figure 9-1. Block Diagram of Watchdog Timer



Notes 1. When watchdog timer mode

2. When interval timer mode

Remark f_{XX}: Main system clock frequency

(1) Watchdog timer mode

This mode detects program runaway. When runaway is detected, a non-maskable interrupt can be generated.

Table 9-1. Runaway Detection Time by Watchdog Timer

Clock	Runaway Detection Time		
	f_{XX} = 20 MHz	f_{XX} = 10 MHz	f_{XX} = 2 MHz
$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
$2^{15}/f_{XX}$	1.6 ms	3.2 ms	16.4 ms
$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
$2^{19}/f_{XX}$	26.2 ms	52.4 ms	262.1 ms
$2^{20}/f_{XX}$	52.4 ms	104.9 ms	524.3 ms
$2^{22}/f_{XX}$	209.7 ms	419.4 ms	2.1 s

(2) Interval timer mode

Interrupts are generated at a preset time interval.

Table 9-2. Interval Time of Interval Timer

Clock	Runaway Detection Time		
	f_{XX} = 20 MHz	f_{XX} = 10 MHz	f_{XX} = 2 MHz
$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
$2^{15}/f_{XX}$	1.6 ms	3.2 ms	16.4 ms
$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
$2^{19}/f_{XX}$	26.2 ms	52.4 ms	262.1 ms
$2^{20}/f_{XX}$	52.4 ms	104.9 ms	524.3 ms
$2^{22}/f_{XX}$	209.7 ms	419.4 ms	2.1 s

9.2 Configuration

The watchdog timer consists of the following hardware.

Table 9-3. Watchdog Timer Configuration

Item	Configuration
Control registers	Oscillation stabilization time select register (OSTS) Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM)

9.3 Watchdog Timer Control Register

Three registers control the watchdog timer.

- Oscillation stabilization time select register (OSTS)
- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Oscillation stabilization time select register (OSTS)

This register selects the oscillation stabilization time after a reset is applied or the STOP mode is released until the oscillation is stable.

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H.

Figure 9-2. Oscillation Stabilization Time Select Register (OSTS)

After reset: 04H R/W Address: FFFFF380H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation Stabilization Time Selection			
			Clock	f_{XX}		
				20 MHz	10 MHz	2 MHz
0	0	0	$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
0	0	1	$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
0	1	0	$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
0	1	1	$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
1	0	0	$2^{19}/f_{XX}$ (after reset)	26.2 ms	52.4 ms	262.1 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.

Figure 9-3. Watchdog Timer Clock Select Register (WDCS)

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	10 MHz	2 MHz
0	0	0	$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
0	0	1	$2^{15}/f_{XX}$	1.6 ms	3.2 ms	16.4 ms
0	1	0	$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
0	1	1	$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
1	0	0	$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
1	0	1	$2^{19}/f_{XX}$	26.2 ms	52.4 ms	262.1 ms
1	1	0	$2^{20}/f_{XX}$	52.4 ms	104.9 ms	524.3 ms
1	1	1	$2^{22}/f_{XX}$	209.7 ms	419.4 ms	2.1 s

(3) Watchdog timer mode register (WDTM)

This register sets the operating mode of the watchdog timer, and enables and disables counting.

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

$\overline{\text{RESET}}$ input sets WDTM to 00H.

Figure 9-4. Watchdog Timer Mode Register (WDTM)

After reset: 00H R/W Address: FFFFF384H

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

RUN	Operating Mode Selection for the Watchdog Timer[Note 1]
0	Disable count
1	Clear count and start counting

WDTM4	Operating Mode Selection for the Watchdog Timer[Note 2]
0	Interval timer mode (If an overflow occurs, a maskable interrupt INTWDTM is generated.)
1	Watchdog timer mode 1 (If an overflow occurs, a non-maskable interrupt INTWDT is generated.)

Notes 1. If RUN is set (1) once, 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. If WDTM4 is set (1) once, the register cannot be cleared (0) by software.

Caution If RUN is set (1) and the watchdog timer is cleared, the actual overflow time may be a maximum of $2^{10}/f_{XX}$ (seconds) less than the set time.

9.4 Operation

9.4.1 Operation as watchdog timer

Set bit 4 (WDTM4) of the watchdog timer mode register (WDTM) to 1 to operate as a watchdog timer to detect program runaway.

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 runaway detection, the watchdog timer is cleared and counting starts again.

If RUN is not set to 1 and the runaway detection time has elapsed, a non-maskable interrupt (INTWDT) is generated (no reset functions).

The watchdog timer stops running in the STOP mode and IDLE mode. Consequently, set RUN to 1 and clear the watchdog timer before entering the STOP mode or IDLE mode. Do not set the watchdog timer when operating the HALT mode since the watchdog timer running in HALT mode.

Cautions 1. Sometimes, the actual runaway detection time is a maximum of $2^{10}/f_{XX}$ (seconds) less than the set time.

2. When the subclock is selected in a CPU clock, the watchdog timer stops (retains) counting.

Table 9-4. Runaway Detection Time of Watchdog Timer

Clock	Runaway Detection Time		
	f_{XX} = 20 MHz	f_{XX} = 10 MHz	f_{XX} = 2 MHz
$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
$2^{15}/f_{XX}$	1.6 ms	3.2 ms	16.4 ms
$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
$2^{19}/f_{XX}$	26.2 ms	52.4 ms	262.1 ms
$2^{20}/f_{XX}$	52.4 ms	104.9 ms	524.3 ms
$2^{22}/f_{XX}$	209.7 ms	419.4 ms	2.1 s

9.4.2 Operation as interval timer

Set bit 4 (WDTM4) to 0 in the watchdog timer mode register (WDTM) to operate the watchdog timer as an interval timer that repeatedly generates interrupts with a preset count value as the interval.

When operating as an interval timer, the interrupt mask flag (WDTMK) of the WDTIC register and the priority setting flag (WDTPR0 to WDTPR2) become valid, and a maskable interrupt (INTWDTM) can be generated. The default priority of INTWDTM has the highest priority setting of the maskable interrupts.

The interval timer continues operating in the HALT mode and stops in the STOP mode and IDLE mode. Therefore, after the RUN bit of WDTM register is set to 1 and the interval timer is cleared before entering the STOP mode/IDLE mode, execute the STOP instruction.

Cautions 1. If bit 4 (WDTM4) of WDTM is set to 1 once (selecting the watchdog timer mode), the interval timer mode is not entered as long as $\overline{\text{RESET}}$ is not input.

2. Sometimes, the interval time immediately after setting in WDTM is a maximum of $2^{10}/f_{XX}$ (seconds) less than the set time.

3. When the subclock is selected in the CPU clock, the watchdog timer stops (retains) counting.

Table 9-5. Interval Time of Interval Timer

Clock	Runaway Detection Time		
	f_{XX} = 20 MHz	f_{XX} = 10 MHz	f_{XX} = 2 MHz
$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
$2^{15}/f_{XX}$	1.6 ms	3.2 ms	16.4 ms
$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
$2^{19}/f_{XX}$	26.2 ms	52.4 ms	262.1 ms
$2^{20}/f_{XX}$	52.4 ms	104.9 ms	524.3 ms
$2^{22}/f_{XX}$	209.7 ms	419.4 ms	2.1 s

9.5 Standby Function Control Register

The wait time from releasing the STOP mode until the oscillation stabilizes is controlled by the oscillation stabilization time select register (OSTS).

OSTS is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 04H.

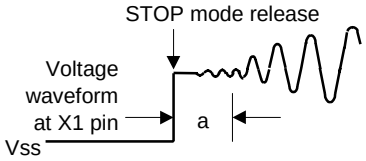
Figure 9-5. Oscillation Stabilization Time Select Register (OSTS)

After reset: 04H R/W Address: FFFFF380H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation Stabilization Time Selection			
			Clock	fxx		
				20 MHz	10 MHz	2 MHz
0	0	0	$2^{14}/f_{XX}$	819.2 μs	1.6 ms	8.2 ms
0	0	1	$2^{16}/f_{XX}$	3.3 ms	6.6 ms	32.8 ms
0	1	0	$2^{17}/f_{XX}$	6.6 ms	13.1 ms	65.5 ms
0	1	1	$2^{18}/f_{XX}$	13.1 ms	26.2 ms	131.1 ms
1	0	0	$2^{19}/f_{XX}$ (after reset)	26.2 ms	52.4 ms	262.1 ms
Other than above			Setting prohibited			

Caution The wait time at the release of the STOP mode does not include the time (a in the figure below) until clock oscillation starts after releasing the STOP mode when $\overline{\text{RESET}}$ is input or an interrupt is generated.



CHAPTER 10 SERIAL INTERFACE FUNCTION

10.1 Outline

The V850/SV1 supports the following on-chip serial interfaces.

- Channel 0: 3-wire serial I/O (CSI0)/I^2C0[Note]
- Channel 1: 3-wire serial I/O (CSI1)/Asynchronous serial interface (UART0)
- 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: 8- to 16-bit variable 3-wire serial I/O (CSI4)

Note I^2C0 and I^2C1 support Multi-master (μPD703039Y, 703040Y, 703041Y, and 70F3040Y only). Either 3-wire serial I/O or I^2C can be used as a serial interface.

10.2 3-Wire Serial I/O (CSI0 to CSI3)

The CSIn (n = 0 to 3) has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed.

(2) 3-wire serial I/O mode (fixed as MSB first)

This is an 8-bit data transfer mode using three lines: a serial clock line ($\overline{\text{SCKn}}$), serial output line (SOn), and serial input line (SIn).

Since simultaneous transmit and receive operations are enabled in 3-wire serial I/O mode, the processing time for data transfer is reduced.

The first bit in the 8-bit data in serial transfers is fixed as the MSB.

3-wire serial I/O mode is useful for connection to a peripheral I/O device that includes a clock-synchronous serial interface, a display controller, etc.

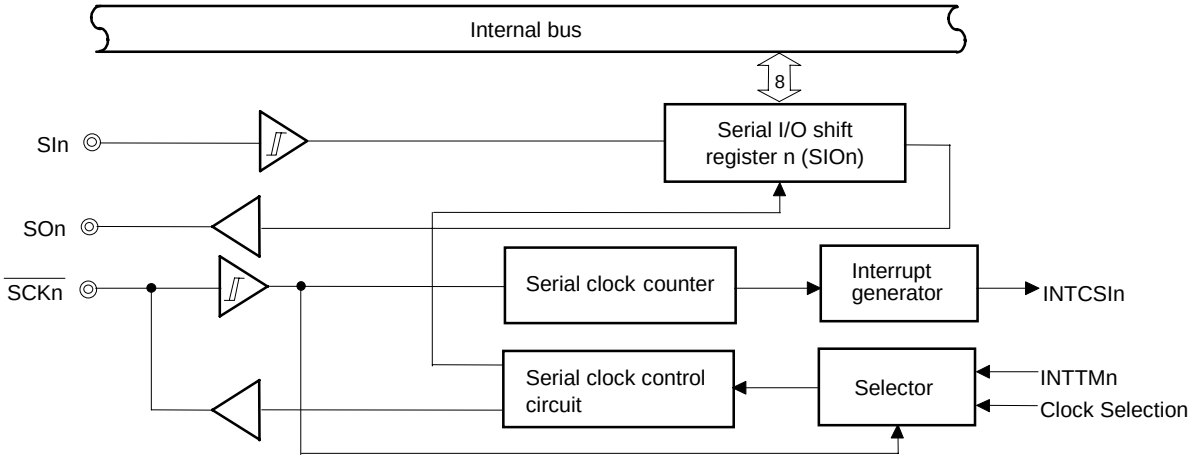
10.2.1 Configuration

The CSIn includes the following hardware.

Table 10-1. Configuration of CSIn

Item	Configuration
Registers	Serial I/O shift registers 0 to 3 (SIO0 to SIO3)
Control registers	Serial operation mode registers 0 to 3 (CSIM0 to CSIM3)
	Serial clock select registers 0 to 3 (CSIS0 to CSIS3)

Figure 10-1. Block Diagram of 3-Wire Serial I/O



Remark n = 0 to 3

(1) Serial I/O shift registers 0 to 3 (SIO0 to SIO3)

SIOn is an 8-bit register that performs parallel-serial conversion and serial transmit/receive (shift operations) synchronized with the serial clock.

SIOn is set by an 8-bit memory manipulation instruction.

When “1” is set to bit 7 (CSIEn) of the serial operation mode register n (CSIMn), a serial operation can be started by writing data to or reading data from SIOn.

When transmitting, data written to SIOn is output via the serial output (SOn).

When receiving, data is read from the serial input (SIn) and written to SIOn.

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

Caution Do not execute SIOn accesses except for the accesses that become transfer start trigger during transfer operation (read is disabled when MODEn = 0 and write is disabled when MODEn = 1).

10.2.2 CSIn control registers

The CSIn uses the following type of register for control functions.

- Serial operation mode register n (CSIMn)
- Serial clock select register n (CSISn)

(1) Serial operation mode registers 0 to 3 (CSIM0 to CSIM3)

CSIMn is used to enable or disable serial interface channel n's serial clock, operation modes, and specific operations.

CSIMn can be set by a 1-bit or 8-bit memory manipulation instruction.

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

(2) Serial clock select registers 0 to 3 (CSIS0 to CSIS3)

CSISn is used to set serial interface channel n's serial clock.

CSISn can be set by an 8-bit memory manipulation instruction.

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

Figure 10-2. Serial Operation Mode Registers 0 to 3 (CSIM0 to CSIM3) (1/2)

After reset: 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

CSIEn	SIOn Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disable	Clear	Port function[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

Notes 1. When CSIEn = 0 (SIOn operation disable status), the port function is available for the SIn, SOn, and $\overline{\text{SCKn}}$ pins.

2. When CSIEn = 1 (SIOn operation enable status), the port function is available for the SIn pin when only the transmit function is used and for the SOn pin when only the receive function is used.

Figure 10-2. Serial Operation Mode Registers 0 to 3 (CSIM0 to CSIM3) (2/2)

After reset: 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

SCLn2	SCLn1	SCLn0	Clock Selection
0	0	0	External clock input ($\overline{SCKn}$)
0	0	1	When n = 0, 3: TO2 When n = 1, 2: TO3
0	1	0	$f_{XX}/8$
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	$f_{XX}/64$

Caution Do not perform bit manipulation of the SCLn1 and SCLn0.

Remark Refer to **Figure 10-3** for the SCLn2 bit.

Figure 10-3. Serial Clock Select Registers 0 to 3 (CSIS0 to CSIS3)

After reset : 00H R/W Address: CSIS0 FFFFF2A4H
CSIS1 FFFFF2B4H
CSIS2 FFFFF2C4H
CSIS3 FFFFF2D4H

	7	6	5	4	3	2	1	0
CSISn	0	0	0	0	0	0	0	SCLn2

(n = 0 to 3)

Remark Refer to **Figure 10-2** for the setting of the SCLn2 bit.

10.2.3 Operations

The CSIn has the following two operation modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode does not perform serial transfers and can therefore reduce power consumption.

When in operation stop mode, if SIn, SOn, and the serial clock pin are also used as I/O ports, they can be used as normal I/O ports as well.

(a) Register settings

Operation stop mode are set via serial operation mode register n (CSIMn).

CSIMn can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the register to 00H.

Figure 10-4. Serial Operation Mode Registers 0 to 3 (CSIM0 to CSIM3)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

CSIEn	SIOn Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disable	Clear	Port function
1	Operation enable	Count operation enable	Serial function + port function

(2) 3-wire serial I/O mode

3-wire serial I/O mode is useful when connecting to a peripheral I/O device that includes a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCKn}}$), serial output line (SOn), and serial input line (SIn).

(a) Register settings

3-wire serial I/O mode is set via serial operation mode register n (CSIMn).

CSIMn can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the register to 00H .

Figure 10-5. Serial Operation Mode Registers 0 to 3 (CSIM0 to CSIM3) (1/2)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

CSIEn	SIOn Operation Enable/Disable Specification		
	Shift Register Operation	Serial Counter	Port
0	Operation disable	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	Serial function + port function[Note 2]

Notes 1. When CSIEn = 0 (SIOn operation disable status), the port function is available for the SIn, SOn, and $\overline{\text{SCKn}}$ pins.

2. When CSIEn = 1 (SIOn operation enable status), the port function is available for the SIn pin when only the transmit function is used and for the SOn pin when only the receive function is used.

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

Figure 10-5. Serial Operation Mode Registers 0 to 3 (CSIM0 to CSIM3) (2/2)

After reset : 00H R/W Address: CSIM0 FFFFF2A2H
CSIM1 FFFFF2B2H
CSIM2 FFFFF2C2H
CSIM3 FFFFF2D2H

	7	6	5	4	3	2	1	0
CSIMn	CSIEn	0	0	0	0	MODEn	SCLn1	SCLn0

(n = 0 to 3)

SCLn2	SCLn1	SCLn0	Clock Selection
0	0	0	External clock input ($\overline{SCKn}$)
0	0	1	When n = 0, 3: TO2 When n = 1, 2: TO3
0	1	0	fxx/8
0	1	1	fxx/16
1	0	0	Setting prohibited
1	0	1	Setting prohibited
1	1	0	fxx/32
1	1	1	fxx/64

Remark Refer to **Figure 10-3** for the SCLn2 bit.

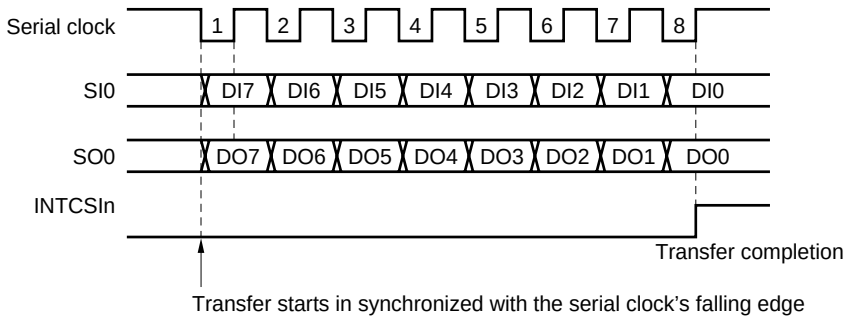
(b) Communication operations

In 3-wire serial I/O mode, data is transmitted and received in 8-bit units. Each bit of data is sent or received in synchronized with the serial clock.

The serial I/O shift register n (SIOn) is shifted in synchronized with the falling edge of the serial clock. Transmission data is held in the SOn latch and is output from the SOn pin. Data that is received via the SIn pin in synchronized with the rising edge of the serial clock is latched to SIOn.

Completion of an 8-bit transfer automatically stops operation of SIOn and sets the interrupt request flag (INTCSIn).

Figure 10-6. 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.

Data is transferred to SIOn as follows:

- Transmit/receive mode
  When CSIEn = 1 and MODEn = 0, transfer starts when writing to SIOn.
- Receive-only mode
  When CSIEn = 1 and MODEn = 1, transfer starts when reading from SIOn.

Caution After data has been written to SIOn, transfer will not start even if the CSIEn bit value is set to “1”.

Completion of an 8-bit transfer automatically stops the serial transfer operation and sets the interrupt request flag (INTCSIn).

10.3 I^2C Bus (μPD703039Y, 703040Y, 703041Y, and 70F3040Y)

To use the I^2C bus function, set the P10/SDA0, P12/SCL0, P20/SDA1, and P22/SCL1 pins to N-ch open drain output.

The I^2C0 and I^2C1 have the following two modes.

- Operation stop mode
- I^2C (Inter IC) bus mode (multi-master supported)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multi-master support)

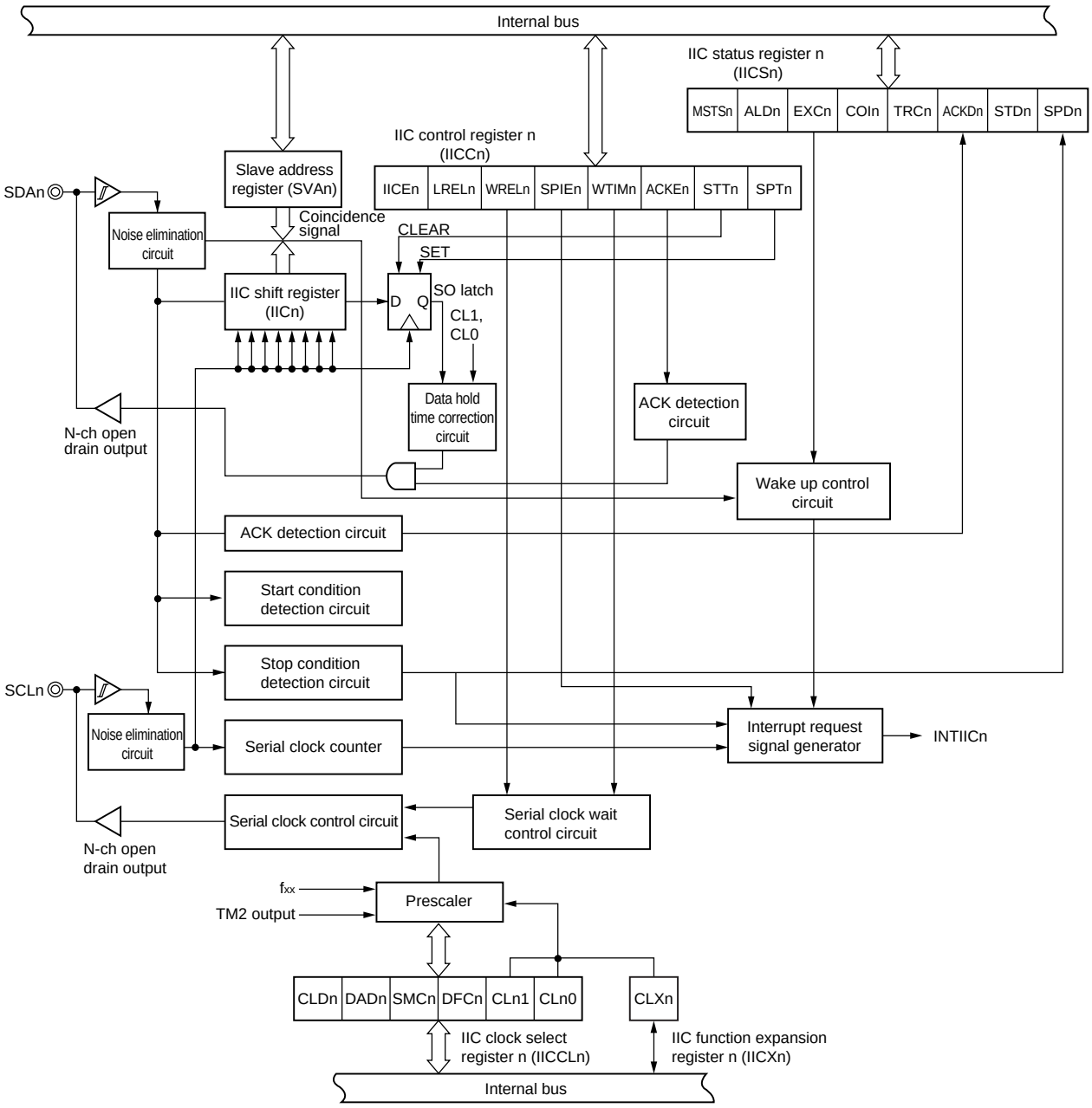
This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLn) line and a serial data bus (SDAn) line.

This mode complies with the I^2C bus format and the master device can output “start condition”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since SCLn and SDAn are open drain outputs, the I^2Cn requires pull-up resistors for the serial clock line and the serial data bus line.

Remark n = 0, 1

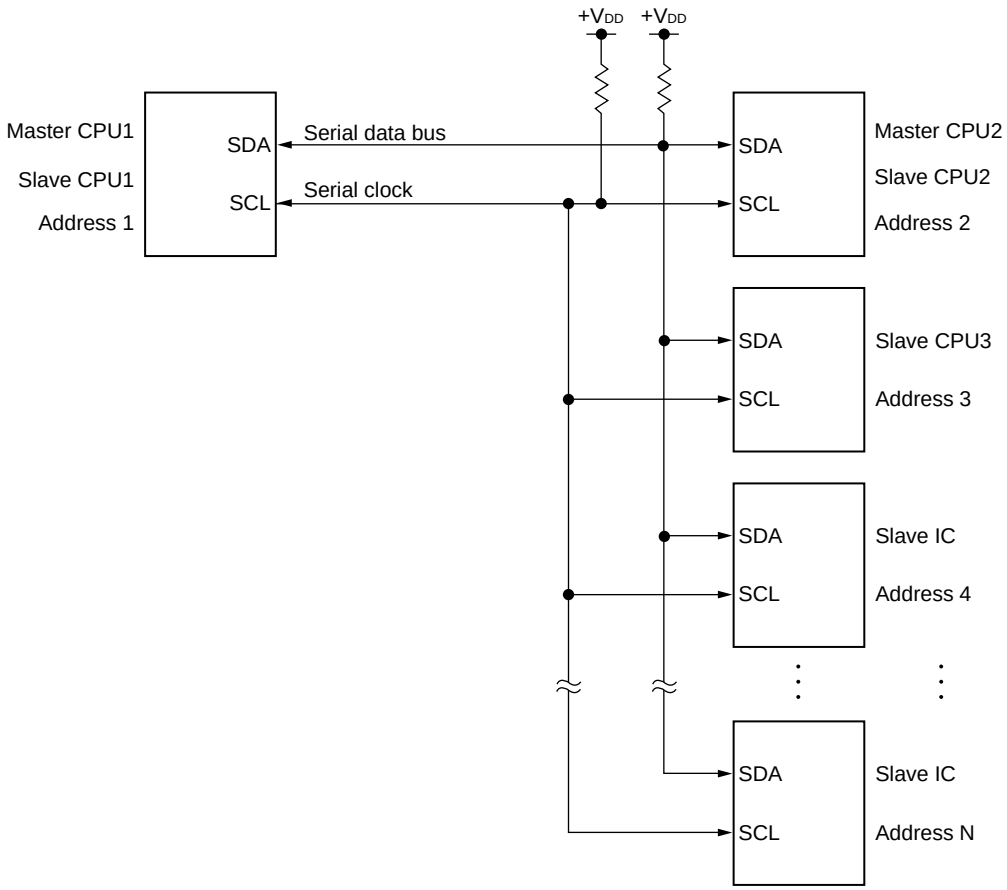
Figure 10-7. Block Diagram of I^2C



Remark n = 0, 1

Figure 10-8 shows a serial bus configuration example.

Figure 10-8. Serial Bus Configuration Example Using I^2C Bus



10.3.1 Configuration

The I^2Cn includes the following hardware (n = 0, 1).

Table 10-2. Configuration of I^2Cn

Item	Configuration
Registers	IIC shift registers 0 and 1 (IIC0, IIC1) Slave address registers 0 and 1 (SVA0, SVA1)
Control registers	IIC control registers 0 and 1 (IICC0, IICC1) IIC status registers 0 and 1 (IICS0, IICS1) IIC clock select registers 0 and 1 (IICCL0, IICCL1) IIC function expansion registers 0 and 1 (IICX0, IICX1)

(1) IIC shift registers 0 and 1 (IIC0, IIC1)

IICn is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data.
IICn can be used for both transmission and reception (n = 0, 1).
Write and read operations to IICn are used to control the actual transmit and receive operations.
IICn is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the IIC0 and IIC1 to 00H.

(2) Slave address registers 0 and 1 (SVA0, SVA1)

SVAn sets local addresses when in slave mode.
SVAn is set by an 8-bit memory manipulation instruction (n = 0, 1).
$\overline{\text{RESET}}$ input sets the SVA0 and SVA1 to 00H.

(3) SO latch

The SO latch is used to retain the SDAn pin's output level (n = 0, 1).

(4) Wake-up control circuit

This circuit generates an interrupt request when the address received by this register matches the address value set to the slave address register (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.
An I^2C interrupt is generated following either of two triggers.

- Eighth or ninth clock of the serial clock (set by WTIMn bit[Note])
- Interrupt request generated when a stop condition is detected (set by SPIEn bit[Note])

Note WTIMn bit: Bit 3 of the IIC control register (IICCn)
SPIEn bit: Bit 4 of the IIC control register (IICCn)

Remark n = 0, 1

(8) Serial clock control circuit

In master mode, this circuit generates the clock output via the SCLn pin from a sampling clock (n = 0, 1).

(9) Serial clock wait control circuit

This circuit controls the wait timing.

(10) ACK output circuit, stop condition detection circuit, start condition detection circuit, and ACK detection circuit

These circuits are used to output and detect various control signals.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

10.3.2 I^2C control register

I^2C0 and I^2C1 are controlled via four types of registers.

- IIC control registers 0, 1 (IICC0, IICC1)
- IIC status registers 0, 1 (IICS0, IICS1)
- IIC clock select registers 0, 1 (IICCL0, IICCL1)
- IIC function expansion registers 0, 1 (IICX0, IICX1)

The following registers are also used.

- IIC shift registers 0, 1 (IIC0, IIC1)
- Slave address registers 0, 1 (SVA0, SVA1)

(1) IIC control registers 0, 1 (IICC0, IICC1)

IICCn is used to enable/disable I^2C operations, set wait timing, and set other I^2C operations.

IICCn can be set by a 1-bit or 8-bit memory manipulation instruction (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICCn to 00H.

Caution In I^2C0, I^2C1 bus mode, set port 1 mode register (PM1) and port 2 mode register (PM2) as follows. In addition, set each output latch to 0.

- **Set P10 (SDA0) to output mode (PM10 = 0)**
- **Set P12 (SCL0) to output mode (PM12 = 0)**
- **Set P20 (SDA1) to output mode (PM20 = 0)**
- **Set P22 (SCL1) to output mode (PM22 = 0)**

Figure 10-9. IIC Control Register n (IICCn) (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	Stops operation. Presets IIC status register n (IICSn). Stops internal operation.
1	Enables operation.

Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

LRELn	Exit from Communications
0	Normal operation
1	This exits from the current communications operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLn and SDAn lines are set 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 communications entry conditions are met.
- After a stop condition is detected, restart is in master mode.
- An address match or extension code reception occurs after the start condition.

Condition for clearing (LRELn = 0)[Note]	Condition for setting (LRELn = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark STDn: Bit 1 of IIC status register n (IICSn)
ACKDn: Bit 2 of IIC status register n (IICSn)
TRCn: Bit 3 of IIC status register n (IICSn)
COIn: Bit 4 of IIC status register n (IICSn)
EXCn: Bit 5 of IIC status register n (IICSn)
MSTSn: Bit 7 of IIC status register n (IICSn)

Figure 10-9. IIC Control Register n (IICCn) (2/4)

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)

WRELn	Wait Cancellation Control
0	Does not cancel wait
1	Cancels wait. This setting is automatically cleared after wait is canceled.
If WRELn is set (wait is canceled) during the wait period of the ninth clock in the transmit status (TRCn = 1), the SDAn line is placed in the high impedance status (TRCn = 0).	
Condition for clearing (WRELn = 0)[Note]	Condition for setting (WRELn = 1)
• Automatically cleared after execution • When $\overline{\text{RESET}}$ is input	• Set by instruction

SPIEn	Enable/Disable Generation of Interrupt Request When Stop Condition Is Detected
0	Disable
1	Enable
Condition for clearing (SPIEn = 0)[Note]	Condition for setting (SPIEn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

WTIMn	Control of Wait and Interrupt Request Generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode : After output of eight clocks, clock output is set to low level and wait is set. Slave mode : After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode : After output of nine clocks, clock output is set to low level and wait is set. Slave mode : After input of nine clocks, the clock is set to low level and wait is set for master device.
This bit's setting is invalid during an address transfer and is valid as the transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an $\overline{\text{ACK}}$ signal is issued. When the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIMn = 0)[Note]	Condition for setting (WTIMn = 1)
• Cleared by instruction • When $\overline{\text{RESET}}$ is input	• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark TRCn: Bit 3 of IIC status register n (IICSn)

Figure 10-9. IIC Control Register n (IICCn) (3/4)

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)

ACKEn	Acknowledge Control
0	Disable acknowledge.
1	Enable acknowledge. During the ninth clock period, the SDAn 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	Does not generate a start condition.
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: This trigger functions as a start condition reserve flag. When set, it releases the bus and then automatically generates a start condition. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set during transfer. Can be set only when 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. STTn should be set during the wait period.
- Cannot be set at the same time as SPTn

Condition for clearing (STTn = 0)[Note]	Condition for setting (STTn = 1)
• Cleared by instruction • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When LRELn = 1 • Cleared when $\overline{RESET}$ is input	• Set by instruction

Note This flag's signal is invalid when IICEn = 0.

Remark Bit 1 (STTn) is 0 if it is read immediately after data setting.

Figure 10-9. IIC Control Register n (IICCn) (4/4)

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)

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 during the wait period after ACKEn has been set to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the ACKn period. SPTn should be set during the wait period.
- Cannot be set at the same time as STTn.
- SPTn can be set only when in master mode.Note 1
- 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)Note 2	Condition for setting (SPTn = 1)
• Cleared by instruction • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When LRELn = 1 • Cleared when $\overline{\text{RESET}}$ is input	• Set by instruction

Notes 1. Set SPTn only in master mode. However, you must set SPTn and generate a stop condition before the first stop condition is detected following the switch to operation enable status. For details, see **10.3.13 Other cautions**.

2. This flag's signal is invalid when IICEn = 0.

Caution When bit 3 (TRCn) of the IIC status register (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 for high impedance.

Remark Bit 0 (SPTn) is 0 if it is read immediately after data setting.

(2) IIC status registers 0, 1 (IICS0, IICS1)

IICSn indicates the status of the I^2Cn bus.

IICSn can be set by a 1-bit or 8-bit memory manipulation instruction. IICSn is a read-only register (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICSn to 00H.

Figure 10-10. IIC Status Register n (IICSn) (1/3)

After reset: 00H R Address: FFFFF342H, FFFFF352H

	7	6	5	4	3	2	1	0
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

(n = 0, 1)

MSTSn	Master Device Status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When a start condition is generated

ALDn	Detection of Arbitration Loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTSn is cleared.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after IICSn is read[Note] • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the arbitration result is a “loss”.

EXCn	Detection of Extension Code Reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXCn = 0)	Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the high-order four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This register is also cleared when a bit manipulation instruction is executed for bits other than IICSn.

Remark LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

Figure 10-10. IIC Status Register n (IICSn) (2/3)

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)

COIn	Detection of Matching Addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COIn = 0)	Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVAn) (set at the rising edge of the eighth clock).

TRCn	Detection of Transmit/Receive Status
0	Receive status (other than transmit status). The SDAn line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDAn line (valid starting at the falling edge of the first byte's ninth clock).

Condition for clearing (TRCn = 0)	Condition for setting (TRCn = 1)
• When a stop condition is detected • Cleared by LRELn = 1 • When IICEn changes from 1 to 0 • Cleared by WRELn = 1[Note] • When ALDn changes from 0 to 1 • When $\overline{\text{RESET}}$ is input Master • When "1" is output to the first byte's LSB (transfer direction specification bit) Slave • When a start condition is detected • When "0" is input by the first byte's LSB (transfer direction specification bit) 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 This bit is cleared when WRELn is set during the wait period of the ninth clock.

Remark LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)
WRELn: Bit 5 of IIC control register n (IICCn)

Figure 10-10. IIC Status Register n (IICSn) (3/3)

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)

ACKDn	Detection of ACK
0	ACK was not detected.
1	ACK was detected.

Condition for clearing (ACKDn = 0)	Condition for setting (ACKDn = 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

Remark LRELn: Bit 6 of IIC control register n (IICCn)
IICEn: Bit 7 of IIC control register n (IICCn)

(3) IIC clock select registers 0, 1 (IICCL0, IICCL1)

IICCLn is used to set the transfer clock for the I²Cn bus.

IICCLn can be set by a 1-bit or 8-bit memory manipulation instruction (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICCLn to 00H.

Remark The SMCn, DFCn, CLn1, and CLn0 bits are set in combination with the CLXn bit. For details, refer to **10.3.2 (4) (a) Setting transfer clock**.

Figure 10-11. IIC Clock Select Register n (IICCLn)

After reset: 00H R/W[Note] Address: FFFFF344H, FFFFF354H

	7	6	5	4	3	2	1	0
IICCLn	0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0

(n = 0, 1)

CLDn	Detection of SCLn Line Level (Valid Only When IICEn = 1)
0	SCLn line was detected at low level.
1	SCLn line was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLn line is at low level • When IICEn = 0 • When $\overline{\text{RESET}}$ is input	• When the SCLn line is at high level

DADn	Detection of SDAn Line Level (Valid Only When IICEn = 1)
0	SDAn line was detected at low level.
1	SDAn line was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAn line is at low level • When IICEn = 0 • When $\overline{\text{RESET}}$ is input	• When the SDAn line is at high level

Note Bits 4 and 5 are read only bits.

Remark IICEn: Bit 7 of IIC control register n (IICCn)

(4) IIC function expansion registers 0, 1 (IICX0, IICX1)

These registers set the expansion of the I^2C function.

IICXn is set by a 1-bit or 8-bit memory manipulation instruction (n = 0, 1).

$\overline{\text{RESET}}$ input sets IICXn to 00H.

Remark The CLXn bit is set in combination with the CLn0, CLn1, DFCn, and SMCn bits. For details, refer to **10.3.2 (4) (a) Setting transfer clock**.

Figure 10-12. IIC Function Expansion Register n (IICXn)

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)

(a) Setting transfer clock

CLXn	SMCn	DFCn	CLn1	CLn0	Clock Selection	IIC Communication Frequency		
						f_{XX} = 16 MHz	f_{XX} = 8 MHz	f_{XX} = 4 MHz
0	1	×	0	×	f_{XX}/24	Setting prohibited	333 kHz	167 kHz
0	1	×	1	0	f_{XX}/48	333 kHz	167 kHz	83.3 kHz
0	1	×	1	1	(TM2 output)/18 (n = 0) (TM3 output)/18 (n = 1)	–		
0	0	0	0	0	f_{XX}/44	Setting prohibited	Setting prohibited	90.9 kHz
0	0	0	0	1	f_{XX}/86	Setting prohibited	93.0 kHz	46.5 kHz
0	0	0	1	0	f_{XX}/172	93.0 kHz	46.5 kHz	23.3 kHz
0	0	0	1	1	(TM2 output)/66 (n = 0) (TM3 output)/66 (n = 1)	–		
0	0	1	0	0	f_{XX}/46	Setting prohibited	Setting prohibited	87.0 kHz
0	0	1	0	1	f_{XX}/88	Setting prohibited	90.9 kHz	45.5 kHz
0	0	1	1	0	f_{XX}/176	90.9 kHz	45.5 kHz	22.7 kHz
0	0	1	1	1	(TM2 output)/68 (n = 0) (TM3 output)/68 (n = 1)	–		
1	1	×	0	×	f_{XX}/12	Setting prohibited	Setting prohibited	333 kHz
Other than above					Setting prohibited			

Remarks 1. CLXn: Bit 0 of IIC function expansion register n (IICXn)

SMCn: Bit 3 of IIC clock select register n (IICCLn)

DFCn: Bit 2 of IIC clock select register n (IICCLn)

CLn1: Bit 1 of IIC clock select register n (IICCLn)

CLn0: Bit 0 of IIC clock select register n (IICCLn)

2. ×: Don't care

3. n = 0, 1

(5) IIC shift registers 0, 1 (IIC0, IIC1)

IICn is used for serial transmission/reception (shift operations) that are synchronized with the serial clock. It can be read from or written to in 8-bit units, but data should not be written to IICn during a data transfer (n = 0, 1).

Figure 10-13. IIC Shift Register n (IICn)

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 as 0.

Figure 10-14. Slave Address Register n (SVAn)

After reset: 00H R/W Address: FFFFF346H, FFFFF356H

	7	6	5	4	3	2	1	0
SVAn								0

(n = 0, 1)

10.3.3 I^2C bus mode functions

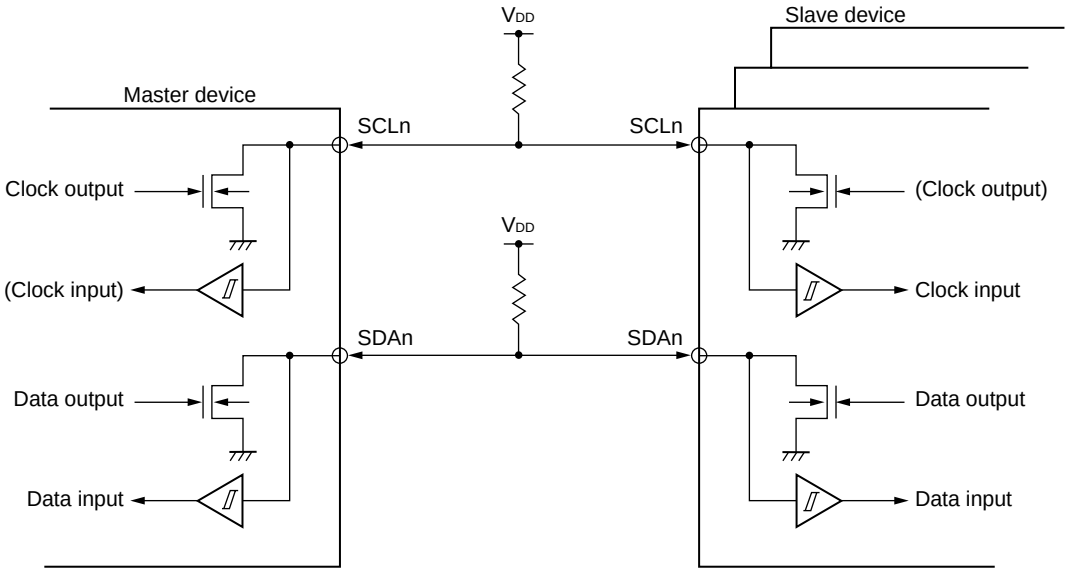
(1) Pin configuration

The serial clock pin (SCLn) and serial data bus pin (SDAn) are configured as follows (n = 0, 1).

SCLn ..... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
SDAn ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

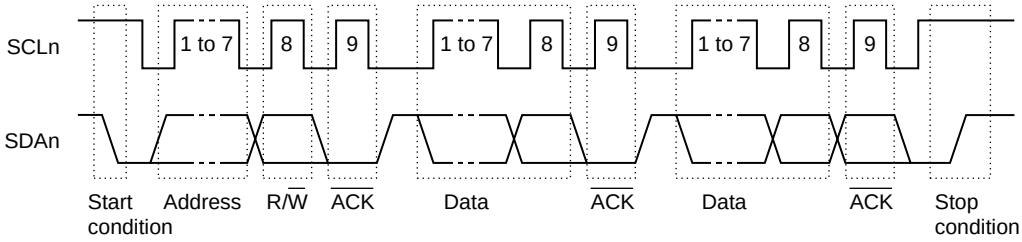
Figure 10-15. Pin Configuration Diagram



10.3.4 I^2C bus definitions and control methods

The following section describes the I^2C bus's serial data communication format and the signals used by the I^2C bus. Figure 10-16 shows the transfer timing for the "start condition", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 10-16. I^2C Bus's Serial Data Transfer Timing



The master device outputs the start condition, slave address, and stop condition.

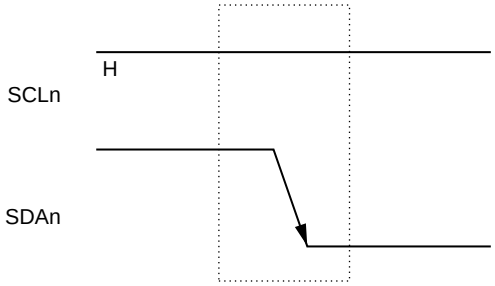
The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLn) is continuously output by the master device. However, in the slave device, the SCLn's low-level period can be extended and a wait can be inserted (n = 0, 1).

(1) Start condition

A start condition is met when the SCLn pin is at high level and the SDAn pin changes from high level to low level. The start conditions for the SCLn pin and SDAn pin are signals that the master device outputs to the slave device when starting a serial transfer. The slave device includes hardware for detecting start conditions (n = 0, 1).

Figure 10-17. Start Conditions



A start condition is output when the IIC control register n (IICCn)'s bit 1 (STTn) is set to 1 after a stop condition has been detected (SPDn: Bit 0 = 1 in the IIC status register n (IICSn)). When a start condition is detected, IICSn's bit 1 (STDn) is set to 1 (n = 0, 1).

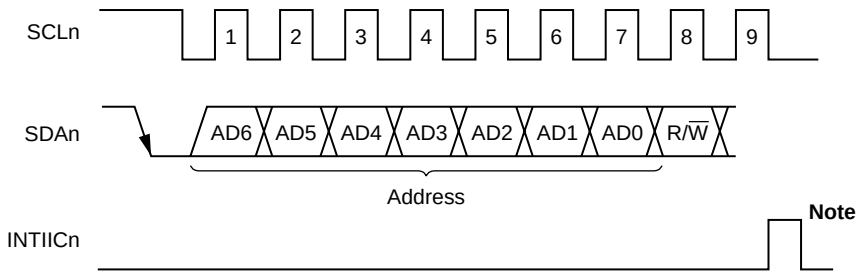
(2) Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register n (SVAn). If the address data matches the SVAn values, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition (n = 0, 1).

Figure 10-18. Address



Note INTIICn is not generated if data other than a local address or extension code is received during slave device operation.

Remark n = 0, 1

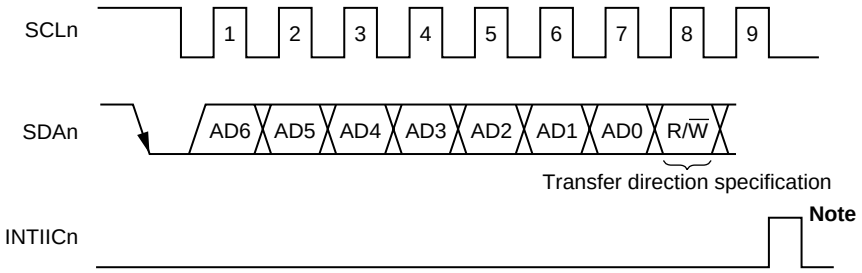
The slave address and the eighth bit, which specifies the transfer direction as described in **(3) Transfer direction specification** below, are together written to the IIC shift register n (IICn) and are then output. Received addresses are written to IICn (n = 0, 1).

The slave address is assigned to the high-order 7 bits of IICn.

(3) Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 10-19. Transfer Direction Specification



Note INTIICn is not generated if data other than a local address or extension code is received during slave device operation.

Remark n = 0, 1

(4) Acknowledge signal ($\overline{\text{ACK}}$)

The acknowledge signal ($\overline{\text{ACK}}$) is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{\text{ACK}}$ signal for each 8 bits of data it receives. The transmitting device normally receives an $\overline{\text{ACK}}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{\text{ACK}}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{\text{ACK}}$ signal is returned after it transmits 8 bits of data. When an $\overline{\text{ACK}}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{\text{ACK}}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{\text{ACK}}$ signal may be caused by the following two factors.

(a) Reception was not performed normally.
(b) The final data was received.

When the receiving device sets the SDAn line to low level during the ninth clock, the $\overline{\text{ACK}}$ signal becomes active (normal receive response).

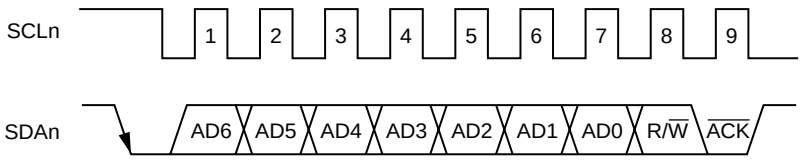
When bit 2 (ACKEn) of the IIC control register n (IICCn) is set to 1, automatic $\overline{\text{ACK}}$ signal generation is enabled (n = 0, 1).

Transmission of the eighth bit following the 7 address data bits causes bit 3 (TRCn) of the IIC status register n (IICSn) to be set. When this TRCn bit's value is 0, it indicates receive mode. Therefore, ACKEn should be set to 1 (n = 0, 1).

When the slave device is receiving (when TRCn = 0), if the slave device does not need to receive any more data after receiving several bytes, setting ACKEn to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRCn = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting ACKEn to 0 will prevent the $\overline{\text{ACK}}$ signal from being returned. This prevents the MSB data from being output via the SDAn line (i.e., stops transmission) during transmission from the slave device.

Figure 10-20. $\overline{\text{ACK}}$ Signal



Remark n = 0, 1

When the local address is received, an $\overline{ACK}$ signal is automatically output in synchronization with the falling edge of the SCLn's eighth clock regardless of the ACKEn value. No $\overline{ACK}$ signal is output if the received address is not a local address (n = 0, 1).

The $\overline{ACK}$ signal output method during data reception is based on the wait timing setting, as described below.

When 8-clock wait is selected: $\overline{ACK}$ signal is output when ACKEn is set to 1 before wait cancellation.

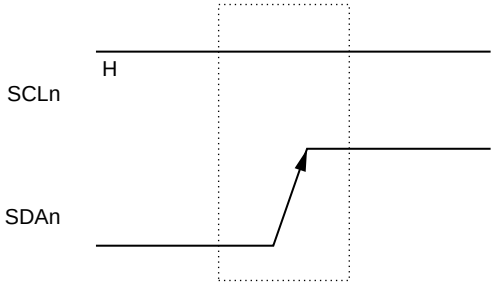
When 9-clock wait is selected: $\overline{ACK}$ signal is automatically output at the falling edge of the SCLn's eighth clock if ACKEn has already been set to 1.

(5) Stop condition

When the SCLn pin is at high level, changing the SDAn pin from low level to high level generates a stop condition (n = 0, 1).

A stop condition is a signal that the master device outputs to the slave device when serial transfer has been completed. The slave device includes hardware that detects stop conditions.

Figure 10-21. Stop Condition



Remark n = 0, 1

A stop condition is generated when bit 0 (SPTn) of the IIC control register n (IICCn) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IIC status register n (IICSn) is set to 1 and INTIICn is generated when bit 4 (SPIEn) of IICCn is set to 1 (n = 0, 1).

(6) Wait signal ($\overline{\text{WAIT}}$)

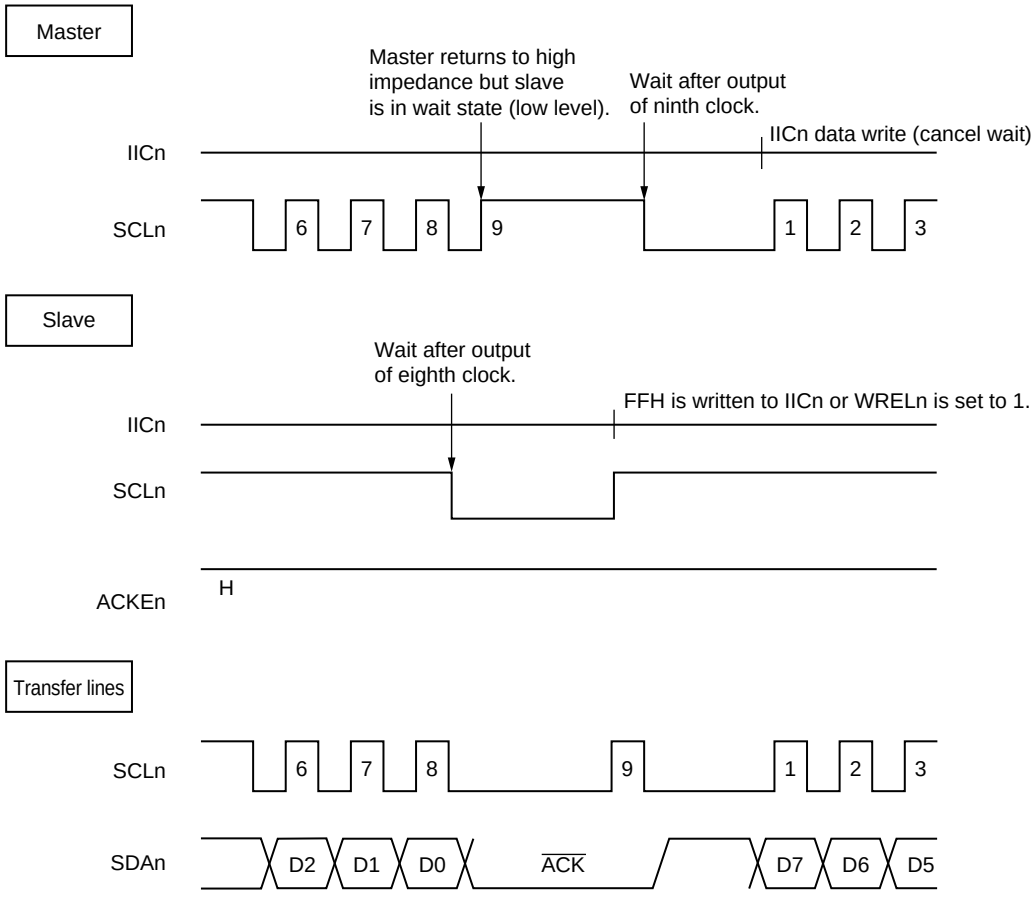
The wait signal ($\overline{\text{WAIT}}$) is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCLn pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin (n = 0, 1).

Figure 10-22. 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)

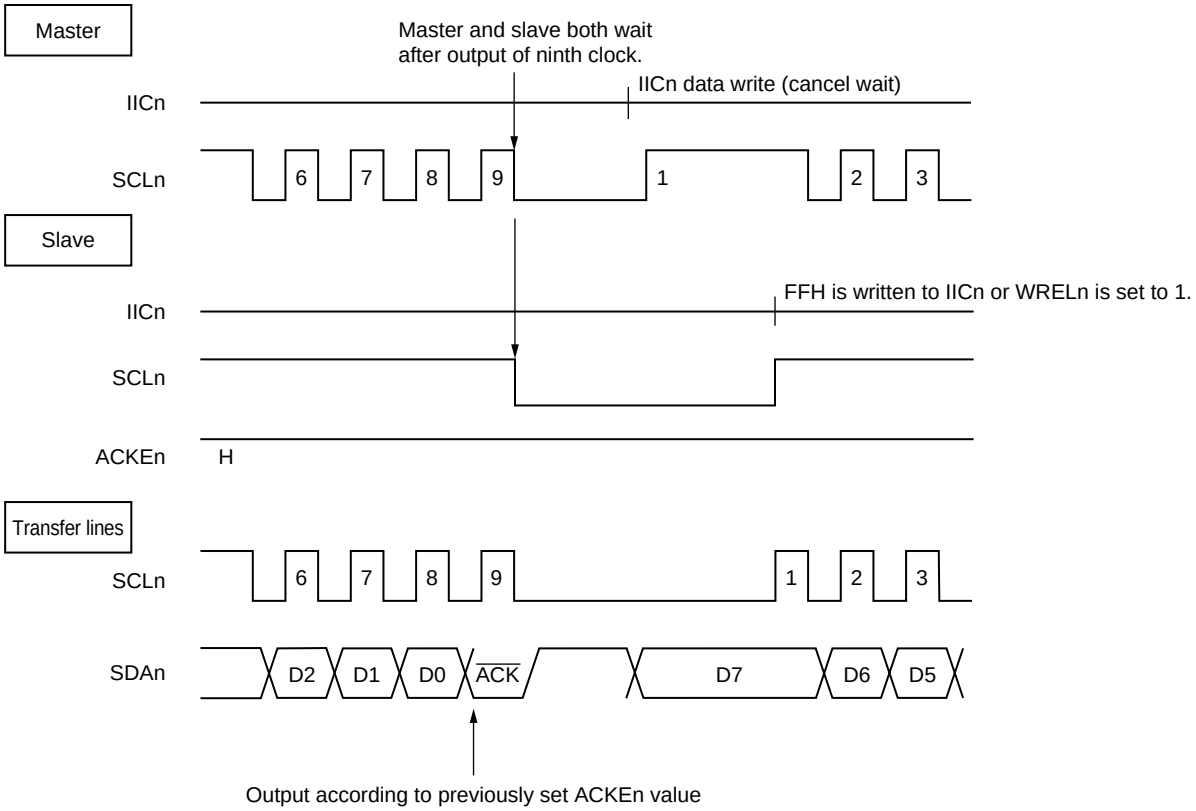


Remark n = 0, 1

Figure 10-22. 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 for bit 3 (WTIMn) of the 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 the IIC shift register n (IICn), the wait status is canceled and the transmitting side writes data to IICn to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting bit 1 (STTn) of IICCn to 1
- By setting bit 0 (SPTn) of IICCn to 1

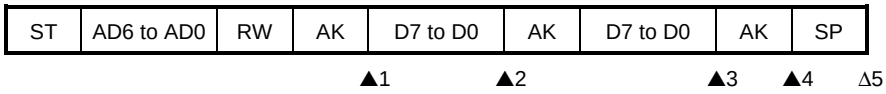
10.3.5 I^2C interrupt requests (INTIICn)

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 = 1000X110B

▲2: IICSn = 1000X000B

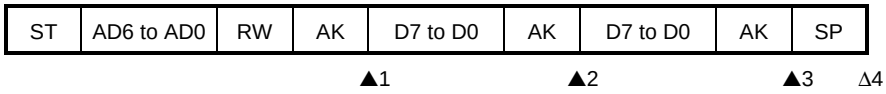
▲3: IICSn = 1000X000B (WTIMn is set)

▲4: IICSn = 1000XX00B (SPTn is set)

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 = 1000X110B

▲2: IICSn = 1000X100B

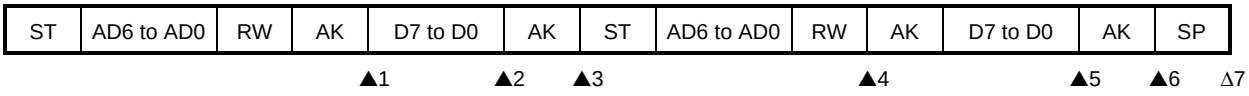
▲3: IICSn = 1000XX00B (SPTn is set)

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 = 1000X110B

▲2: IICSn = 1000X000B (WTIMn is set)

▲3: IICSn = 1000XX00B (WTIMn is cleared; STTn is set)

▲4: IICSn = 1000X110B

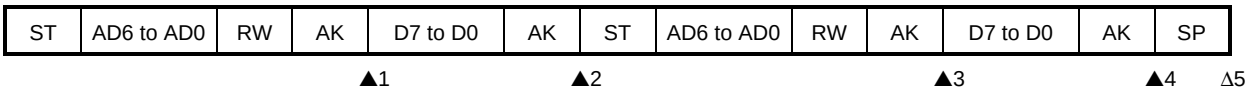
▲5: IICSn = 100XX000B (WTIMn is set)

▲6: IICSn = 1000XX00B (SPTn is set)

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 = 1000X110B

▲2: IICSn = 1000XX00B (STTn is set)

▲3: IICSn = 1000X110B

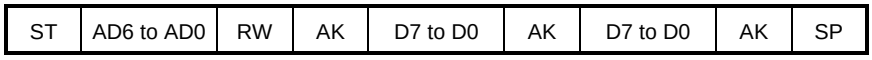
▲4: IICSn = 1000XX00B(SPTn is set)

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 ▲2 ▲3 ▲4 Δ5

▲1: IICSn = 1010X110B

▲2: IICSn = 1010X000B

▲3: IICSn = 1010X000B (WTIMn is set)

▲4: IICSn = 1010XX00B (SPTn is set)

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 ▲2 ▲3 Δ4

▲1: IICSn = 1010X110B

▲2: IICSn = 1010X100B

▲3: IICSn = 1010XX00B (SPTn is set)

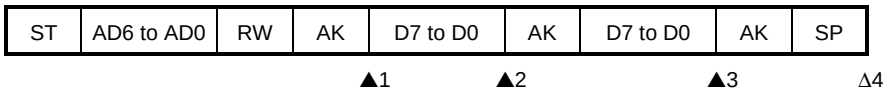
Δ 4: IICSn = 00000001B

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



▲1: IICSn = 0001X110B

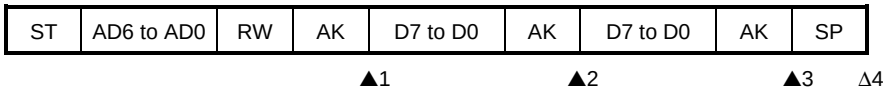
▲2: IICSn = 0001X000B

▲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



▲1: IICSn = 0001X110B

▲2: IICSn = 0001X100B

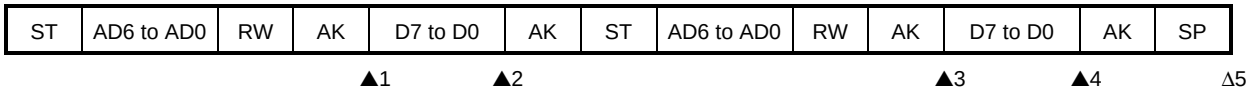
▲3: IICSn = 0001XX00B

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, matches with SVAn)



▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

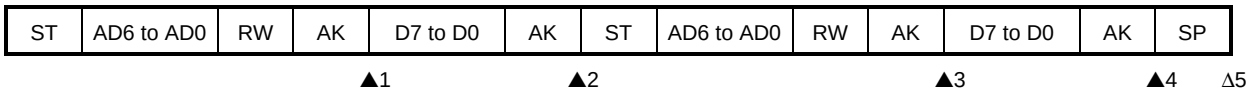
▲3: IICSn = 0001X110B

▲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, matches with SVAn)



▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0001X110B

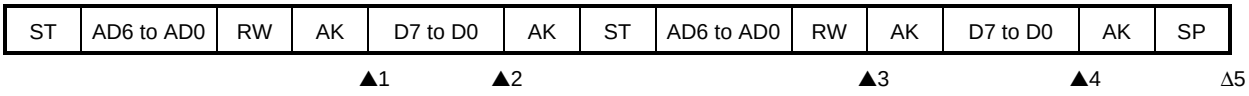
▲4: IICSn = 0001XX00B

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)



▲1: IICSn = 0001X110B

▲2: IICSn = 0001X000B

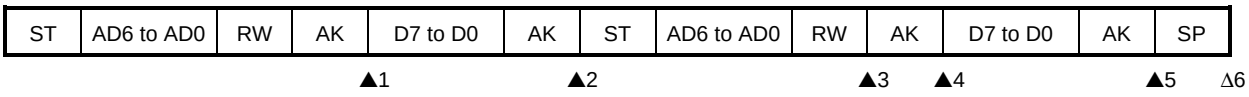
▲3: IICSn = 0010X010B

▲4: IICSn = 0010X000B

∆ 5: IICSn = 00000001B

Remark ▲: Always generated
∆: Generated only when SPIEn = 1
X: Don't care
n = 0, 1

<2> When WTIMn = 1 (after restart, extension code reception)



▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0010X010B

▲4: IICSn = 0010X110B

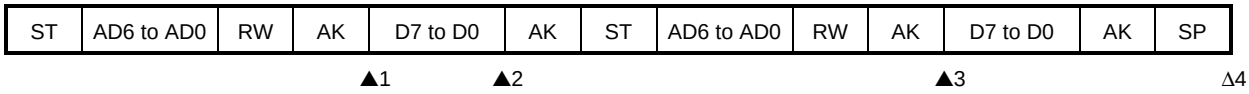
▲5: IICSn = 0010XX00B

∆ 6: IICSn = 00000001B

Remark ▲: Always generated
∆: 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, does not match with address (= not extension code))



▲1: IICSn = 0001X110B

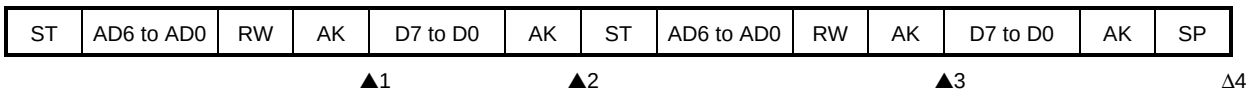
▲2: IICSn = 0001X000B

▲3: IICSn = 0000XX10B

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, does not match with address (= not extension code))



▲1: IICSn = 0001X110B

▲2: IICSn = 0001XX00B

▲3: IICSn = 0000XX10B

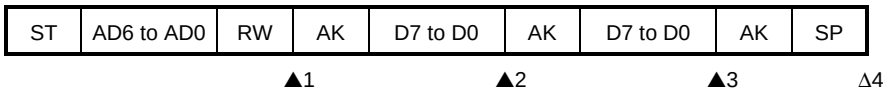
Δ 4: IICSn = 00000001B

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



▲1: IICSn = 0010X010B

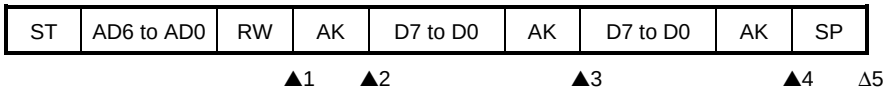
▲2: IICSn = 0010X000B

▲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



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010X100B

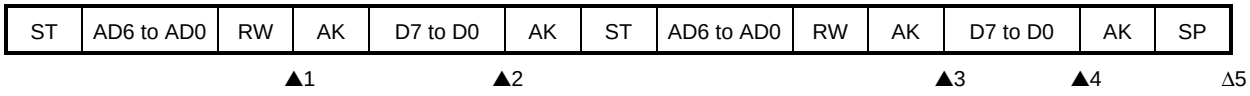
▲4: IICSn = 0010XX00B

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, matches with SVAn)



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

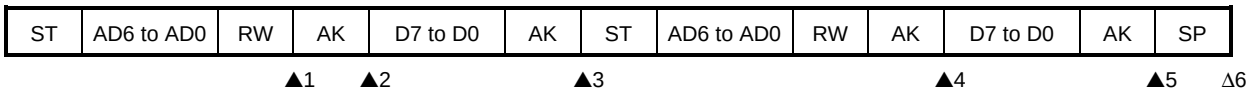
▲3: IICSn = 0001X110B

▲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, matches with SVAn)



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0001X110B

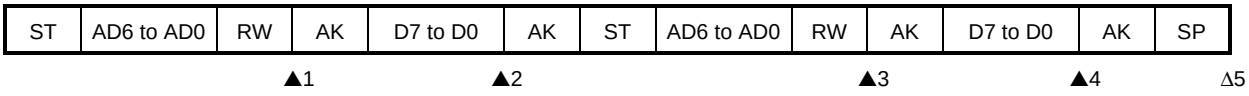
▲5: IICSn = 0001XX00B

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)



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X000B

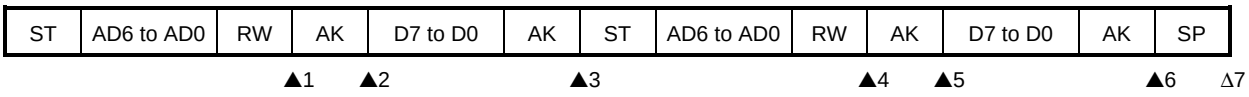
▲3: IICSn = 0010X010B

▲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)



▲1: IICSn = 0010X010B

▲2: IICSn = 0010X110B

▲3: IICSn = 0010XX00B

▲4: IICSn = 0010X010B

▲5: IICSn = 0010X110B

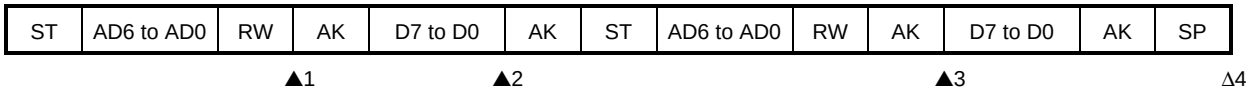
▲6: IICSn = 0010XX00B

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, does not match with address (= not extension code))



▲1: IICSn = 0010X010B

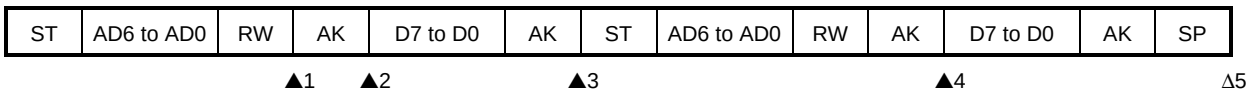
▲2: IICSn = 0010X000B

▲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, does not match 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)

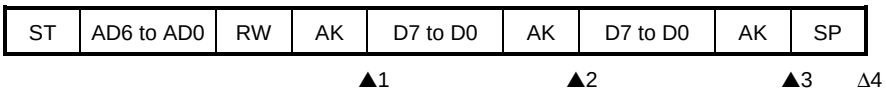
▲2: IICSn = 0001X000B

▲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



▲1: IICSn = 0101X110B (Example: when ALDn is read during interrupt servicing)

▲2: IICSn = 0001X100B

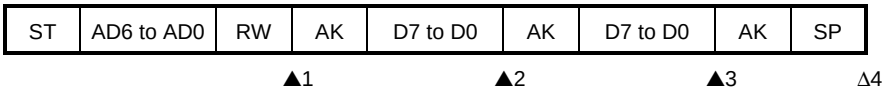
▲3: IICSn = 0001XX00B

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)

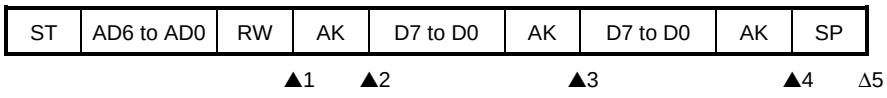
▲2: IICSn = 0010X000B

▲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

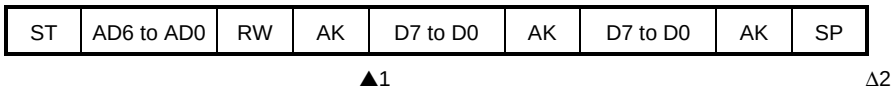
▲4: IICSn = 0010XX00B

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 (when WTIMn = 1)

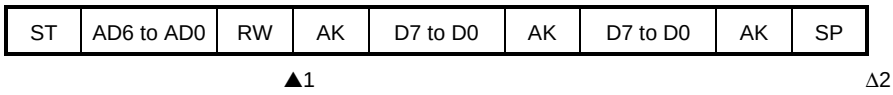


▲1: IICSn = 01000110B (Example: when ALDn is read during interrupt servicing)

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



▲1: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

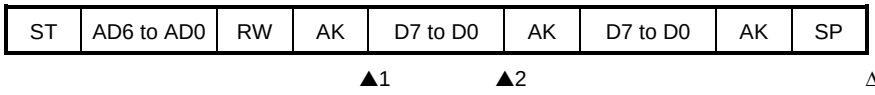
IICCn: LRELn is set to 1 by software

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



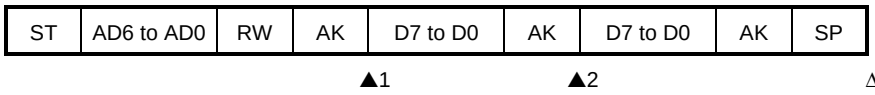
▲1: IICSn = 10001110B

▲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 loss occurs due to restart condition during data transfer

<1> Not extension code (Example: matches with SVAn)

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 Δ3

▲1: IICSn = 1000X110B

▲2: IICSn = 01000110B (Example: when ALDn is read during interrupt servicing)

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: Don't care
Dn = D6 to D0
n = 0, 1

<2> Extension code

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP

▲1 ▲2 Δ3

▲1: IICSn = 1000X110B

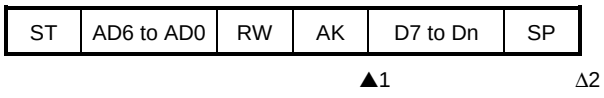
▲2: IICSn = 0110X010B (Example: when ALDn is read during interrupt servicing)

IICCn: LRELn is set to 1 by software

Δ 3: IICSn = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: Don't care
Dn = D6 to D0
n = 0, 1

(e) When loss occurs due to stop condition during data transfer



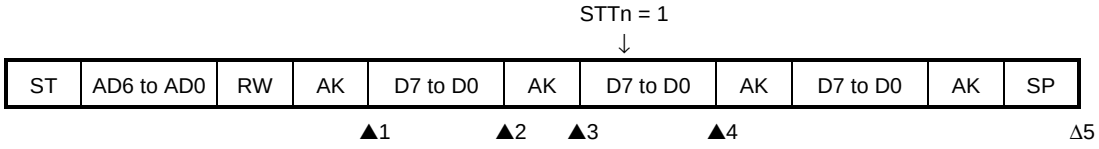
▲1: IICSn = 1000X110B

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

<1> When WTIMn = 0



▲1: IICSn = 1000X110B

▲2: IICSn = 1000X000B (WTIMn is set)

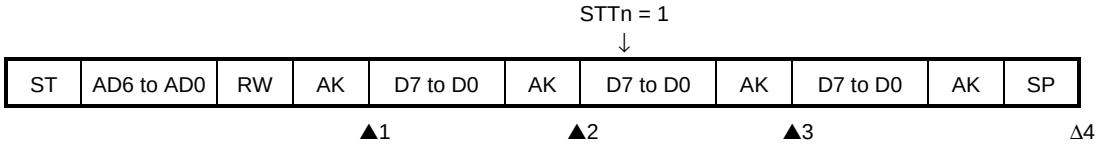
▲3: IICSn = 1000XX00B (WTIMn is cleared; SSTn is set)

▲4: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

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 = 1000X110B

▲2: IICSn = 1000XX00B (STTn is set)

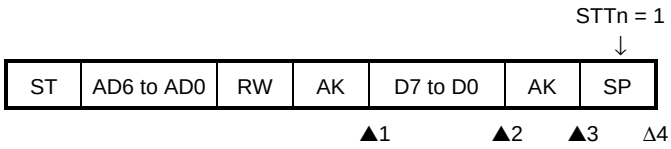
▲3: IICSn = 01000100B (Example: when ALDn is read during interrupt servicing)

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

<1> When WTIMn = 0



▲1: IICSn = 1000X110B

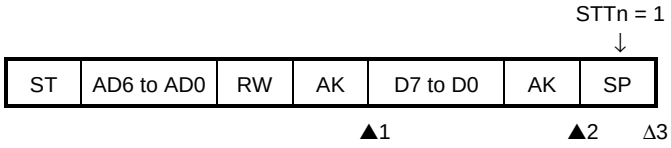
▲2: IICSn = 1000X000B (WTIMn is set)

▲3: IICSn = 1000XX00B (STTn is set)

Δ 4: IICSn = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIEn = 1
X: Don't care
n = 0, 1

<2> When WTIMn = 1



▲1: IICSn = 1000X110B

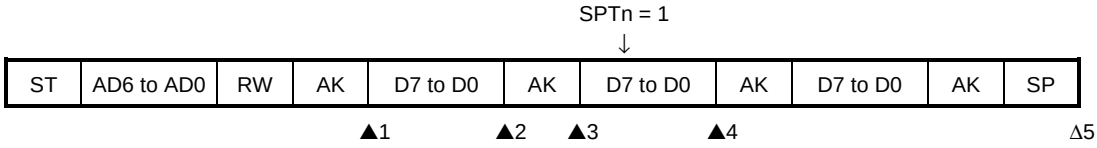
▲2: IICSn = 1000XX00B (STTn is set)

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

<1> When WTIMn = 0



▲1: IICSn = 1000X110B

▲2: IICSn = 1000X000B (WTIMn is set)

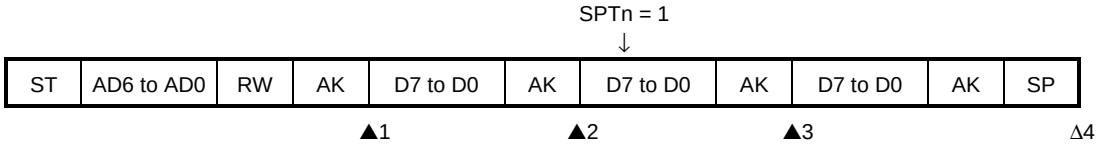
▲3: IICSn = 1000XX00B (WTIMn is cleared; SPTn is set)

▲4: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

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 = 1000X110B

▲2: IICSn = 1000XX00B (SPTn is set)

▲3: IICSn = 01000000B (Example: when ALDn is read during interrupt servicing)

Δ 4: IICSn = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

X: Don't care

n = 0, 1

10.3.6 Interrupt request (INTIICn) generation timing and wait control

The setting of bit 3 (WTIMn) in the IIC control register n (IICCn) determines the timing by which INTIICn is generated and the corresponding wait control, as shown in Table 10-3 (n = 0, 1).

Table 10-3. 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 the slave address register (SVAn).
At this point, $\overline{ACK}$ is output regardless of the value set to IICCn's bit 2 (ACKEn). For a slave device that has received an extension code, INTIICn occurs at the falling edge of the eighth clock.

2. If the received address does not match the contents of the slave address register (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 setting start condition (by setting bit 1 (STTn) of IIC control register n (IICCn) to 1)
- By setting stop condition (by setting bit 0 (SPTn) of IIC control register n (IICCn) to 1)

When 8-clock wait has been selected (WTIMn = 0), the output level of $\overline{ACK}$ must be determined prior to wait cancellation.

Remark n = 0, 1

(5) Stop condition detection

INTIICn is generated when a stop condition is detected.

Remark n = 0, 1

10.3.7 Address match detection method

When in I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An interrupt request (INTIICn) occurs when a local address has been set to the slave address register n (SVAn) and when the address set to SVAn matches the slave address sent by the master device, or when an extension code has been received (n = 0, 1).

10.3.8 Error detection

In I^2C bus mode, the status of the serial data bus (SDAn) during data transmission is captured by the IIC shift register n (IICn) of the transmitting device, so the IICn data prior to transmission can be compared with the transmitted IICn data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match (n = 0, 1).

10.3.9 Extension code

(1) When the high-order 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 the 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).

- High-order 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 10-4. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for different bus format
1111 0xx	X	10-bit slave address specification

10.3.10 Arbitration

When several master devices simultaneously output a start condition (when STTn is set to 1 before STDn is set to 1[Note]), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration (n = 0, 1).

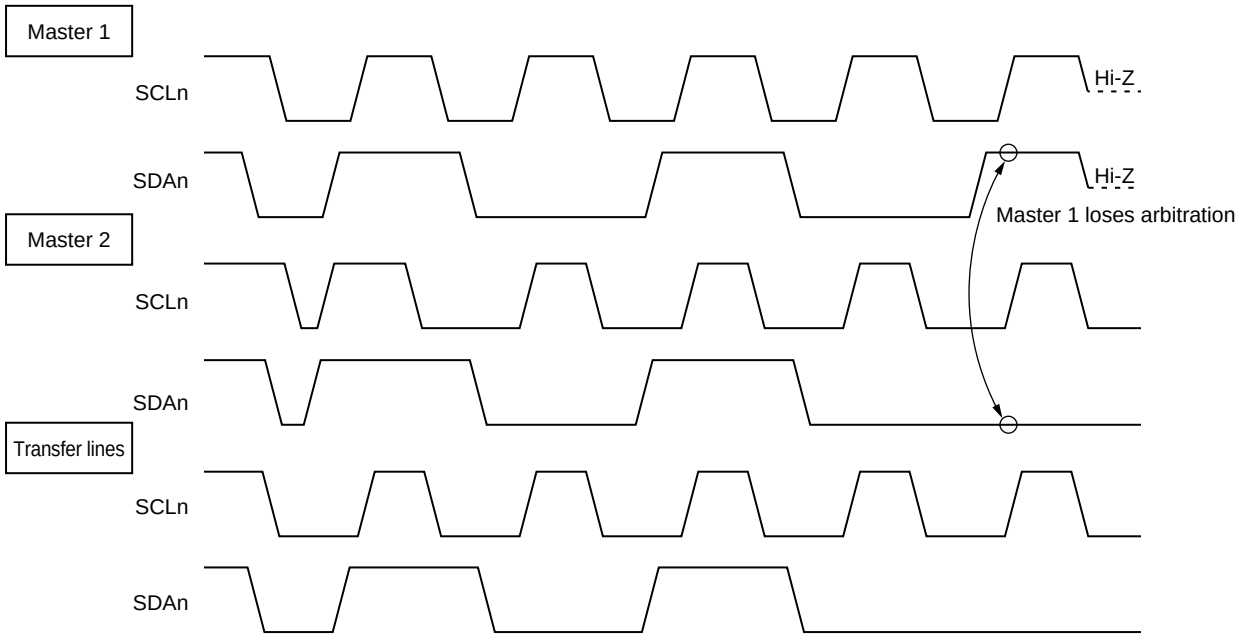
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IIC status register n (IICSn) is set via the timing by which the arbitration loss occurred, and the SCLn and SDAn lines are both set for high impedance, which releases the bus (n = 0, 1).

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALDn = 1 setting that has been made by software (n = 0, 1).

For details of interrupt request timing, refer to **10.3.5 I²C interrupt requests (INTIICn)**.

Note STDn: Bit 1 of IIC status register n (IICSn)
STTn: Bit 1 of IIC control register n (IICCn)

Figure 10-23. Arbitration Timing Example



Remark n = 0, 1

Table 10-5. Status during Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ signal transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIEn = 1)[Note 2]
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIEn = 1)[Note 2]
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCLn is at low level while attempting to output a restart condition	

Notes 1. When WTIMn (bit 3 of the IIC control register n (IICCn)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock (n = 0, 1).

2. When there is a possibility that arbitration will occur, set SPIEn = 1 for master device operation (n = 0, 1).

Remark SPIEn: Bit 5 of the IIC control register n (IICCn)

10.3.11 Wake up 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 the 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).

10.3.12 Communication reservation

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when bit 6 (LRELn) of the 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 the IIC shift register n (IICn) causes the master’s address transfer to start. At this point, IICCn’s bit 4 (SPIEn) should be set (n = 0, 1).

When STTn has been set, the operation mode (as start condition or as communication reservation) is determined according to the bus status (n = 0, 1).

If the bus has been released.....a start condition is generated
If the bus has not been released (standby mode).....communication reservation

To detect which operation mode has been determined for STTn, set STTn, wait for the wait period, then check the MSTSn (bit 7 of the IIC status register n (IICSn)) (n = 0, 1).

Wait periods, which should be set via software, are listed in Table 10-6. These wait periods can be set via the settings for bits 3, 1, and 0 (SMCn, CLn1, and CLn0) in the IIC clock select register n (IICCLn) (n = 0, 1).

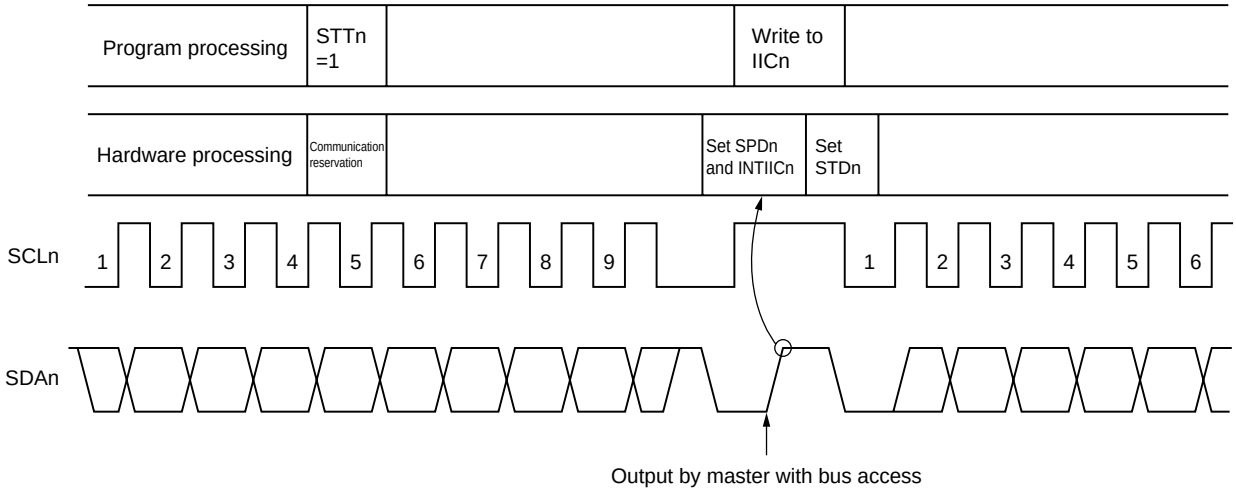
Table 10-6. Wait Periods

SMCn	CLn1	CLn0	Wait Period
0	0	0	26 clocks
0	0	1	46 clocks
0	1	0	
0	1	1	37 clocks
1	0	0	16 clocks
1	0	1	
1	1	0	
1	1	1	13 clocks

Remark n = 0, 1

Figure 10-24 shows the communication reservation timing.

Figure 10-24. Communication Reservation Timing

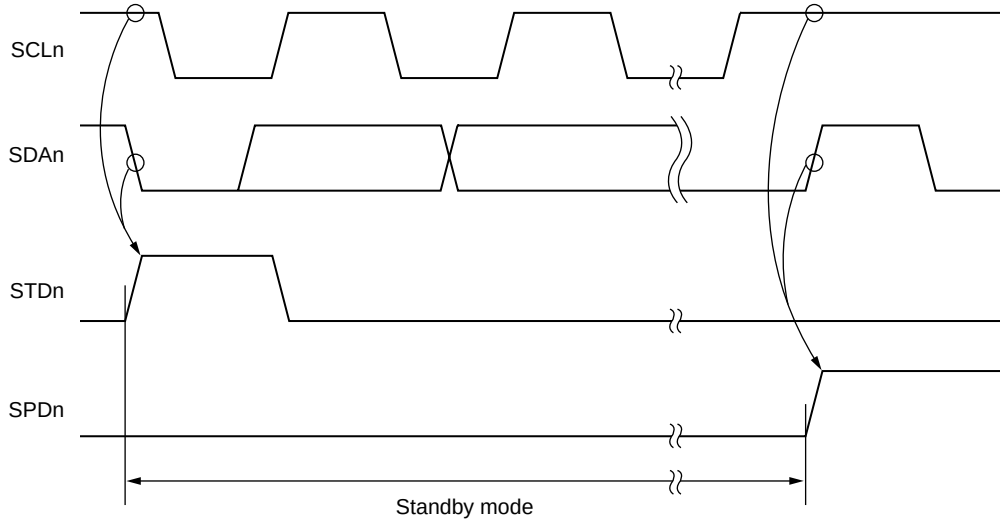


IICn: IIC shift register n
STTn: Bit 1 of IIC control register n (IICCn)
STDn: Bit 1 of IIC status register n (IICSn)
SPDn: Bit 0 of IIC status register n (IICSn)

Remark n = 0, 1

Communication reservations are accepted via the following timing. After bit 1 (STDn) of the IIC status register n (IICSn) is set to “1”, a communication reservation can be made by setting bit 1 (STTn) of the IIC control register n (IICCn) to “1” before a stop condition is detected (n = 0, 1).

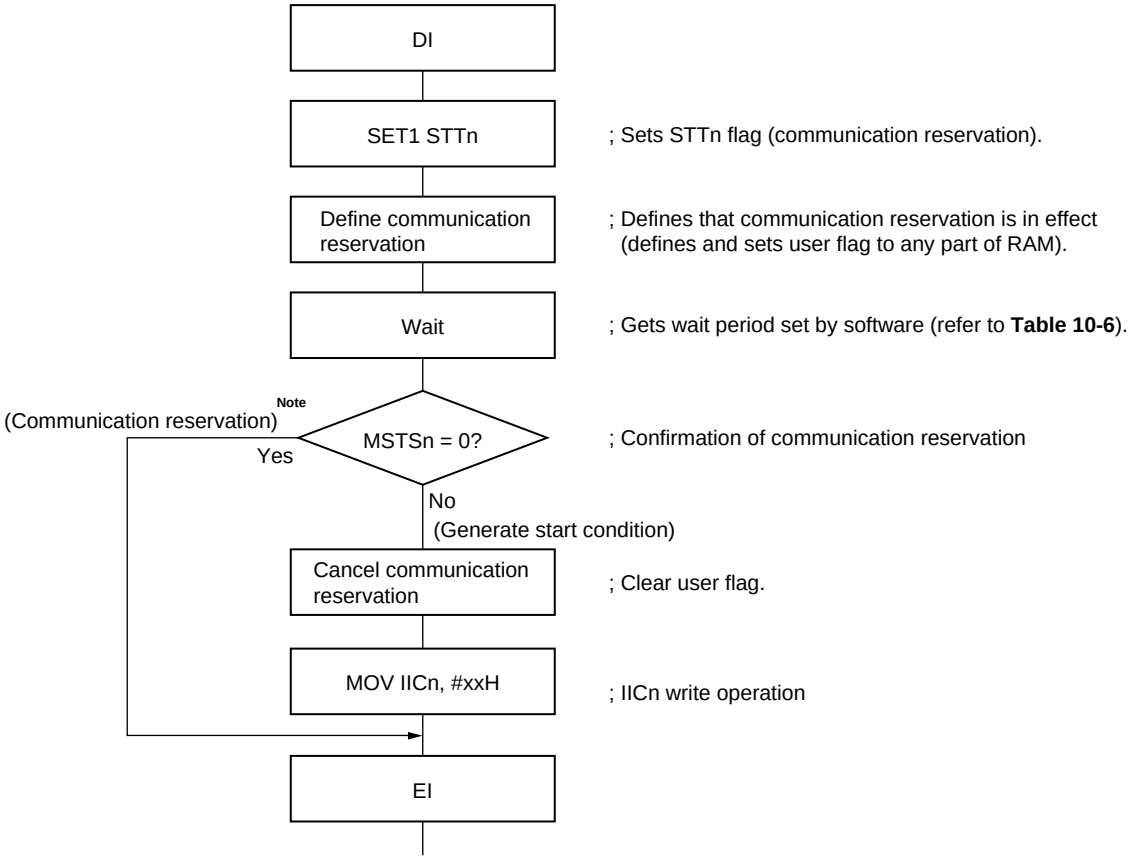
Figure 10-25. Timing for Accepting Communication Reservations



Remark n = 0, 1

The communication reservation flow chart is illustrated in Figure 10-26.

Figure 10-26. Communication Reservation Flow Chart



Note The communication reservation operation executes a write to the IIC shift register n (IICn) when a stop condition interrupt request occurs.

Remark n = 0, 1

10.3.13 Other cautions

After a reset, when changing from a mode in which no stop condition has been detected (the bus has not been released) to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

(a) Set IIC clock select register n (IICCLn).
(b) Set bit 7 (IICEn) of the IIC control register (IICCn).
(c) Set bit 0 of IICCn.

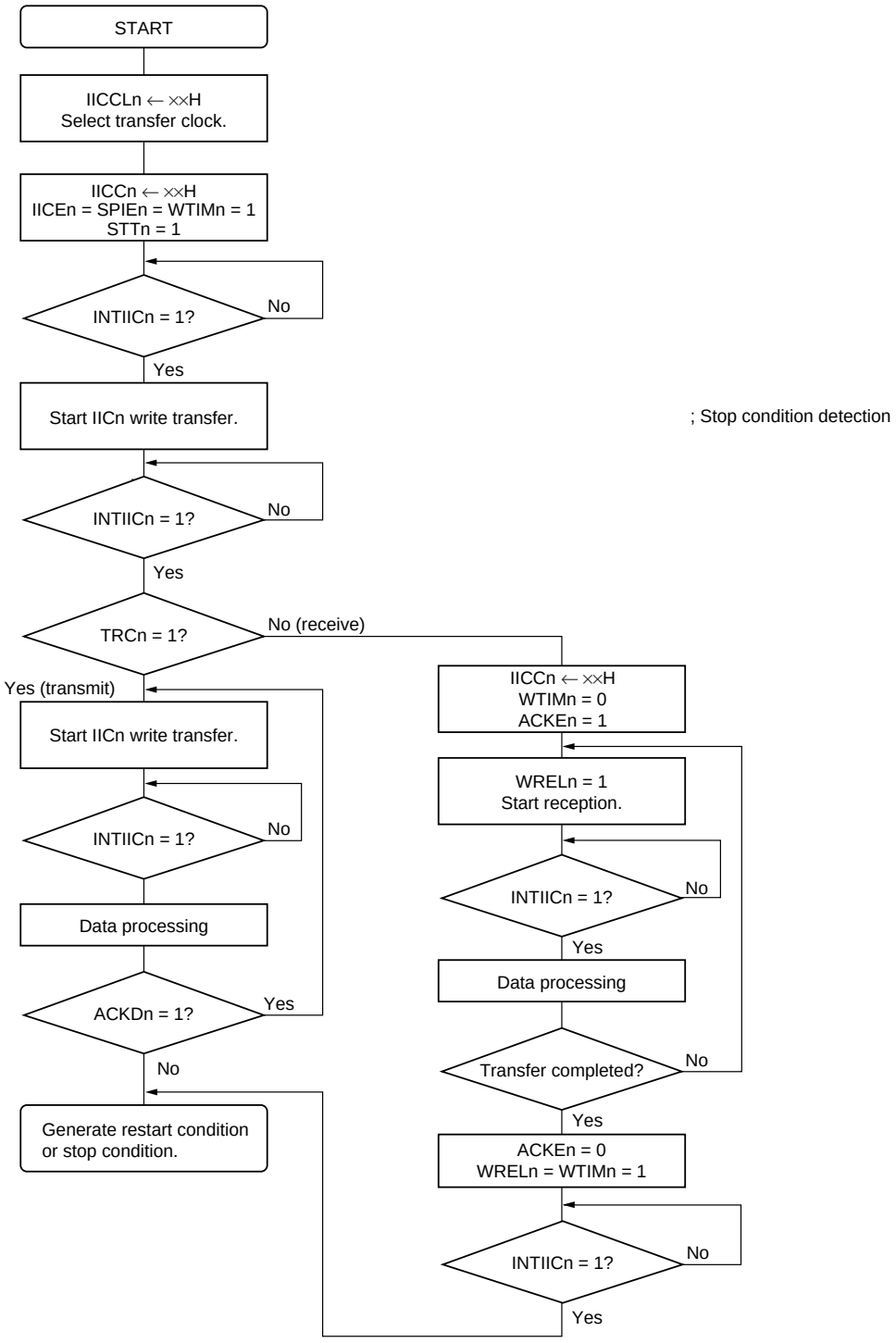
Remark n = 0, 1

10.3.14 Communication operations

(1) Master operations

The following is a flow chart of the master operations.

Figure 10-27. Master Operation Flow Chart

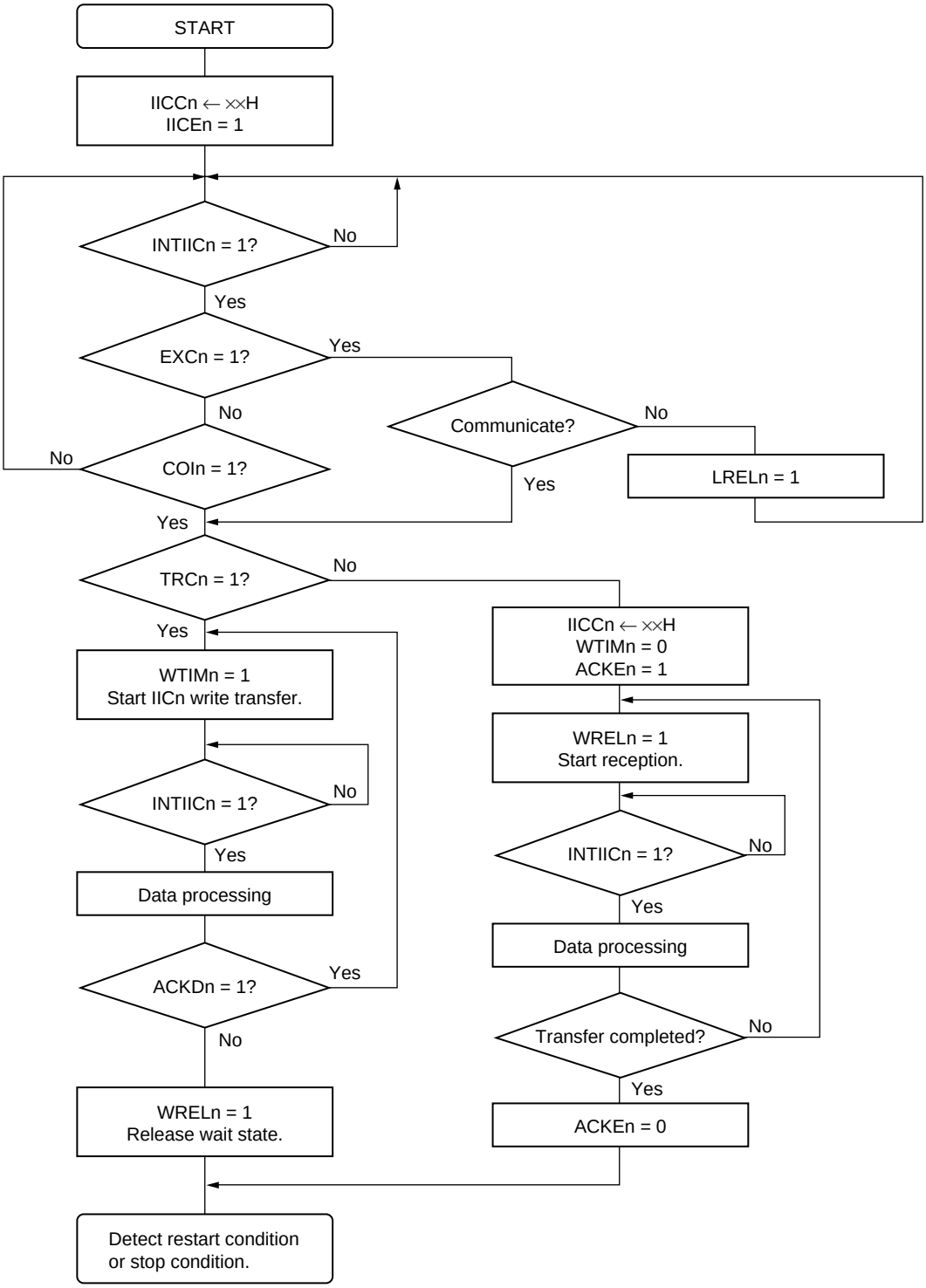


Remark n = 0, 1

(2) Slave operation

An example of slave operation is shown below.

Figure 10-28. Slave Operation Flow Chart



Remark n = 0, 1

10.3.15 Timing of data communication

When using I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRCn bit (bit 3 of the IIC status register n (IICSn)) that specifies the data transfer direction and then starts serial communication with the slave device.

Figures 10-29 and 10-30 show timing charts of the data communication.

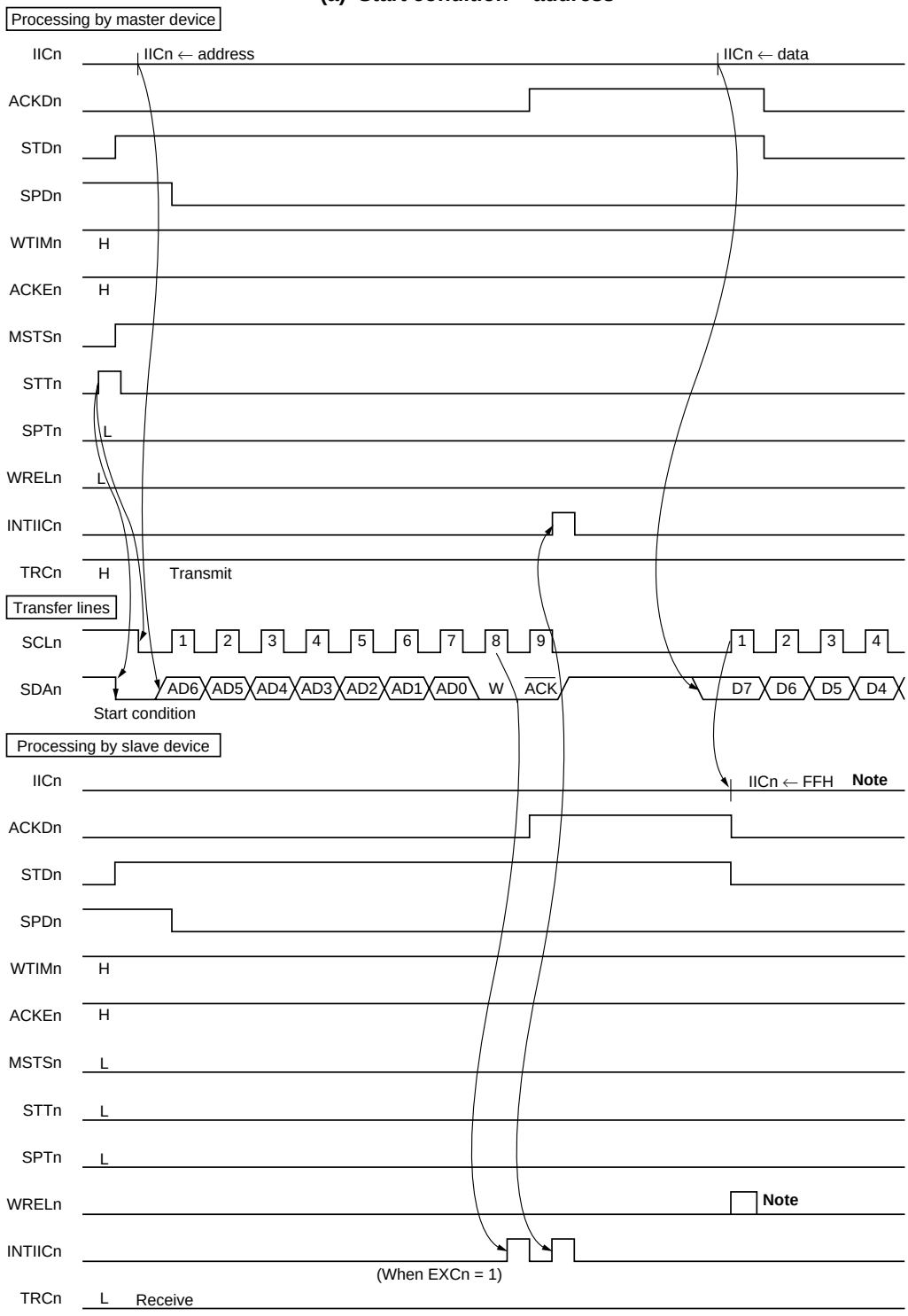
The IIC bus shift register n (IICn)'s shift operation is synchronized with the falling edge of the serial clock (SCLn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAn pin.

Data input via the SDAn pin is captured by IICn at the rising edge of SCLn.

Remark n = 0, 1

Figure 10-29. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address

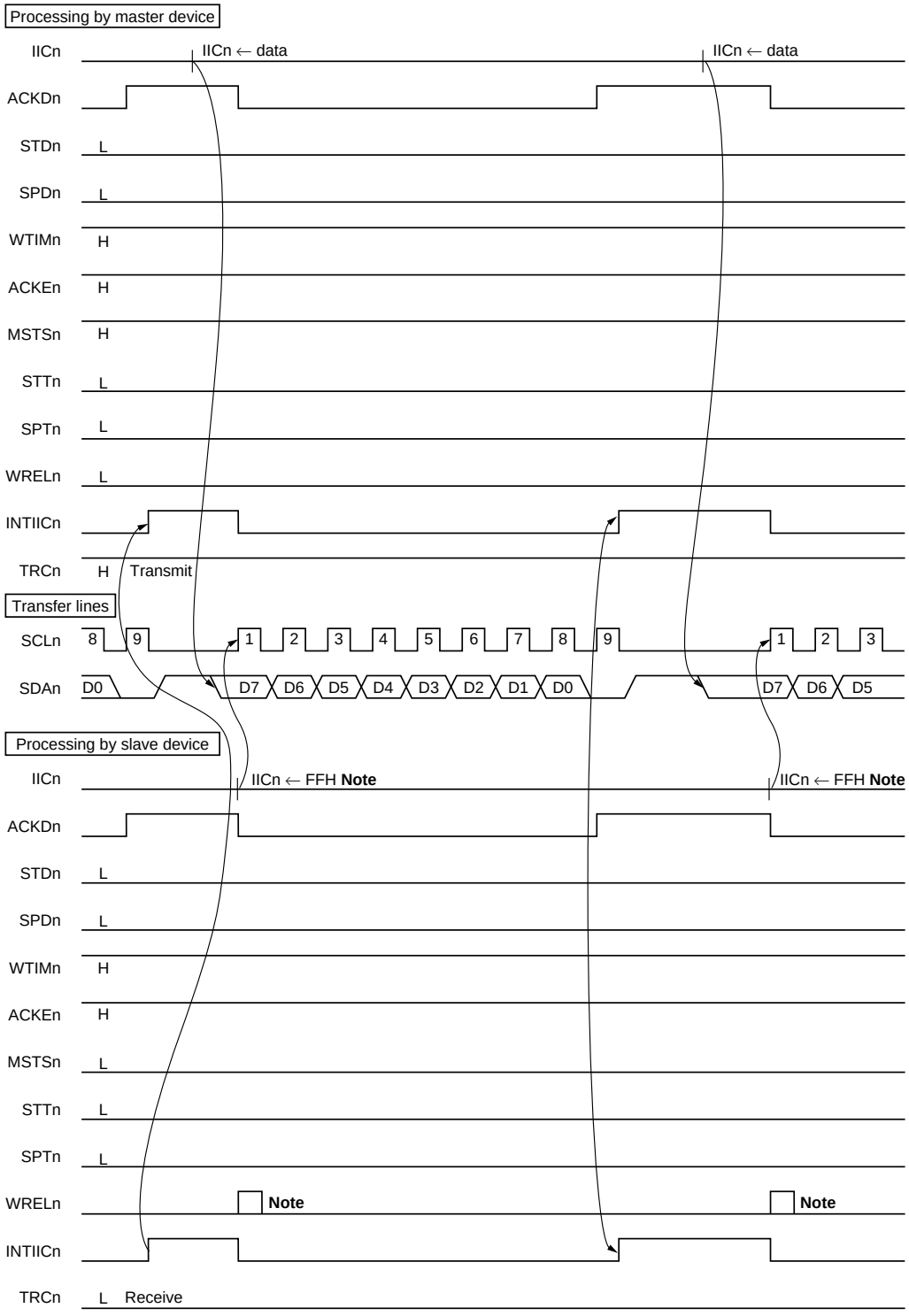


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-29. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

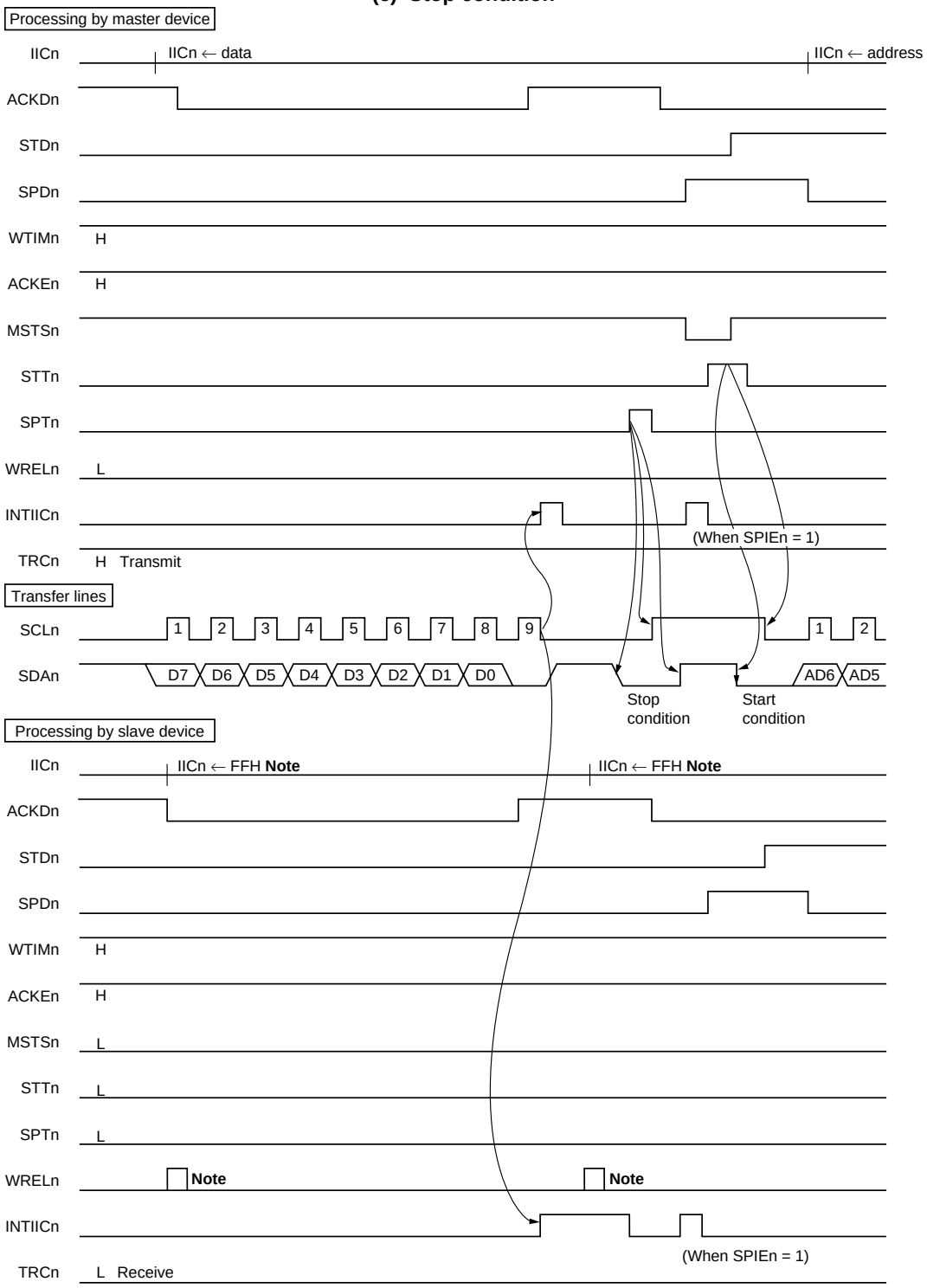


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-29. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition

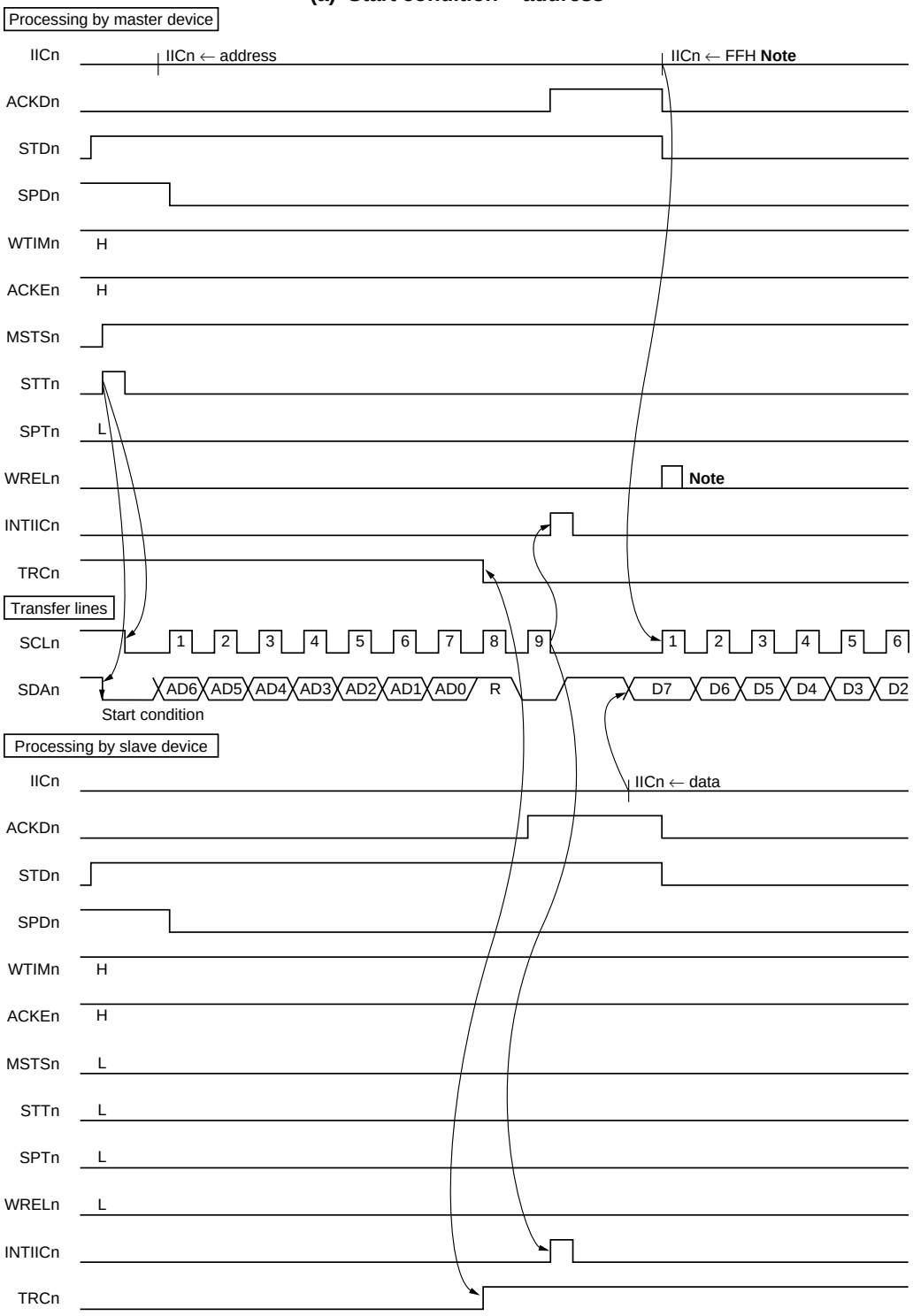


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-30. Example of Slave to Master Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(a) Start condition ~ address

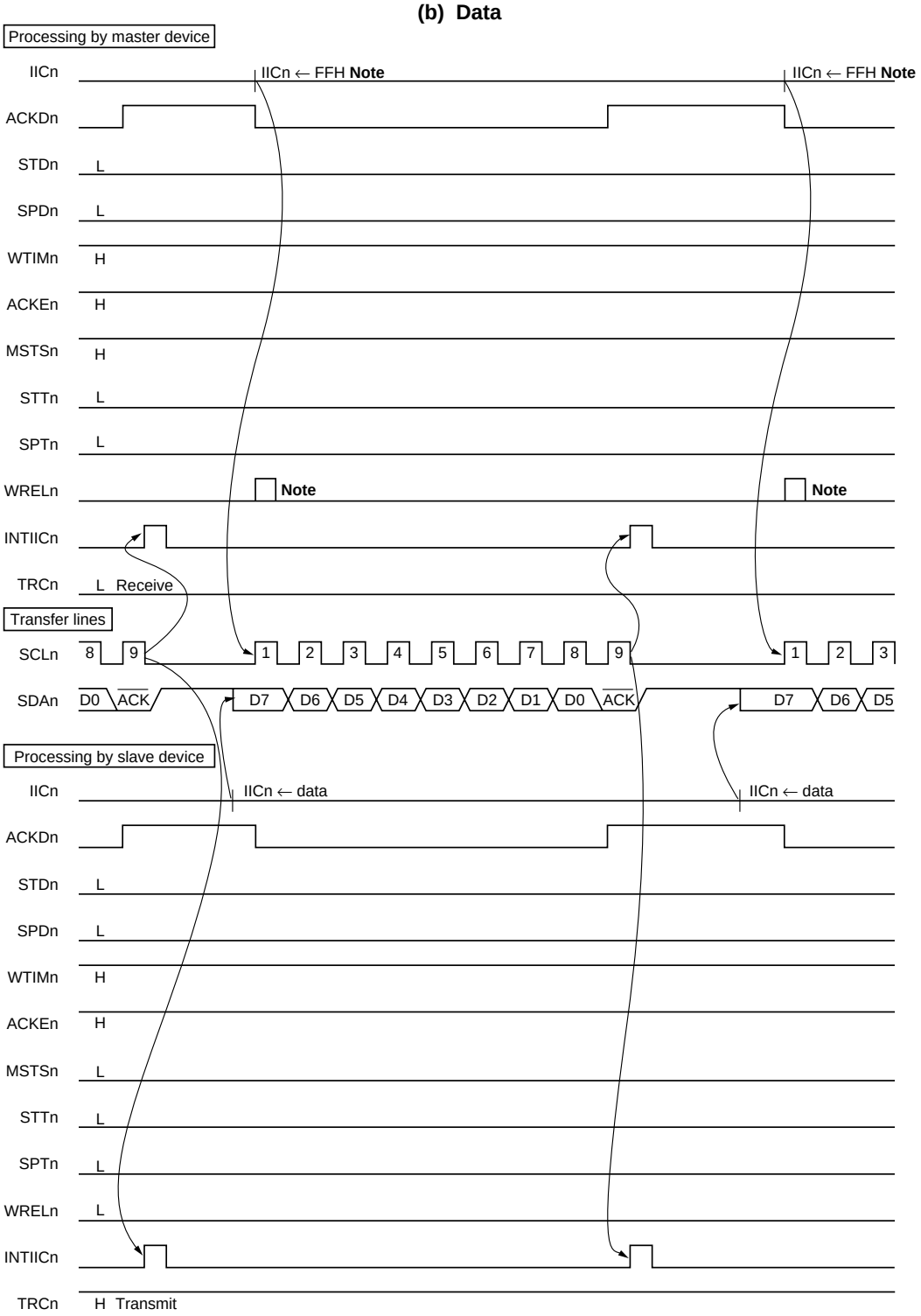


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-30. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(b) Data

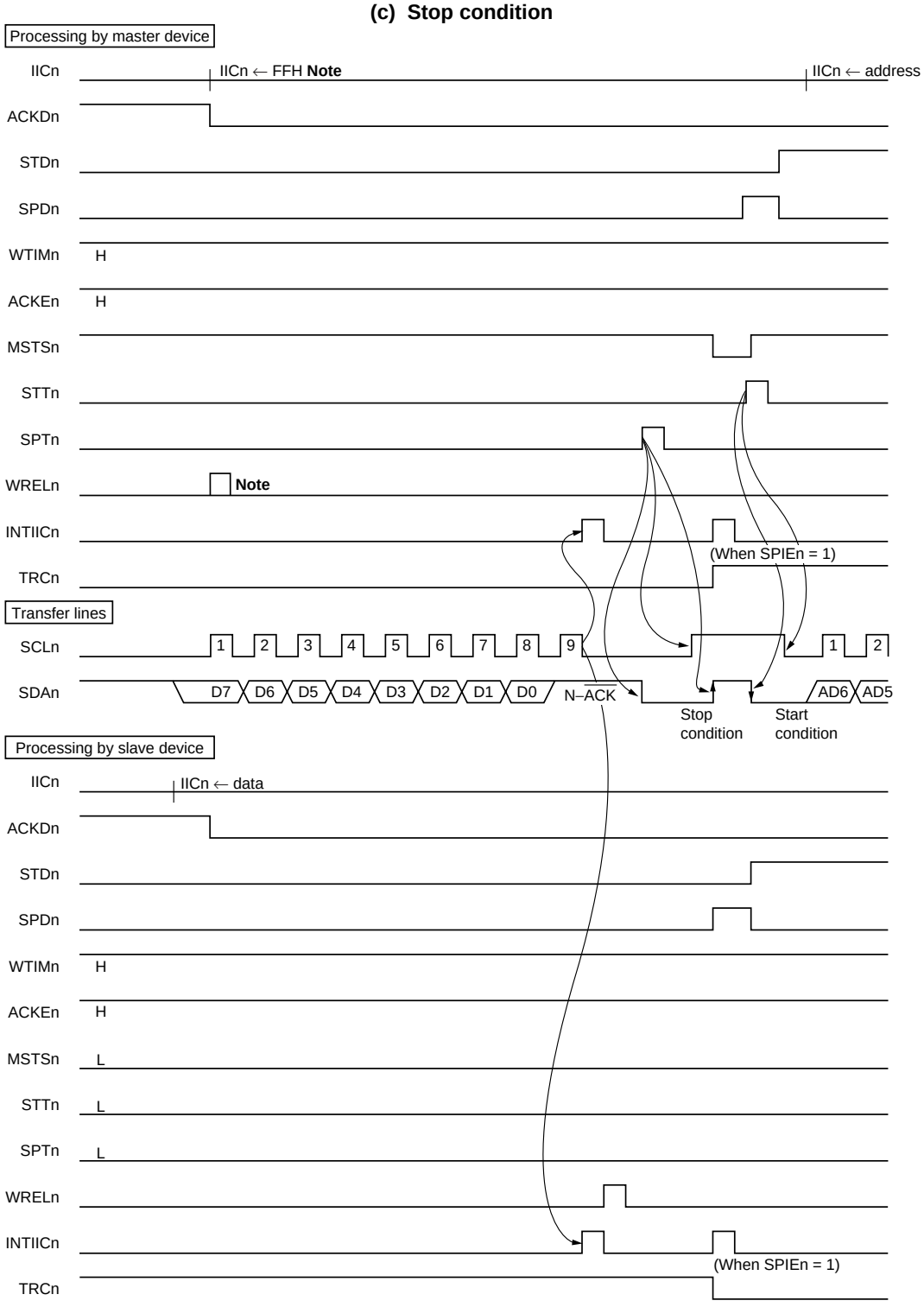


Note To cancel slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

Figure 10-30. 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 slave wait, write FFH to IICn or set WRELn.

Remark n = 0, 1

10.4 Asynchronous Serial Interface (UART0, UART1)

The UARTn (n = 0, 1) has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) Asynchronous serial interface (UARTn) mode

This mode enables full-duplex operation which transmits and receives one byte of data after the start bit.

The on-chip dedicated UARTn baud rate generator enables communications using a wide range of selectable baud rates. In addition, a baud rate based on divided clock input to the ASCKn pin can also be defined.

The UARTn baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

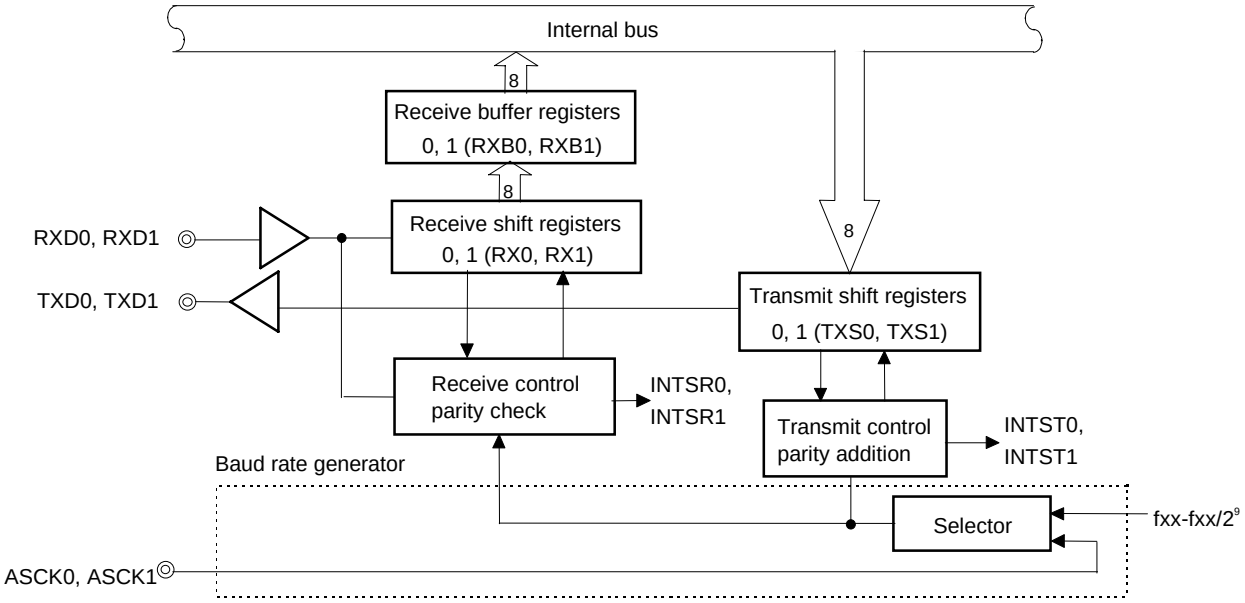
10.4.1 Configuration

The UARTn includes the following hardware.

Table 10-7. Configuration of UARTn

Item	Configuration
Registers	Transmit shift registers 0, 1 (TXS0, TXS1) Receive buffer registers 0, 1 (RXB0, RXB1)
Control registers	Asynchronous serial interface mode registers 00, 10 (ASIM00, ASIM10) Asynchronous serial interface mode registers 01, 11 (ASIM01, ASIM11) Asynchronous serial interface status registers 0, 1 (ASIS0, ASIS1) Baud rate generator control registers 0, 1 (BRGC0, BRGC1) Baud rate generator mode control registers 00, 01 (BRGMC00, BRGMC01) Baud rate generator mode control registers 10, 11 (BRGMC10, BRGMC11)

Figure 10-31. Block Diagram of UARTn



(1) Transmit shift registers 0, 1 (TXS0, TXS1)

TXSn is the register for setting transmit data. Data written to TXSn is transmitted as serial data.

When the data length is set 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 can be written to by an 8-bit memory manipulation instruction. It cannot be read from.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Caution Do not write to TXSn during a transmit operation.

(2) Receive shift registers 0, 1 (RX0, RX1)

RXn register converts serial data input via the RXD0, RXD1 pins to parallel data. When one byte of data is received at RXn, the received data is transferred to the receive buffer register (RXB0, RXB1).

RX0, RX1 cannot be manipulated directly by a program.

(3) Receive buffer registers 0, 1 (RXB0, RXB1)

RXBn is used to hold receive data. When one byte of data is received, one byte of new receive data is transferred.

When the data length is set 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 can be read by an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input sets RXBn to FFH.

(4) Transmission control circuit

The transmission control circuit controls transmit operations, such as adding a start bit, parity bit, and stop bit to data that is written to the transmit shift register (TXSn), based on the values set to the asynchronous serial interface mode register n0 (ASIMn0).

(5) Reception control circuit

The reception control circuit controls receive operations based on the values set to the asynchronous serial interface mode register n0 (ASIMn0). During a receive operation, it performs error checking, such as for parity errors, and sets various values to the asynchronous serial interface status register (ASISn) according to the type of error that is detected.

10.4.2 UARTn control registers

The UARTn uses the following four types of registers for control function (n = 0, 1).

- Asynchronous serial interface mode register n0 (ASIMn0)
- Asynchronous serial interface mode register n1 (ASIMn1)
- 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 00, 10 (ASIM00, ASIM10)

ASIMn0 is an 8-bit register that controls UARTn’s serial transfer operations.

ASIMn0 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-32. Asynchronous Serial Interface Mode Registers 00, 10 (ASIM00, ASIM10)

After reset: 00H R/W Address: FFFFF300H, FFFFF310H

	7	6	5	4	3	2	1	0
ASIMn0 (n = 0, 1)	TXEn	RXEn	PS1n	PS0n	UCLn	SLn	ISRMn	0

TXEn	RXEn	Operation Mode	RXDn/Pxx Pin Function	TXDn/Pxx Pin Function
0	0	Operation stop	Port function	Port function
0	1	UARTn mode (receive only)	Serial function	Port function
1	0	UARTn mode (transmit only)	Port function	Serial function
1	1	UARTn mode (transmit and receive)	Serial function	Serial function

PS1n	PS0n	Parity Bit Specification
0	0	No parity
0	1	Zero parity always added during transmission No parity detection during reception (parity errors do not occur)
1	0	Odd parity
1	1	Even parity

UCLn	Character Length Specification
0	7 bits
1	8 bits

SLn	Stop Bit Length Specification for Transmit Data
0	1 bit
1	2 bits

ISRMn	Receive Completion Interrupt Control When Error Occurs
0	Receive completion interrupt is issued when an error occurs
1	Receive completion interrupt is not issued when an error occurs

Caution Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

(2) Asynchronous serial interface mode registers 01, 11 (ASIM01, ASIM11)

ASIMn1 is an 8-bit register that controls UARTn's serial transfer operations.

If an external circuit that may invert TXDn is connected to the TXDn pin, ASIMn1 should be used.

ASIMn1 can be set by a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-33 shows the format of ASIMn1.

Figure 10-33. Asynchronous Serial Interface Mode Registers 01, 11 (ASIM01, ASIM11)

After reset: 00H R/W Address: FFFFF324H, FFFFF326H

	7	6	5	4	3	2	1	0
ASIMn1 (n = 0, 1)	0	0	0	0	0	0	0	NOTn

NOTn	TXDn Output Level Inversion Control
0	Output level is not inverted.
1	Output level is inverted.

Caution When using P14/TXD0 or P24/TXD1 as a port function, set the NOT0 or NOT1 bit to 0.

(3) Asynchronous serial interface status registers 0, 1 (ASIS0, ASIS1)

When a receive error occurs in UARTn mode, these registers indicate the type of error.

ASISn can be read using a 1-bit or 8-bit memory manipulation instruction.

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

Figure 10-34. Asynchronous Serial Interface Status Registers 0, 1 (ASIS0, ASIS1)

After reset: 00H R Address: FFFFF302H, FFFFF312H

	7	6	5	4	3	2	1	0
ASISn	0	0	0	0	0	PEn	FEn	OVEn

(n = 0, 1)

PEn	Parity Error Flag
0	No parity error
1	Parity error (Transmit data parity does not match)

FEn	Framing Error Flag
0	No framing error
1	Framing error[Note 1] (Stop bit not detected)

OVEn	Overrun Error Flag
0	No overrun error
1	Overrun error[Note 2] (Next receive operation was completed before data was read from receive buffer register)

Notes 1. Even if a stop bit length has been set as two bits by setting bit 2 (SLn) in the asynchronous serial interface mode register (ASIMn0), 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 the receive buffer register (RXBn) when an overrun error has occurred.

Until the contents of RXBn are read, further overrun errors will occur when receiving data.

(4) Baud rate generator control registers 0, 1 (BRGC0, BRGC1)

These registers set the serial clock for UARTn.

BRGCn can be set by an 8-bit memory manipulation instruction.

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

Figure 10-35. Baud Rate Generator Control Registers 0, 1 (BRGC0, BRGC1)

After reset: 00H R/W Address: FFFFF304H, FFFFF314H

	7	6	5	4	3	2	1	0
BRGCn (n = 0, 1)	MDLn7	MDLn6	MDLn5	MDLn4	MDLn3	MDLn2	MDLn1	MDLn0

MD Ln7	MD Ln6	MD Ln5	MD Ln4	MD Ln3	MD Ln2	MD Ln1	MD Ln0	Selection of Input Clock	k
0	0	0	0	0	×	×	×	Setting prohibited	–
0	0	0	0	1	0	0	0	f_{SCK}/8	8
0	0	0	0	1	0	0	1	f_{SCK}/9	9
0	0	0	0	1	0	1	0	f_{SCK}/10	10
0	0	0	0	1	0	1	1	f_{SCK}/11	11
0	0	0	0	1	1	0	0	f_{SCK}/12	12
0	0	0	0	1	1	0	1	f_{SCK}/13	13
0	0	0	0	1	1	1	0	f_{SCK}/14	14
0	0	0	0	1	1	1	1	f_{SCK}/15	15
0	0	0	1	0	0	0	0	f_{SCK}/16	16
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	f_{SCK}/255	255

Cautions 1. The value of BRGCn becomes 00H after reset. Before starting operation, select a setting other than "Setting prohibited". Selecting the "Setting prohibited" setting in stop mode does not cause any problems.

2. If write is performed to BRGCn during communication processing, the output of the baud rate generator will be disturbed and communication will not be performed normally. Therefore, do not write to BRGCn during communication processing.

Remark f_{SCK}: Source clock of 8-bit counter

×: Don't care

(5) 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, 1)

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

Figure 10-36. Baud Rate Generator Mode Control Registers n0, n1 (BRGMCn0, BRGMCn1)

After reset: 00H R/W Address: FFFFF30EH, FFFFF31EH

	7	6	5	4	3	2	1	0
BRGMCn0 (n = 0, 1)	0	0	0	0	0	TPSn2	TPSn1	TPSn0

After reset: 00H R/W Address: FFFFF320H, FFFFF322H

	7	6	5	4	3	2	1	0
BRGMCn1 (n = 0, 1)	0	0	0	0	0	0	0	TPSn3

TPSn3	TPSn2	TPSn1	TPSn0	8-Bit Counter Source Clock Selection	m
0	0	0	0	External clock (ASCKn)	–
0	0	0	1	f_{XX}	0
0	0	1	0	$f_{XX}/2$	1
0	0	1	1	$f_{XX}/4$	2
0	1	0	0	$f_{XX}/8$	3
0	1	0	1	$f_{XX}/16$	4
0	1	1	0	$f_{XX}/32$	5
0	1	1	1	at n = 0: TM3 output at n = 1: TM2 output	–
1	0	0	0	$f_{XX}/64$	6
1	0	0	1	$f_{XX}/128$	7
1	0	1	0	$f_{XX}/256$	8
1	0	1	1	$f_{XX}/512$	9
1	1	0	0	Setting prohibited	–
1	1	0	1		–
1	1	1	0		–
1	1	1	1		–

Caution If write is performed to BRGMCn0, BRGMCn1 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, BRGMCn1 during communication processing.

Remark f_{SCK}: Source clock of 8-bit counter

10.4.3 Operations

The UARTn has the following two operation modes.

- Operation stop mode
- Asynchronous serial interface (UARTn) mode

(1) Operation stop mode

This mode does not perform serial transfers and can therefore be used to reduce power consumption.
In operation stop mode, pins can be used as ordinary ports.

(a) Register settings

Operation stop mode settings are made via the asynchronous serial interface mode register n0 (ASIMn0).
ASIMn0 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets the ASIMn0 to 00H.

Figure 10-37. ASIMn0 Settings (Operation Mode)

After reset: 00H R/W Address: FFFFF300H, FFFFF310H

ASIMn0 (n = 0, 1)

7	6	5	4	3	2	1	0
TXEn	RXEn	PS1n	PS0n	CLn	SLn	ISRMn	0

TXEn	RXEn	Operation Mode	RXDn/Pxx Pin Function	TXDn/Pxx Pin Function
0	0	Operation stop	Port function	Port function
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

Caution Do not switch the operation mode until after the current serial transmit/receive operation has stopped.

10.4.4 Asynchronous serial interface (UARTn) mode

This mode enables full-duplex operation which transmits and receives one byte of data after the start bit.

The on-chip dedicated UARTn baud rate generator enables communications using a wide range of selectable baud rates.

The UARTn baud rate generator can also be used to generate a MIDI-standard baud rate (31.25 kbps).

(1) Register settings

UARTn mode settings are made via the asynchronous serial interface mode register n0 (ASIMn0), asynchronous serial interface mode register n1 (ASIMn1), asynchronous serial interface status register n (ASISn), baud rate generator control register n (BRGCn), and the baud rate generator mode control registers n0 and n1 (BRGMCn0 and BRGMCn1).

Refer to **10.4.1 Configuration**.

(2) 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>

$$[\text{Baud rate}] = \frac{f_{XX}}{2^{m+1} \times k} \text{ [Hz]}$$

f_{XX}: Main clock oscillation frequency
m: Value set by TPSn3 to TPSn0 ($0 \le m \le 9$)
k: Value set by MDLn7 to MDLn0 ($8 \le k \le 255$)

- **Baud rate tolerance**

The baud rate tolerance depends on the number of bits in a frame and the counter division ratio [1/(16 + k)].

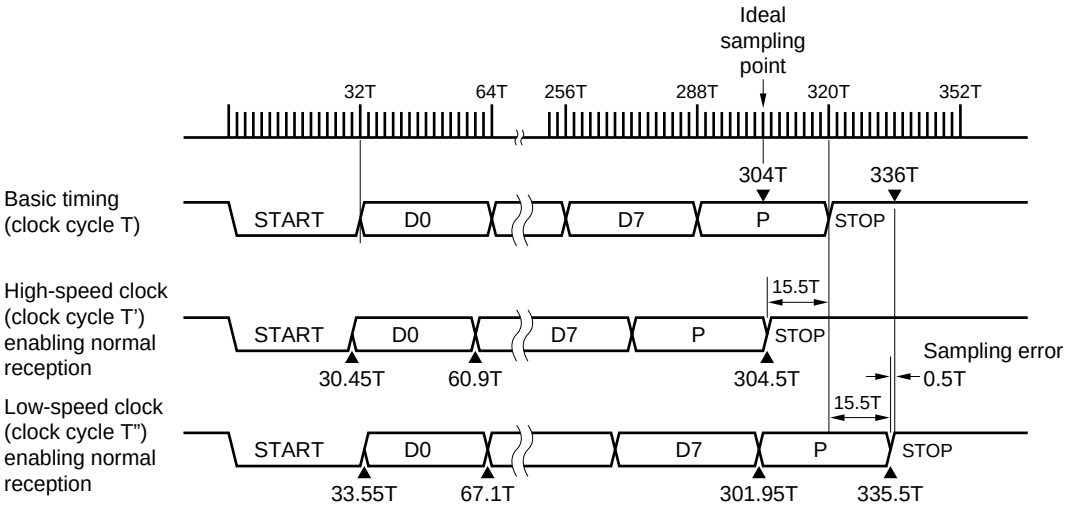
Table 10-8 shows the relationship between the main clock and the baud rate, and Figure 10-38 shows an example of the baud rate tolerance.

Table 10-8. Relationship between Main Clock and Baud Rate

Baud Rate (bps)	f_{XX} = 2 MHz			f_{XX} = 4.194 MHz			f_{XX} = 8.388 MHz			f_{XX} = 20 MHz		
	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)	k	m	Error (%)
8	255	9	–4.26	–	–	–	–	–	–	–	–	–
16	244	8	0.06	255	9	0.39	–	–	–	–	–	–
32	244	7	0.06	255	8	0.38	255	9	0.38	–	–	–
75	208	6	0.16	218	7	0.20	218	8	0.20	–	–	–
76	206	6	–0.20	216	7	–0.20	216	8	–0.20	255	9	0.78
256	244	4	0.06	128	6	0.01	128	7	0.01	152	8	0.39
1,200	208	2	0.16	217	3	0.20	217	4	0.20	130	6	0.16
2,400	208	1	0.16	218	2	0.20	218	3	0.20	130	5	0.16
4,800	208	0	0.16	218	1	0.20	218	2	0.20	130	4	0.16
9,600	104	0	0.16	218	0	0.20	218	1	0.20	130	3	0.16
19,200	52	0	0.16	109	0	0.20	218	0	0.20	130	2	0.16
31,250	32	0	0.00	67	0	0.16	134	0	0.16	160	1	0.00
38,400	26	0	0.16	55	0	–0.71	110	0	–0.71	130	1	0.16
76,800	13	0	0.16	27	0	1.13	54	0	1.13	130	0	0.16
125,000	8	0	0.00	17	0	–1.32	34	0	–1.32	80	0	0.00
150,000	–	–	–	14	0	0.14	28	0	0.14	67	0	–0.50
262,000	–	–	–	8	0	0.05	16	0	0.05	38	0	0.44
300,000	–	–	–	–	–	–	14	0	–0.14	33	0	1.01
524,000	–	–	–	–	–	–	8	0	0.05	19	0	0.44
1,250,000	–	–	–	–	–	–	–	–	–	8	0	0.00

Remark f_{XX}: Main clock oscillation frequency

Figure 10-38. Error Tolerance Including Sampling Errors (When k = 16)



Remark T: 8-bit counter's source clock cycle

$$\text{Baud rate error tolerance (when k = 16)} = \frac{\pm 15.5}{320} \times 100 = 4.8438\ (\%)$$

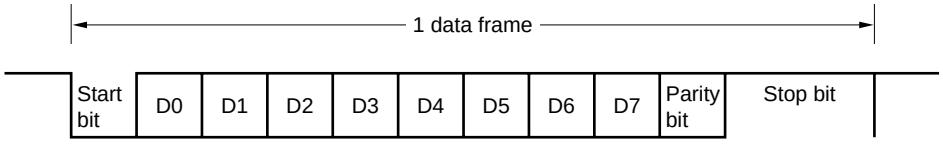
(3) Communication operations

(a) Data format

As shown in Figure 10-39, the format of the transmit/receive data consists of a start bit, character bits, a parity bit, and one or more stop bits.

The asynchronous serial interface mode register n0 (ASIMn0) is used to set the character bit length, parity selection, and stop bit length within each data frame (n = 0, 1).

Figure 10-39. 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 low-order 7 bits (from bit 0 to bit 6) are valid, so that during a transmission the highest bit (bit 7) is ignored and during reception the highest bit (bit 7) must be set to 0.

The asynchronous serial interface mode register n0 (ASIMn0) and the baud rate generator control register n (BRGCn) are used to set the serial transfer rate (n = 0, 1).

If a receive error occurs, information about the receive error can be recognized by reading the asynchronous serial interface status register n (ASISn).

(b) Parity types and operations

The parity bit is used to detect bit errors in transfer data. Usually, the same type of parity bit is used by the transmitting and receiving sides. When odd parity or even parity is set, errors in the parity bit (the odd-number bit) can be detected. When zero parity or no parity is set, errors are not detected.

(i) Even parity

- **During transmission**

  The number of bits in transmit data including a parity bit is controlled so that the number is set an even number of “1” bits. The value of the parity bit is as follows.

  If the transmit data contains an odd number of “1” bits : the parity bit value is “1”
  If the transmit data contains an even number of “1” bits : the parity bit value is “0”

- **During reception**

  The number of “1” bits is counted among the receive data including a parity bit, and a parity error 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 is set an odd number of “1” bits. The value of the parity bit is as follows.

  If the transmit data contains an odd number of “1” bits : the parity bit value is “0”
  If the transmit data contains an even number of “1” bits : the parity bit value is “1”

- **During reception**

  The number of “1” bits is counted among the receive data including a parity bit, and a parity error 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

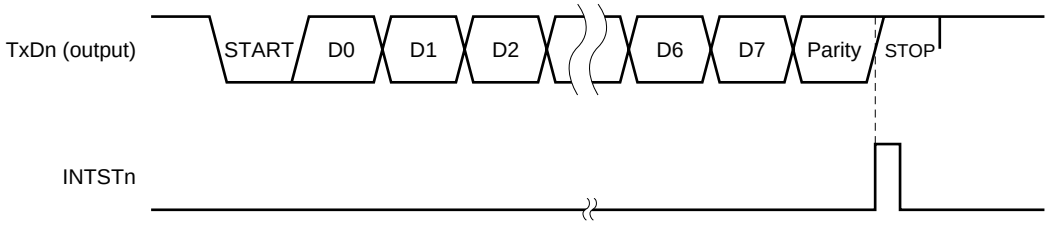
The transmit operation is started when transmit data is written to the transmit shift register (TXSn). A start bit, parity bit, and stop bit(s) are automatically added to the data.

Starting the transmit operation shifts out the data in TXSn, thereby emptying TXSn, after which a transmit completion interrupt (INTSTn) is issued.

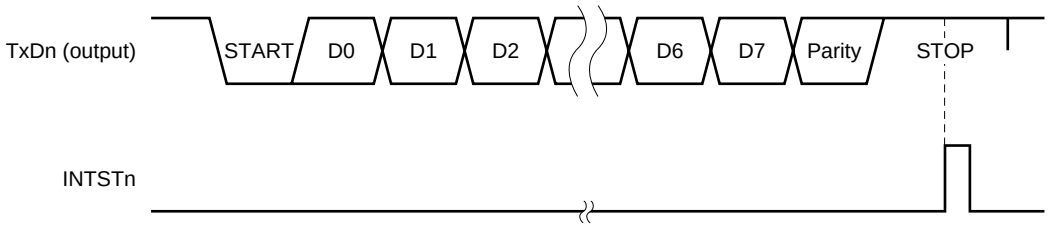
The timing of the transmit completion interrupt is shown in Figure 10-40.

Figure 10-40. Timing of Asynchronous Serial Interface Transmit Completion Interrupt

(a) Stop bit length: 1



(b) Stop bit length: 2



Caution Do not write to the asynchronous serial interface mode register n0 (ASIMn0) during a transmit operation. Writing to ASIMn0 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 transmit completion interrupt (INTSTn) or the interrupt request flag (STIFn) that is set by INTSTn.

Remark n = 0, 1

(d) Reception

The receive operation is enabled when “1” is set to bit 6 (RXEn) of the asynchronous serial interface mode register n0 (ASIMn0), and input via the RXDn pin is sampled.

The serial clock specified by ASIMn0 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 the receive buffer register (RXBn) and a receive completion interrupt request (INTSRn) occurs.

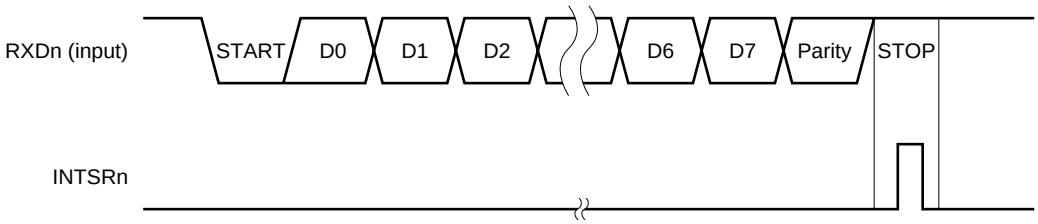
Even if an error has occurred, the receive data in which the error occurred is still transferred to RXBn.

When an error occurs, INSTRn is generated if bit 1 (ISRMn) of ASIMn0 is cleared (0). On the other hand, INTSRn is not generated if the ISRMn bit is set (1) (refer to **Figure 10-32**).

If the RXEn bit is reset to 0 during a receive operation, the receive operation is stopped immediately. At this time, the contents of RXBn and ASISn do not change, nor does INTSRn or INTSERn occur.

Figure 10-41 shows the timing of the asynchronous serial interface receive completion interrupt.

Figure 10-41. Timing of Asynchronous Serial Interface Receive Completion Interrupt



Caution Be sure to read the contents of the receive buffer register (RXBn) even when a receive error has occurred. If the contents of RXBn are not read, an overrun error will occur during the next data receive operation and the receive error status will remain.

Remark n = 0, 1

(e) Receive error

There are three types of error during the receive operation, pariy error, framing error, and overrun error. When, as the result of data receive, an error flag is set in the asynchronous serial interface status register n (ASISn), the receive error interrupt request (INTSERn) is generated. The receive error interrupt request is generated prior to the receive completion interrupt request (INTSRn). Table 10-9 shows receive error causes.

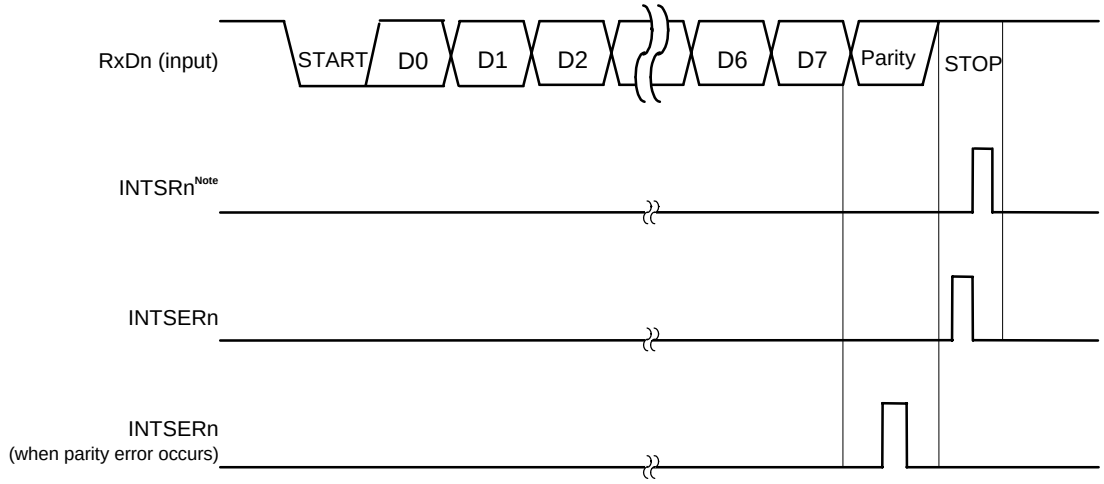
By reading the contents of ASISn during receive error interrupt servicing (INTSERn), it is possible to detect which error has occurred at reception (refer to **Table 10-9** and **Figure 10-42**).

The contents of ASISn are reset (0) by reading the receive buffer register (RXBn) or receiving subsequent data (if there is an error in the subsequent data, the error flag is set).

Table 10-9. Receive Error Causes

Receive Error	Cause	ASISn Value
Parity error	Parity specification at transmission and receive data parity do not match.	04H
Framing error	Stop bit is not detected.	02H
Overrun error	Reception of subsequent data was completed before data was read from the receive buffer register.	01H

Figure 10-42. Receive Error Timing



Note Even if a receive error occurs when the ISRMn bit of ASIMn0 is set (1), INTSRn is not generated.

Cautions 1. The contents of the asynchronous serial interface status register n (ASISn) are reset (0) by reading the receive buffer register (RXBn) or receiving subsequent data. To check the contents of error, always read ASISn before reading RXBn.

2. Be sure to read the receive buffer register (RXBn) even in receive error generation. If RXBn is not read out, an overrun error will occur during subsequent data reception and as a result receive errors will continue to occur.

Remark n = 0, 1

10.4.5 Standby function

(1) Operation in HALT mode

Serial transfer operation 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 n0 (ASIMn0), transmit shift register n (TXSn), and receive buffer register n (RXBn) are stopped and their values immediately before the clock stopped are hold.

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 are stored and subsequent receive operation is stopped. Reception resumes upon clock restart.

(b) When external clock is selected as serial clock

Serial transfer operation is performed normally.

10.5 3-Wire Variable Length Serial I/O (CSI4)

The CSI4 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed.

(2) 3-wire variable length serial I/O mode (MSB/LSB first switchable)

This mode transfers variable data of 8 to 16 bits via three lines: serial clock ($\overline{SCK4}$), serial output (SO4), and serial input (SI4).

Since the data can be transmitted and received simultaneously in the 3-wire variable length serial I/O mode, the processing time of data transfer is shortened.

MSB and LSB can be switched for the first bit of data to be transferred in serial.

The 3-wire variable length serial I/O mode is useful when connecting to a peripheral I/O device that includes a clock-synchronous serial interface, a display controller, etc.

Caution The serial clock ($\overline{SCK4}$) must be used within the range of $\overline{SCK4} \leq$ 2.5 MHz.

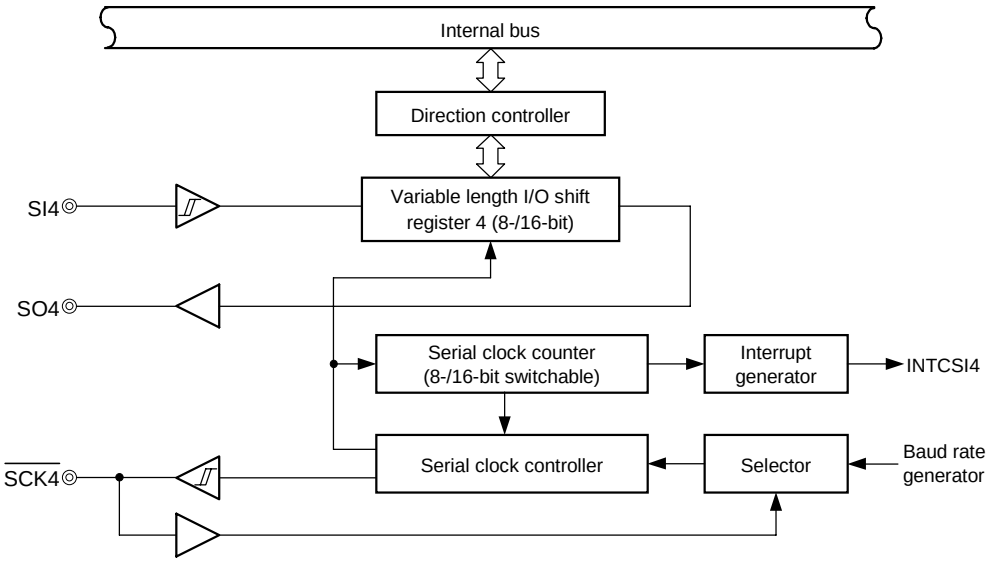
10.5.1 Configuration

The CSI4 includes the following hardware.

Table 10-10. Configuration of CSI4

Item	Configuration
Register	Variable length serial I/O shift register 4 (SIO4)
Control registers	Variable length serial control register 4 (CSIM4) Variable length serial setting register 4 (CSIB4) Baud rate generator source clock select register 4 (BRGCN4) Baud rate output clock select register 4 (BRGCK4)

Figure 10-43. Block Diagram of CSI4



(1) Variable length serial I/O shift register 4 (SIO4)

SIO4 is the 16-bit variable register that performs parallel-serial conversion and transmit/receive (shift operations) synchronized with the serial clock.

SIO4 is set by a 16-bit memory manipulation instruction.

The serial operation starts when data is written to or read from SIO4, while the bit 7 (CSIE4) of variable length serial control register 4 (CSIM4) is 1.

When transmitting, data written to SIO4 is output via the serial output (SO4).

When receiving, data is read from the serial input (SI4) and written to SIO4.

$\overline{\text{RESET}}$ input resets SIO4 to 0000H.

Caution Do not execute SIO4 access except for the accesses that become transfer start trigger during transfer operation (read is disabled when MODE4 = 0 and write is disabled when MODE4 = 1).

Figure 10-44. Variable Length Serial I/O Shift Register 4 (SIO4)

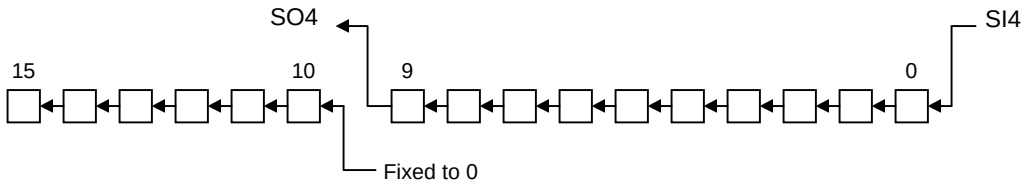
After reset: 0000H R/W Address: FFFFF2E0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIO4																

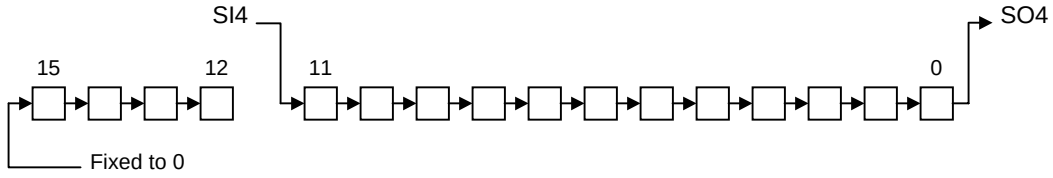
When transfer bit length is set other than 16 bits no matter which data should be aligned from the lowest bit of the shift register for setting data, regardless of whether MSB or LSB is set for the transfer first bit. Any data can be set to the unused high-order bits, however, in this case the received data after a serial transfer operation becomes 0.

Figure 10-45. When Transfer Bit Length Other Than 16 Bits Is Set

(a) When transfer bit length is 10 bits and MSB first



(b) When transfer bit length is 12 bits and LSB first



10.5.2 CSI4 control registers

The CSI4 uses the following type of registers for control functions.

- 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 output clock select register 4 (BRGCK4)

(1) Variable length serial control register 4 (CSIM4)

This register is used to enable or disable serial interface channel 4's serial clock, operation modes, and specific operations.

CSIM4 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIM4 to 00H.

Figure 10-46. Variable Length Serial Control Register 4 (CSIM4)

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 disable	Clear	Port function[Note 1]
1	Operation enable	Count operation enable	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	Serial Clock Selection
0	External clock input ($\overline{\text{SCK4}}$)
1	BRG (Baud rate generator)

Notes 1. When CSIE4 = 0 (SIO4 operation disable status), the port function is available for the SI4, SO4, and $\overline{\text{SCK4}}$ pins.

2. When CSIE4 = 1 (SIO4 operation enable status), the port function is available for the SI4 pin when only the transmit function is used and for the SO4 pin when the receive function is used.

(2) Variable length serial setting register 4 (CSIB4)

CSIB4 is used to set the operation format of serial interface channel 4.

Bit length of transmission data is set by setting bits 3 to 0 (BSEL3 to BSEL0) of this register. Data is transferred in MSB first while bit 4 (DIR) is 1, and is transferred in LSB first while DIR is 0.

CSIB4 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets CSIB4 to 00H.

Figure 10-47. Variable Length Serial Setting Register 4 (CSIB4)

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 Fetch 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 Transmission Data
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 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGCN4 to 00H.

Figure 10-48. Baud Rate Generator Source Clock Select Register 4 (BRGCN4)

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 output clock select register 4 (BRGCK4)

BRGCK4 is set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGCK4 to 7FH.

Figure 10-49. Baud Rate Output Clock Select Register 4 (BRGCK4)

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	0	0	0	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 Use the baud rate transmit/receive clock of variable length serial I/O (CSI4) at a transfer rate not exceeding the operating frequency of the CPU. Otherwise, correct data cannot be transferred.

10.5.3 Operations

The CSI4 has the following two operation modes.

(1) Operation stop mode

This mode does not perform serial transfers and can therefore reduce power consumption.

When in operation stop mode, SI4, SO4, and $\overline{\text{SCK4}}$ can be used as normal I/O ports.

(a) Register settings

Operation stop mode is set via variable length serial control register 4 (CSIM4). While CSIE4 = 0 (SIO4 operation stop state), the pins connected to SI4, SO4, or $\overline{\text{SCK4}}$ function as port pins.

(2) 3-wire variable length serial I/O mode

The 3-wire variable length serial I/O mode is useful when connecting to a peripheral I/O device that includes a clock-synchronous serial interface, a display controller, etc.

This mode executes data transfers via three lines: a serial clock line ($\overline{\text{SCK4}}$), serial output line (SO4), and serial input line (SI4).

(a) Register settings

The 3-wire variable length serial I/O mode is set by variable length serial control register 4 (CSIM4) and variable length serial setting register 4 (CSIB4). For details, refer to **10.5.2 (1) Variable length serial control register 4 (CSIM4)** and **10.5.2 (2) Variable length serial setting register 4 (CSIB4)**.

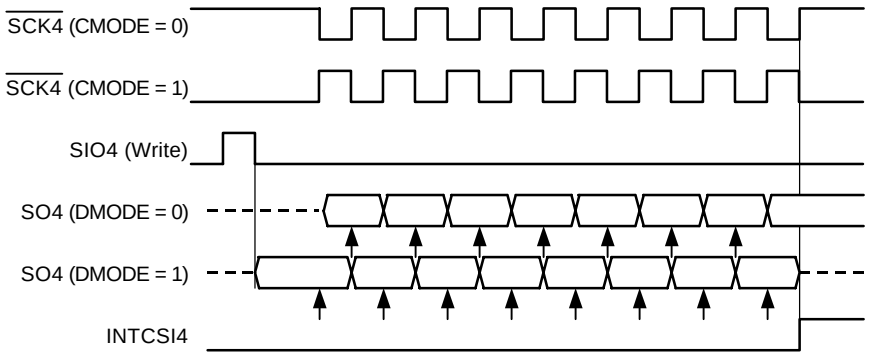
(b) Communication operations

In the 3-wire variable length serial I/O mode, data is transmitted and received in 8- to 16-bit units, and is specified by setting bits 3 to 0 (BSEL3 to BSEL0) of variable length serial setting register 4 (CSIB4). Each bit of data is transmitted or received in synchronization with the serial clock.

After transfer of all bits is completed, 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 change the attribute of the serial clock ($\overline{SCK4}$) and the phases of serial data (SI4 and SO4).

Figure 10-50. Timing of 3-Wire Variable Length Serial I/O Mode



Remark An arrow shows the SI4 data fetch timing.

When CMODE = 0, the serial clock ($\overline{SCK4}$) stops at the high level during the operation stop, and outputs the low level during a data transfer operation. When CMODE = 1, on the other hand, $\overline{SCK4}$ stops at the low level during the operation stop and outputs the high level during a data transfer operation.

The phases of the SO4 output timing and the SI4 fetch timing can transit half a clock by setting DMODE.

However, the interrupt signal (INTCSI4) is generated at the final edge of the serial clock ($\overline{SCK4}$), regardless the setting of CSIB4.

(c) Transfer start

A serial transfer becomes possible when the following two conditions have been satisfied.

- The SIO4 operation control bit (CSIE4) = 1
- After a serial transfer, the internal serial clock is stopped.

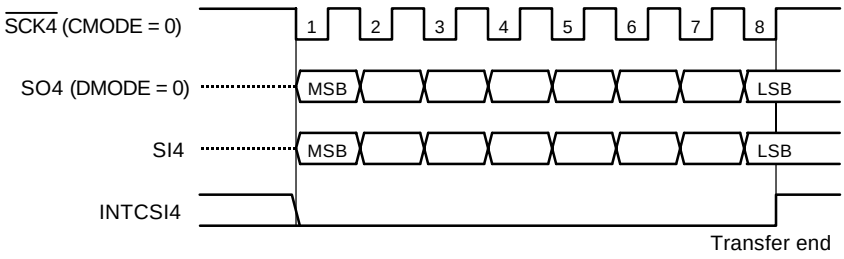
Serial transfer starts when the following operation is performed after the above two conditions have been satisfied.

- Transmit/transmit and receive mode (MODE4 = 0)
  Transfer starts when writing to SIO4.
- Receive-only mode (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 10-51. Timing of 3-Wire Variable Length Serial I/O Mode (When CSIB4 = 08H)



Remark CSIB4 = 08H (CMODE = 0, DMODE = 0, DIR = 0, BSEL3 to BSEL0 = 1,000)

[MEMO]

CHAPTER 11 A/D CONVERTER

11.1 Outline

- Analog input: 16 channels
- 10-bit A/D converter
- On-chip A/D conversion result register (ADCR0 to ADCR7)
  10 bits × 8
- A/D conversion trigger mode
  A/D trigger mode
  Timer trigger mode
  External trigger mode
- Sequential conversion

11.2 Configuration

The A/D converter of the V850/SV1 adopts the sequential conversion method, and uses the A/D converter mode registers 0 and 1 (ADM0, ADM1), and ADCRn register to perform A/D conversion operations (n = 0 to 7).

Table 11-1. Configuration of A/D Converter

Item	Configuration
Analog input	16 channels (ANI0 to ANI15)
Registers	Successive approximation register (SAR) A/D conversion result registers 0 to 7 (ADCR0 to ADCR7)
Control registers	A/D converter mode register 0 (ADM0) A/D converter mode register 1 (ADM1)

(1) Input circuit

Selects the analog input (ANI0 to ANI15) according to the mode set to the ADM0 and ADM1 registers and then sends it to the sample and hold circuit.

(2) Sample and hold circuit

Samples the analog input sent from the input circuit one by one and sends it to the comparator. During A/D conversion operations, holds the analog input sampled.

(3) Voltage comparator

Compares the voltage difference between the input analog input and voltage tap of the serial resistor string output.

(4) Serial resistor string

Generates voltage to coincide with analog input.

The serial resistor string is connected between the reference voltage pin for A/D converter (AV_{REF}) and GND pin for A/D converter (AV_{SS}). The serial resistor consists of 255 equivalent resistors and two resistors with half the resistance so that the connection between the two pins is made of 256 equal voltage steps.

The voltage tap of the serial resistor string is selected by a tap selector controlled by the successive approximation register (SAR).

(5) Successive approximation register (SAR)

10-bit register for setting data for which the value of the voltage tap of the serial resistor string coincides with the voltage value of the analog input from the most significant bit (MSB) in 1-bit units.

When the setting is made to the least significant bit (LSB) (A/D conversion ends), the contents of the SAR (conversion result) are held in the A/D conversion result register (ADCRn).

(6) A/D conversion result registers 0 to 7 (ADCRn)

10-bit register for retaining the A/D conversion result. The conversion result is loaded from the successive approximation register (SAR) each time A/D conversion ends.

This register becomes undefined by $\overline{\text{RESET}}$ input.

(7) Controller

Selects the analog input, generates the sample & hold circuit operation timing, and controls the conversion trigger according to the mode set to the ADM0/ADM1 register.

(8) ANI0 to ANI15 pins

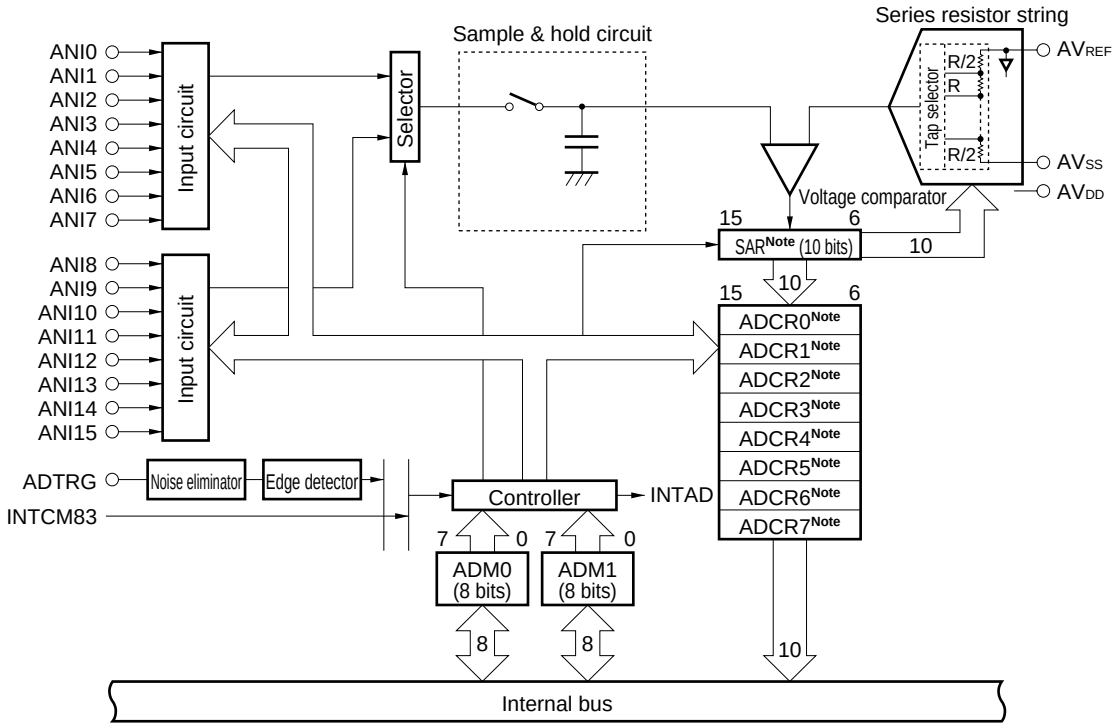
16-channel analog input pins for the A/D converter. Input the analog signal to be A/D converted.

Caution Use ANI0 to ANI15 input voltages within the specification. Especially, if the voltage exceeding AV_{DD} or less than AV_{SS} (even if it is within the range of the absolute maximum rating) is input, the conversion value of the channel may become undefined or the conversion value of other channels may be affected.

(9) AV_{REF} pin

Pin for inputting the reference voltage of the A/D converter. Converts signals input to the ANI0 to ANI15 pins to digital signals based on the voltage applied between AV_{REF} and AV_{SS}.

Figure 11-1. Block Diagram of A/D Converter



Note The low-order 6 bits of SAR and ADCR0 to ADCR7 are fixed to 0.

11.3 Control Registers

The A/D converter is controlled by the following registers:

- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)

11.3.1 A/D converter mode register 0 (ADM0)

ADM0 is an 8-bit register for selecting an analog input pin, specifying an operation mode, and controlling the conversion operation.

The register can be read or written in 1-bit or 8-bit units. If the ADM0 register is written during A/D conversion, the conversion operation is initialized and performed again from the beginning. Bit 6 cannot be written. Any attempt to write the bit is ignored.

Figure 11-2. A/D Converter Mode Register 0 (ADM0) (1/2)

After reset: 00H R/W Address: FFFFF328H

	7	6	5	4	3	2	1	0
ADM0	CE	CS[Note]	BS	MS	PS	ANIS2	ANIS1	ANIS0

CE	A/D Conversion Operation Enable/Disable Specification
0	Disable
1	Enable

CS[Note]	A/D Converter Status Display
0	Stop
1	Operating

BS	Buffer Mode Specification in Select Mode
0	1-buffer mode
1	4-buffer mode

MS	A/D Converter Operation Mode Specification
0	Scan mode
1	Select mode

Note CS is a read-only bit.

Figure 11-2. A/D Converter Mode Register 0 (ADM0) (2/2)

After reset: 00H R/W Address: FFFFF328H

	7	6	5	4	3	2	1	0
ADM0	CE	CS	BS	MS	PS	ANIS2	ANIS1	ANIS0

PS	Analog Input Pin Selection
0	ANI0 to ANI7 are selected.
1	ANI8 to ANI15 are selected.

ANIS2	ANIS1	ANIS0	Select Mode		Scan Mode	
			PS = 0	PS = 1	PS = 0	PS = 1
0	0	0	ANI0	ANI8	ANI0	ANI8
0	0	1	ANI1	ANI9	ANI0, ANI1	ANI8, ANI9
0	1	0	ANI2	ANI10	ANI0 to ANI2	ANI8 to ANI10
0	1	1	ANI3	ANI11	ANI0 to ANI3	ANI8 to ANI11
1	0	0	ANI4	ANI12	ANI0 to ANI4	ANI8 to ANI12
1	0	1	ANI5	ANI13	ANI0 to ANI5	ANI8 to ANI13
1	1	0	ANI6	ANI14	ANI0 to ANI6	ANI8 to ANI14
1	1	1	ANI7	ANI15	ANI0 to ANI7	ANI8 to ANI15

Caution If the CE bit is set to 1 in timer trigger mode or external trigger mode, the converter enters the trigger signal wait status. To clear the CE bit, write 0 or issue a reset.
In A/D trigger mode, writing 1 in the CE bit triggers a conversion. If the mode is changed to timer mode or external trigger mode without clearing the CE bit after the operation, the converter enters the trigger input wait status immediately after the change.

11.3.2 A/D converter mode register 1 (ADM1)

ADM1 is an 8-bit register for specifying the conversion operation time and trigger mode.

The register can be read and written in 1-bit or 8-bit units. If the ADM0 register is written during A/D conversion, the conversion operation is initialized and performed all over again from the beginning.

Figure 11-3. A/D Converter Mode Register 1 (ADM1) (1/2)

After reset: 00H R/W Address: FFFFF32AH

	7	6	5	4	3	2	1	0
ADM1	EGA1	EGA0	TRG1	TRG0	ADPS	FR2	FR1	FR0

EGA1	EGA0	Valid Edge Specification for External Trigger Signal ADTRG
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Falling and rising edge detection

TRG1	TRG0	Trigger Mode Specification
0	0	A/D trigger mode
0	1	Timer trigger mode
1	0	External trigger mode
1	1	Setting prohibited

Figure 11-3. A/D Converter Mode Register 1 (ADM1) (2/2)

After reset: 00H R/W Address: FFFFF32AH

	7	6	5	4	3	2	1	0
ADM1	EGA1	EGA0	TRG1	TRG0	ADPS	FR2	FR1	FR0

ADPS	FR2	FR1	FR0	Conversion Time[Note 1] + Stabilization Time[Note 2] (ADPS = 1)	Conversion Operation Time		
					f_{XX} = 20 MHz	f_{XX} = 10 MHz	f_{XX} = 4 MHz
0	0	0	0	288/f_{XX}	14.4 μs	28.8 μs	72.0 μs
0	0	0	1	216/f_{XX}	10.8 μs	21.6 μs	54.0 μs
0	0	1	0	168/f_{XX}	8.4 μs	16.8 μs	42.0 μs
0	0	1	1	120/f_{XX}	6.0 μs	12.0 μs	30.0 μs
0	1	0	0	108/f_{XX}	5.4 μs	10.8 μs	27.0 μs
0	1	0	1	84/f_{XX}	Setting prohibited	8.4 μs	21.0 μs
0	1	1	0	60/f_{XX}	Setting prohibited	6.0 μs	15.0 μs
0	1	1	1	48/f_{XX}	Setting prohibited	Setting prohibited	12.0 μs
1	0	0	0	288/f_{XX} + 144/f_{XX}	14.4 + 7.2 μs	28.8 + 14.4 μs	72.0 + 36.0 μs
1	0	0	1	216/f_{XX} + 108/f_{XX}	10.8 + 5.4 μs	21.6 + 10.8 μs	54.0 + 27.0 μs
1	0	1	0	168/f_{XX} + 84/f_{XX}	8.4 + 4.2 μs	16.8 + 8.4 μs	42.0 + 21.0 μs
1	0	1	1	120/f_{XX} + 60/f_{XX}	6.0 + 3.0 μs	12.0 + 6.0 μs	30.0 + 15.0 μs
1	1	0	0	108/f_{XX} + 54/f_{XX}	5.4 + 2.7 μs	10.8 + 5.4 μs	27.0 + 13.5 μs
1	1	0	1	84/f_{XX} + 42/f_{XX}	Setting prohibited	8.4 + 4.2 μs	21.0 + 10.5 μs
1	1	1	0	60/f_{XX} + 30/f_{XX}	Setting prohibited	6.0 + 3.0 μs	15.0 + 12.5 μs
1	1	1	1	48/f_{XX} + 24/f_{XX}	Setting prohibited	Setting prohibited	12.0 + 6.0 μs

Notes 1. Conversion time: Actual A/D conversion time. Set a conversion time that satisfies 5 μs ≤ Conversion time ≤ 100 μs.

2. A/D conversion setup time. A single A/D conversion requires the conversion time plus stabilization time. If ADPS is set to 0, the stabilization time is 0.

Remark f_{XX}: System clock

11.3.3 A/D conversion result registers 0 to 7 (ADCR0 to ADCR7)

ADCRn is a 10-bit register for holding the result of an A/D conversion. Eight 10-bit registers are provided. Only 16-bit read access is allowed. The low-order 6 bits are fixed to 0.

Figure 11-4. A/D Conversion Result Registers 0 to 7 (ADCR0 to ADCR7)

After reset: 0000H R Address: FFFFF330H FFFFF332H FFFFF334H FFFFF336H
FFFFF338H FFFFF33AH FFFFF33CH FFFFF33EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn	ADCRn9 to ADCRn0										0[Note]					

Note Low-order 6 bits are fixed to 0.

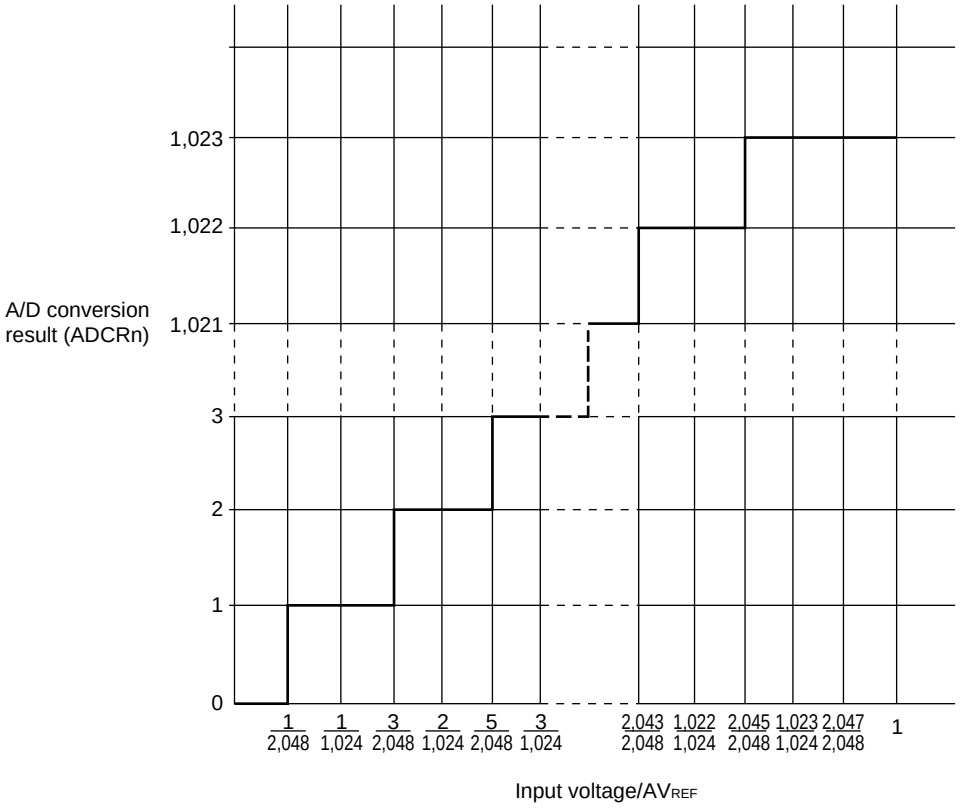
Remark n = 0 to 7

The following shows the correspondence of each analog input pin to the ADCRn register (except 4-buffer mode).

Analog Input Pin		ADCRn Register
PS = 0	PS = 1	
ANI0	ANI8	ADCR0
ANI1	ANI9	ADCR1
ANI2	ANI10	ADCR2
ANI3	ANI11	ADCR3
ANI4	ANI12	ADCR4
ANI5	ANI13	ADCR5
ANI6	ANI14	ADCR6
ANI7	ANI15	ADCR7

Figure 11-5 shows the relation between the analog input voltage and the A/D conversion result.

Figure 11-5. Relation between Analog Input Voltage and A/D Conversion Result



11.4 Operation

11.4.1 Basic operation

A/D conversion is executed in the following order.

(1) The selection of the analog input and specification of the operation mode and trigger mode, etc., should be set in the ADMn register[Note 1] (n = 0, 1).
When the CE bit of the ADM0 register is set (1), A/D conversion starts during the A/D trigger mode. During the timer trigger mode and external trigger mode, the trigger standby state[Note 2] is set.

(2) The voltage generated from the voltage tap of the serial resistor string and analog input are compared by the comparator.

(3) When the comparison of the 10 bits ends, the conversion results are stored in the ADCRn register. When A/D conversion is performed for the specified number of times, the A/D conversion end interrupt (INTAD) is generated (n = 0 to 7).

Notes 1. When the ADMn register (n = 0, 1) is changed during A/D conversion, the A/D conversion operation started before the change is stopped and the conversion results are not stored in the ADCRn register (n = 0 to 7).

2. During the timer trigger mode and external trigger mode, if the CE bit of the ADM0 register is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal, and the trigger standby state is returned when the A/D conversion operation ends.

11.4.2 Operation mode and trigger mode

The A/D converter can specify various conversion operations by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADMn register (n = 0, 1).

The following shows the relation between the operation mode and trigger mode.

Trigger Mode	Operation Mode		Setting Value		Analog Input
			ADM0	ADM1	
A/D trigger	Select	1 buffer	××01××××B	00000×××B	ANI0 to ANI15
		4 buffers	××11××××B	00000×××B	
	Scan		×××0××××B	00000×××B	
Timer trigger	Select	1 buffer	××01××××B	00010×××B	
		4 buffers	××11××××B	00010×××B	
	Scan		×××0××××B	00010×××B	
External trigger	Select	1 buffer	××01××××B	00100×××B	
		4 buffers	××11××××B	00100×××B	
	Scan		×××0××××B	00100×××B	

(1) Trigger mode

There are three types of trigger modes which serve as the start timing of A/D conversion processing: A/D trigger mode, timer trigger mode, and external trigger mode. These trigger modes are set by the ADM0 register.

(a) A/D trigger mode

Generates the conversion timing of the analog input for the ANI0 to ANI15 pins inside the A/D converter unit.

(b) Timer trigger mode

Specifies the conversion timing of the analog input set for the ANI0 to ANI15 pins using the values set to the RPU compare register.

This register creates the analog input conversion timing by generating the coincidence interrupts of the capture/compare registers (CC83) connected to the 24-bit TM8.

(c) External trigger mode

Mode which specifies the conversion timing of the analog input to the ANI0 to ANI15 pins using the ADTRG pin.

(2) Operation mode

There are two types of operation modes which set the ANI0 to ANI15 pins: select mode and scan mode. The select mode has sub-modes including the one-buffer mode and four-buffer mode. These modes are set by the ADM0 register.

(a) Select mode

A/D converts one analog input specified by the ADM0 register. The conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7). For this mode, the one-buffer mode and four-buffer mode are provided for storing the A/D conversion results.

- **One-buffer mode**

  A/D converts one analog input specified by the ADM0 register. The conversion results are stored in the ADCRn register corresponding to the analog input. The analog input and ADCRn register correspond one to one, and an A/D conversion end interrupt (INTAD) is generated each time one A/D conversion ends.

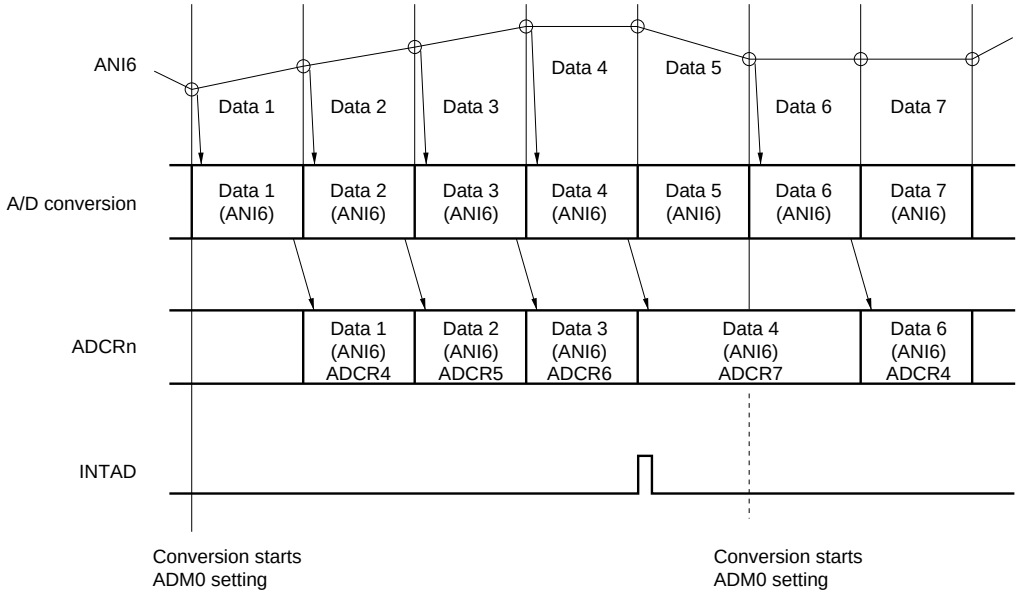
Figure 11-6. Operation Timing Example of Select Mode: 1-Buffer Mode (ANI1)

- **Four-buffer mode**

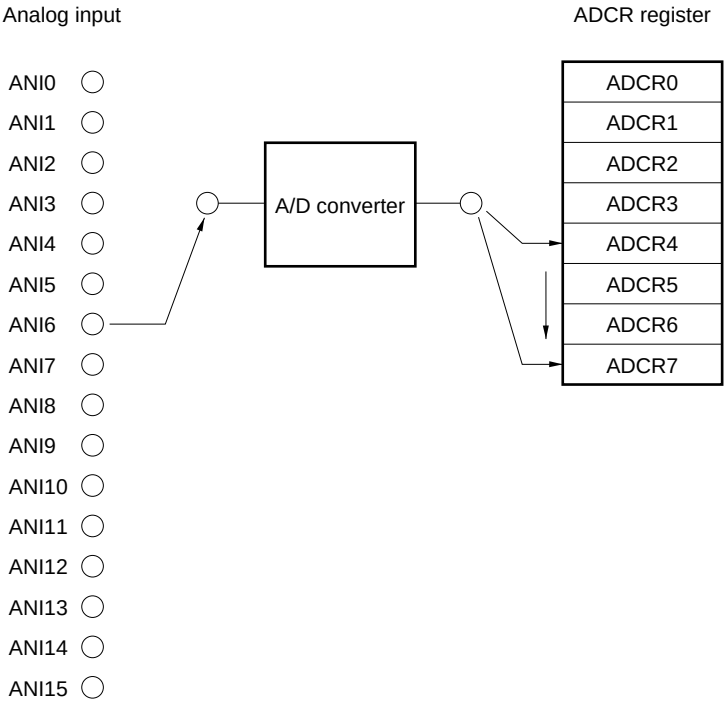
  A/D converts one analog input four times and stores the results in the four ADCRn registers corresponding to analog input. The A/D conversion end interrupt (INTAD) is generated when the four A/D conversions end.

Figure 11-7. Operation Timing Example of Select Mode: 4-Buffer Mode (ANI6)

(a) Timing example



(b) Block diagram

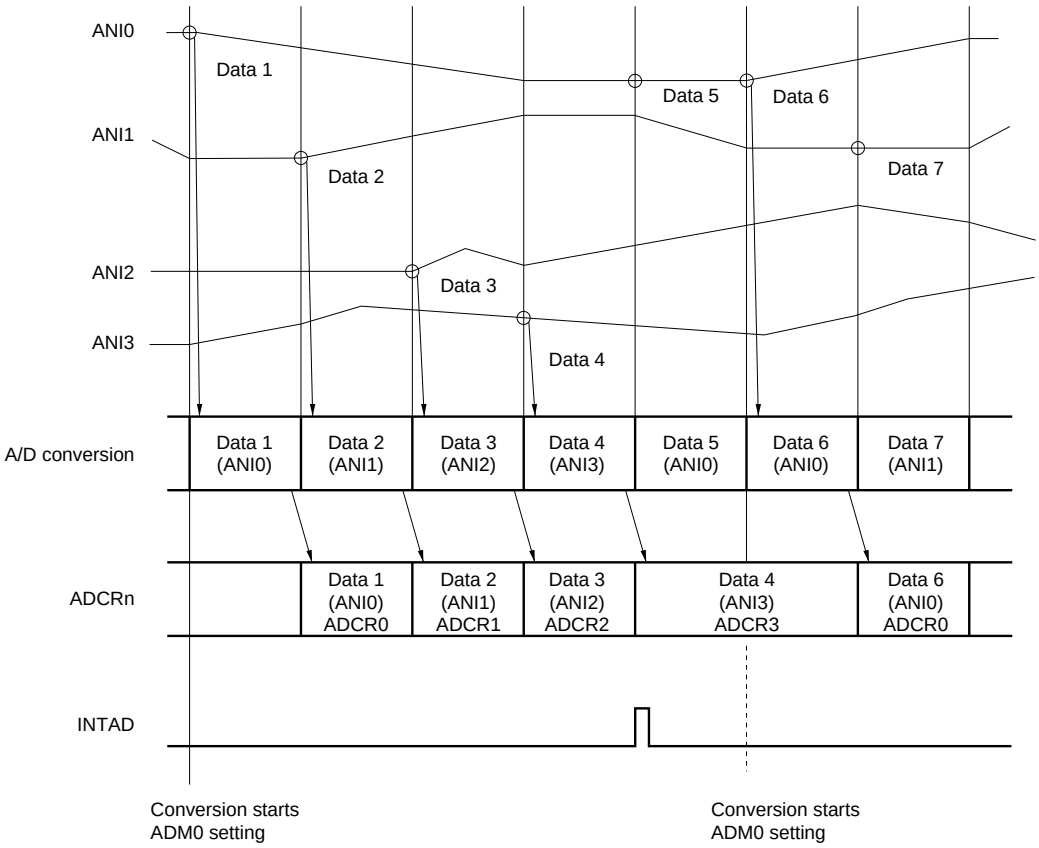


(b) Scan mode

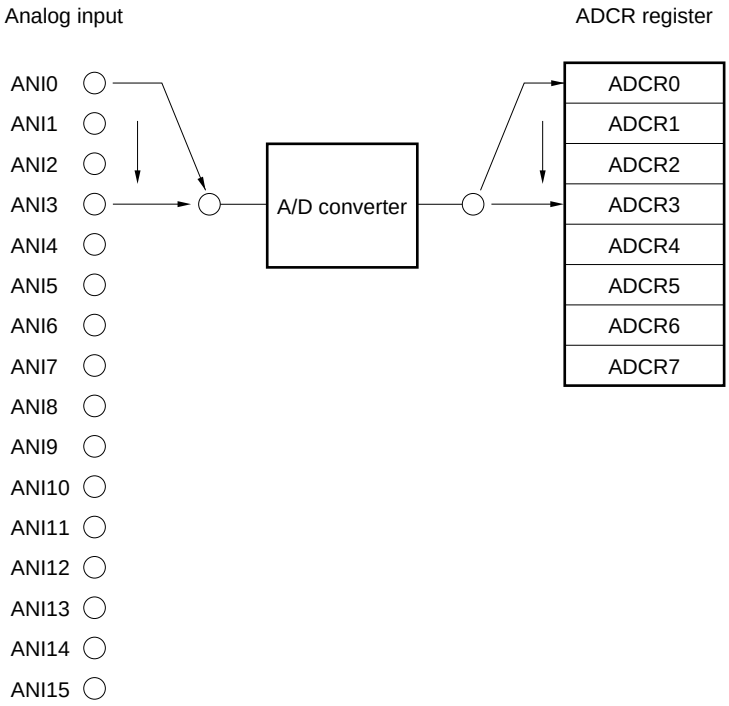
Selects the analog inputs specified by the ADM0 register sequentially from the ANI0 or ANI8 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input. When the conversion of the specified analog input ends, the INTAD interrupt is generated (n = 0 to 7).

Figure 11-8. Operation Timing Example of Scan Mode: 4-Channel Scan (ANI0 to ANI3)

(a) Timing example



(b) Block diagram



11.5 Operation in the A/D Trigger Mode

When the CE bit of the ADM0 register is set to 1, A/D conversion is started.

11.5.1 Select mode operation

The A/D converter converts the analog input specified by the ADM0 register. The conversion results are stored in the ADCRn register corresponding to the analog input. In the select mode, the one-buffer mode and four-buffer mode are supported according to the storing method employed for the A/D conversion results (n = 0 to 7).

(1) 1-buffer mode (A/D trigger select 1-buffer)

The A/D converter converts one analog input once. The conversion results are stored in one ADCRn register. The analog input and ADCRn register correspond one to one. (Refer to **Table 11-2**, **Figure 11-9**.)

Each time an A/D conversion is executed, an INTAD interrupt is generated and the AD conversion terminates.

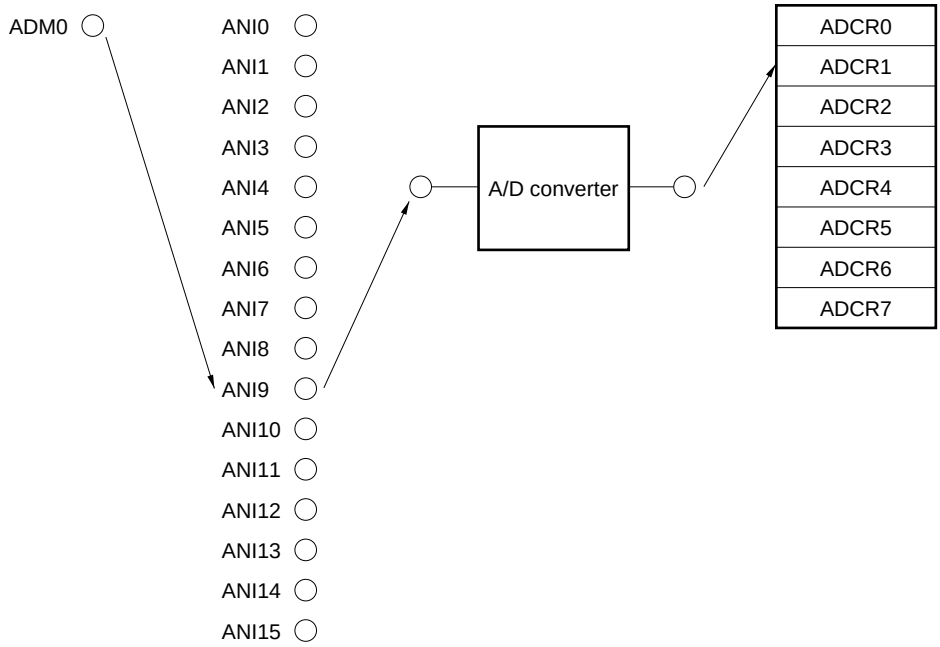
When 1 is written to the CE bit of the ADM0 register, A/D conversion can be restarted.

This mode is suitable for applications which read out the result in each A/D conversion.

Table 11-2. Correspondence between Analog Input Pin and ADCRn Register (1-Buffer Mode (A/D Trigger Select 1-Buffer))

Analog Input	A/D Conversion Result Register
ANI0/ANI8	ADCR0
ANI1/ANI9	ADCR1
ANI2/ANI10	ADCR2
ANI3/ANI11	ADCR3
ANI4/ANI12	ADCR4
ANI5/ANI13	ADCR5
ANI6/ANI14	ADCR6
ANI7/ANI15	ADCR7

Figure 11-9. Example of 1-Buffer Mode (A/D Trigger Select 1-Buffer) Operation (ANI9)



(2) 4-buffer mode (A/D trigger select 4-buffer)

The A/D converter converts one analog input four times and stores the results in four ADCRn registers. (Refer to **Table 11-3**, **Figure 11-10**.) When A/D conversion ends four times, an INTAD interrupt is generated and the A/D conversion terminates.

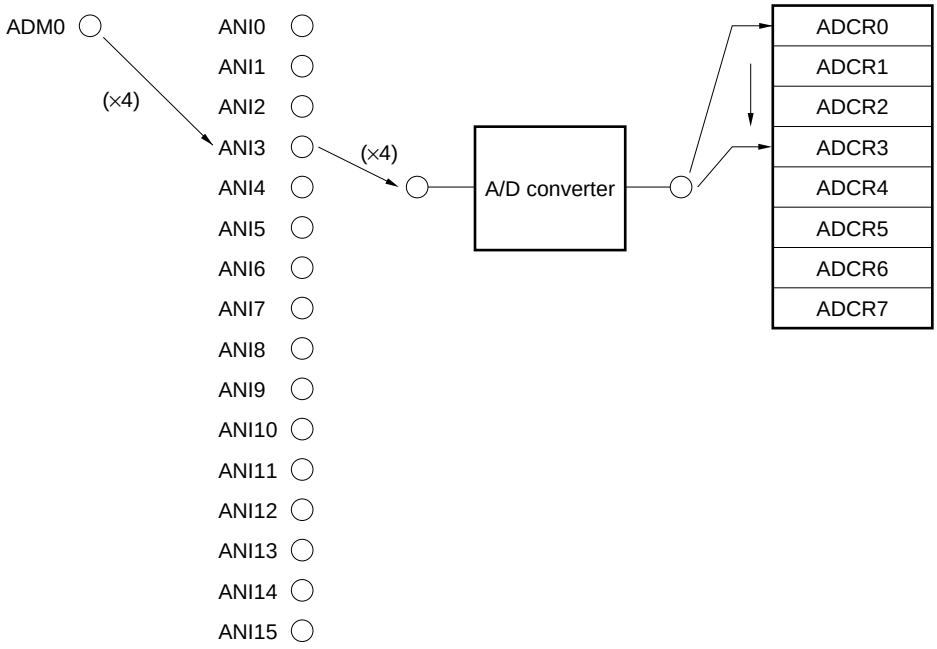
When 1 is written to the CE bit of the ADM0 register, A/D conversion is ended.

This mode is suitable for applications which calculate the average of the A/D conversion result.

Table 11-3. Correspondence between Analog Input Pin and ADCRn Register (4-Buffer Mode (A/D Trigger Select 4-Buffer))

Analog Input	A/D Conversion Result Register
ANI0 to ANI3/ANI8 to ANI11	ADCR0 (First time)
	ADCR1 (Second time)
	ADCR2 (Third time)
	ADCR3 (Fourth time)
ANI4 to ANI17/ANI12 to ANI15	ADCR4 (First time)
	ADCR5 (Second time)
	ADCR6 (Third time)
	ADCR7 (Fourth time)

Figure 11-10. Example of 4-Buffer Mode (A/D Trigger Select 4-Buffer) Operation (ANI3)



11.5.2 Scan mode operation

The analog inputs from ANI0/ANI8 to the analog input specified with the ADM0 register are selected sequentially and converted to digital. The A/D conversion results are stored in the ADCRn register corresponding to the analog input. (Refer to **Table 11-4**, **Figure 11-11**.)

When the conversion of all the specified analog inputs ends, the INTAD interrupt is generated, and A/D conversion is ended.

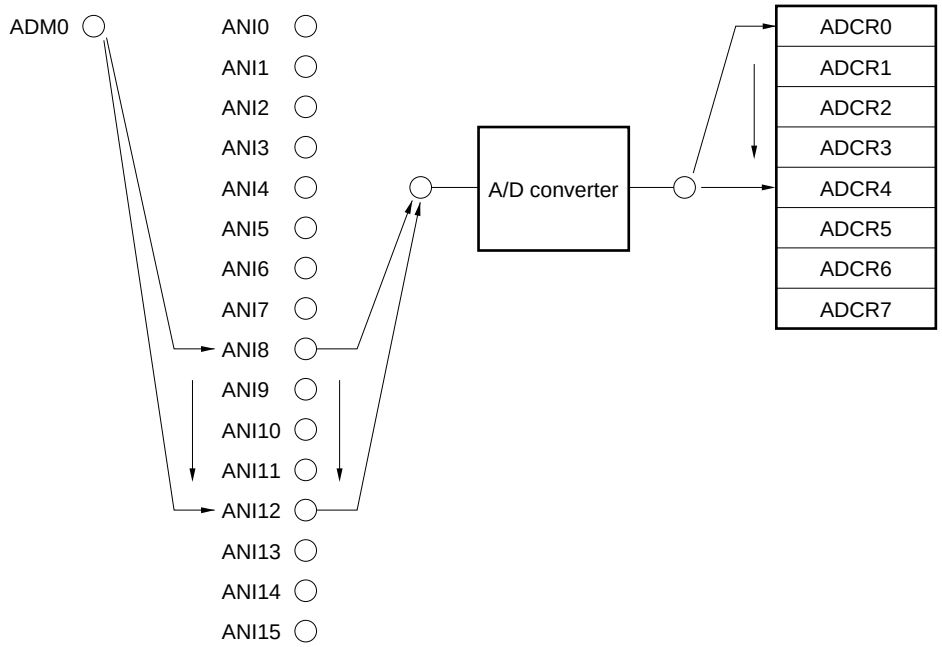
When 1 is written to the CE bit of the ADM0 register, A/D conversion can be restarted.

This mode is suitable for applications which always monitor two or more analog inputs.

Table 11-4. Correspondence between Analog Input Pin and ADCRn Register (Scan Mode (A/D Trigger Scan))

Analog Input	A/D Conversion Result Register
ANI0/ANI8	ADCR0
ANI1/ANI9	ADCR1
ANI2/ANI10	ADCR2
ANI3/ANI11	ADCR3
ANI4/ANI12	ADCR4
ANI5/ANI13	ADCR5
ANI6/ANI14	ADCR6
ANI7/ANI15	ADCR7

Figure 11-11. Example of Scan Mode (A/D Trigger Scan) Operation (ANI8 to ANI12)



11.6 Operation in the Timer Trigger Mode

The A/D converter can set conversion timings with the coincidence interrupt signals of the RPU compare register. TM8 and the capture/compare register (CC83) are used for the timer for specifying the analog conversion trigger. The following two modes are provided according to the specification of the TMC80 register.

(1) One-shot mode

To use the one-shot mode, 1 should be set to the OST8 bit of the TMC80 register (one-shot mode).

When the A/D conversion period is longer than the TM8 period, TM8 generates an overflow, holds 000000H and stops. Thereafter, TM8 does not output the coincidence interrupt signal INTCM83 (A/D conversion trigger) of the compare register, and the A/D converter goes into the A/D conversion standby state. The TM8 count operation restarts when the valid edge of the TCLR8 pin input is detected or when 1 is written to the CE8 bit of the TMC80 register.

(2) Loop mode

To use the loop mode, 0 should be set to the OST8 bit (normal mode) of the TMC80 register.

When the TM8 generates an overflow, the TM8 starts counting from 000000H again, and the coincidence interrupt signal INTCM83 (A/D conversion trigger) of the compare register is repeatedly output and A/D conversion is also repeated.

Coincidence of the compare register can also clear TM8 and restart it.

11.6.1 Select mode operation

The A/D converter converts an analog input (ANI0 to ANI15) specified by the ADM0 register. The conversion results are stored in the ADCRn register corresponding to the analog input. For the select mode, the one-buffer mode and four-buffer mode are provided according to the storing method employed for the A/D conversion results.

(1) 1-buffer mode operation (Timer trigger select 1-buffer)

The A/D converter converts one analog input once and stores the conversion results in one ADCRn register (Refer to **Table 11-5**, **Figure 11-12**).

The A/D converter converts one analog input once using the trigger of the coincidence interrupt signal (INTCM83) and stores the results in one ADCRn register.

An INTAD interrupt is generated for each A/D conversion and the A/D conversion is ended.

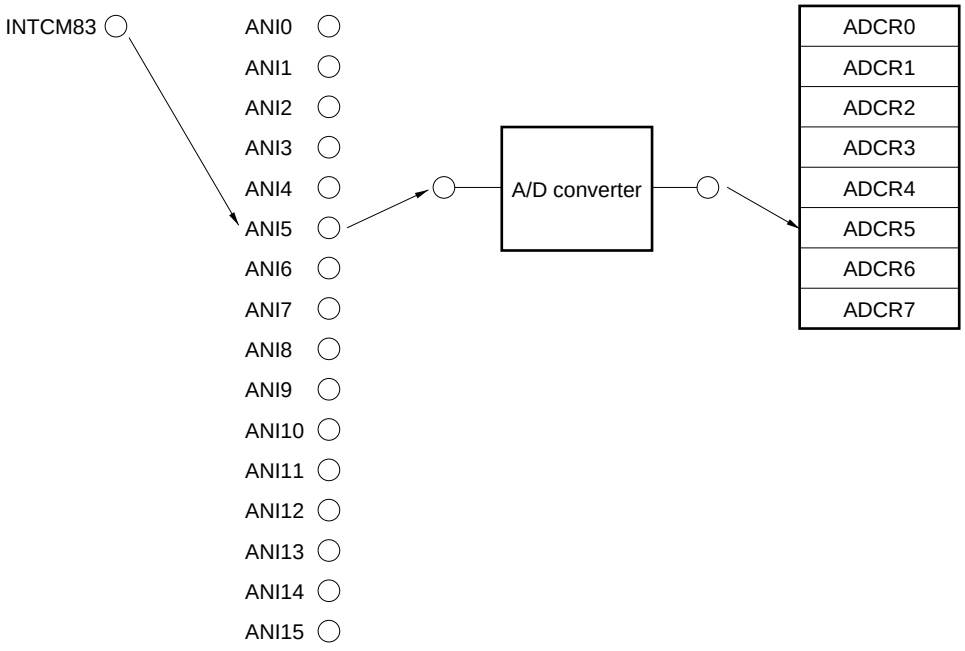
When TM8 is set to the one-shot mode, A/D conversion is ended after one conversion operation. To restart the A/D conversion, input the valid edge to the TCLR8 pin or write 1 to the CE8 bit of the TMC80 register to restart TM8.

When TM8 is set to the loop mode, A/D conversion is repeated each time the coincidence interrupt is generated, unless the CE bit of the ADM0 register is set to 0.

Table 11-5. Correspondence between Analog Input Pin and ADCRn Register (1-Buffer Mode (Timer Trigger Select 1-Buffer))

Trigger	Analog Input	A/D Conversion Result Register
INTCM83 interrupt	ANI0/ANI8	ADCR0
INTCM83 interrupt	ANI1/ANI9	ADCR1
INTCM83 interrupt	ANI2/ANI10	ADCR2
INTCM83 interrupt	ANI3/ANI11	ADCR3
INTCM83 interrupt	ANI4/ANI12	ADCR4
INTCM83 interrupt	ANI5/ANI13	ADCR5
INTCM83 interrupt	ANI6/ANI14	ADCR6
INTCM83 interrupt	ANI7/ANI15	ADCR7

Figure 11-12. Example of 1-Buffer Mode (Timer Trigger Select 1-Buffer) Operation (ANI5)



(2) 4-buffer mode operation (Timer trigger select 4-buffer)

A/D conversion of one analog input is executed four times, and the results are stored in the ADCRn register (Refer to **Table 11-6**, **Figure 11-13**).

The A/D converter converts one analog input four times using the coincidence interrupt signal (INTCM83) as a trigger, and stores the results in four ADCRn registers.

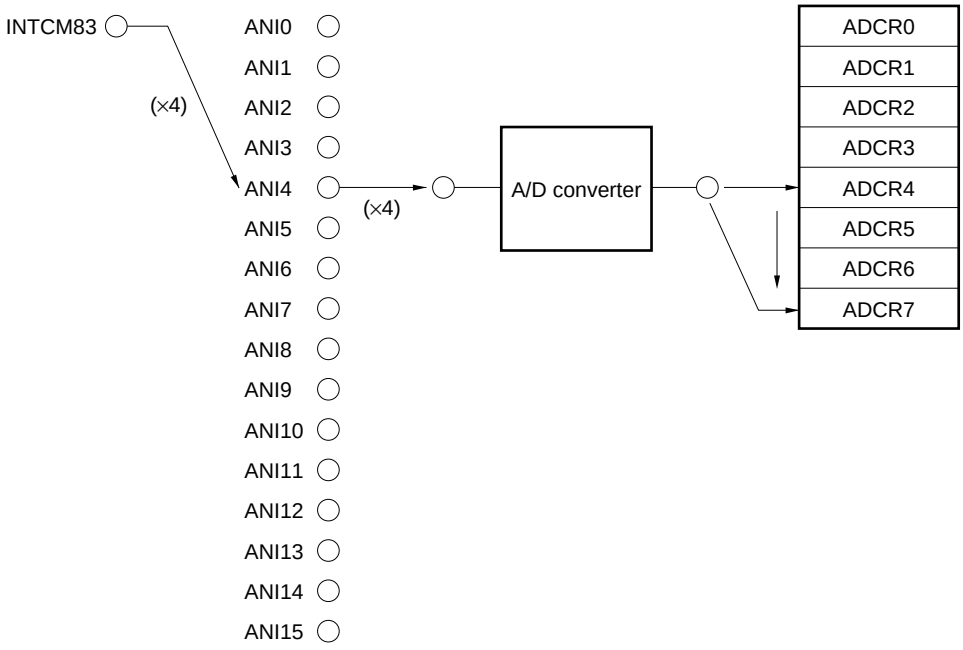
An INTAD interrupt is generated when the four A/D conversion operations end, and the A/D conversion is ended.

This mode is suitable for applications which calculate the average of the A/D conversion result.

Table 11-6. Correspondence between Analog Input Pin and ADCRn Register (4-Buffer Mode (Timer Trigger Select 4-Buffer))

Trigger	Analog Input	A/D Conversion Result Register
INTCM83 interrupt	ANI0 to ANI3/ANI8 to ANI11	ADCR0 (First time) ADCR1 (Second time) ADCR2 (Third time) ADCR3 (Fourth time)
INTCM83 interrupt	ANI4 to ANI7/ANI12 to ANI15	ADCR4 (First time) ADCR5 (Second time) ADCR6 (Third time) ADCR7 (Fourth time)

Figure 11-13. Example of Operation in 4-Buffer Mode (Timer Trigger Select 4-Buffer) (ANI4)



11.6.2 Scan mode operation

The analog inputs from ANI0/ANI8 to the analog input specified with the ADM0 register are selected sequentially, and A/D conversion is executed the number of times specified using the coincidence interrupt as trigger.

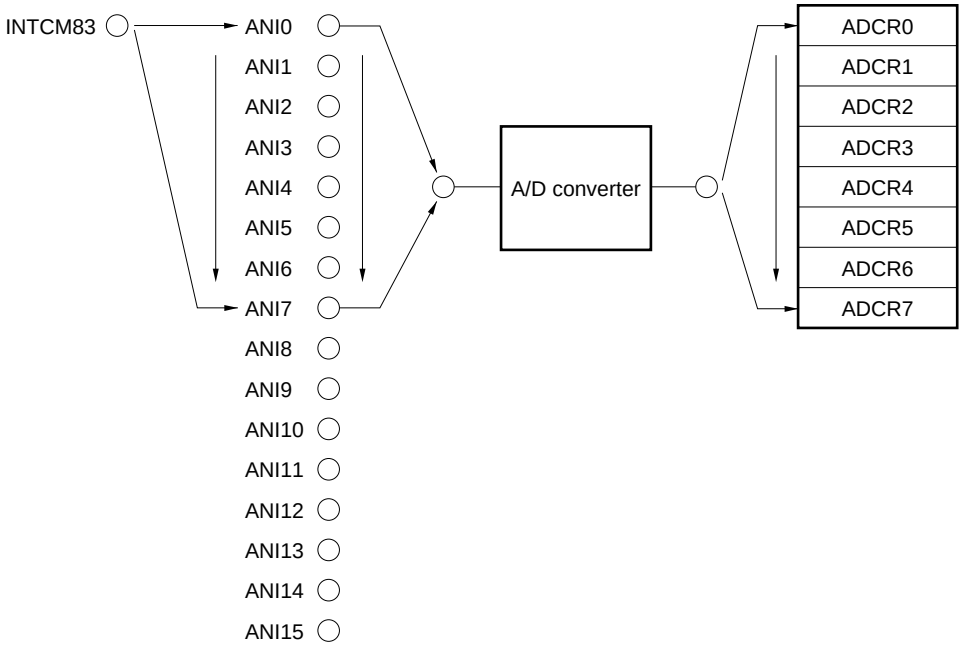
The conversion results are stored in the ADCRn register corresponding to the analog input (refer to **Table 11-7**, **Figure 11-14**). When the conversion of all the specified analog inputs has been ended, the INTAD interrupt is generated and A/D conversion is ended. When the coincidence interrupt is generated after all the specified A/D conversion operations end, A/D conversion is restarted.

This mode is suitable for applications which always monitor two or more analog inputs.

Table 11-7. Correspondence between Analog Input Pin and ADCRn Register (Scan Mode (Timer Trigger Scan))

Trigger	Analog Input	A/D Conversion Result Register
INTCM83 interrupt	ANI0/ANI8	ADCR0
INTCM83 interrupt	ANI1/ANI9	ADCR1
INTCM83 interrupt	ANI2/ANI10	ADCR2
INTCM83 interrupt	ANI3/ANI11	ADCR3
INTCM83 interrupt	ANI4/ANI12	ADCR4
INTCM83 interrupt	ANI5/ANI13	ADCR5
INTCM83 interrupt	ANI6/ANI14	ADCR6
INTCM83 interrupt	ANI7/ANI15	ADCR7

Figure 11-14. Example of Scan Mode (Timer Trigger Scan) Operation (ANI0 to ANI7)



11.7 Operation in the External Trigger Mode

In the external trigger mode, the analog inputs (ANI0 to ANI3) are A/D converted by the ADTRG pin input timing.

The ADTRG pin is also used as the P05 pin. To set the external trigger mode, set the PM05 bit of the PM0 register to 1 and bits TRG1 to TRG0 of the ADM1 register to 10.

For the valid edge of the external input signal during the external trigger mode, the rising edge, falling edge, or both rising and falling edges can be specified using the EGA1 and EGA0 bits of the ADM1 register. For details, refer to **11.3.2 A/D converter mode register 1 (ADM1)**.

11.7.1 Select mode operation

The A/D converter converts one analog input (ANI0 to ANI15) specified by the ADM0 register. The conversion results are stored in the ADCRn register corresponding to the analog input. Two select modes, one-buffer mode and four-buffer mode are available for storing the conversion results.

(1) 1-buffer mode (External trigger select 1-buffer)

The A/D converter converts one analog input using the ADTRG signal as a trigger. The conversion results are stored in one ADCRn register (refer to **Table 11-8**, **Figure 11-15**). The analog input and the A/D conversion result register correspond one to one. An INTAD interrupt is generated after one A/D conversion, and A/D conversion ends.

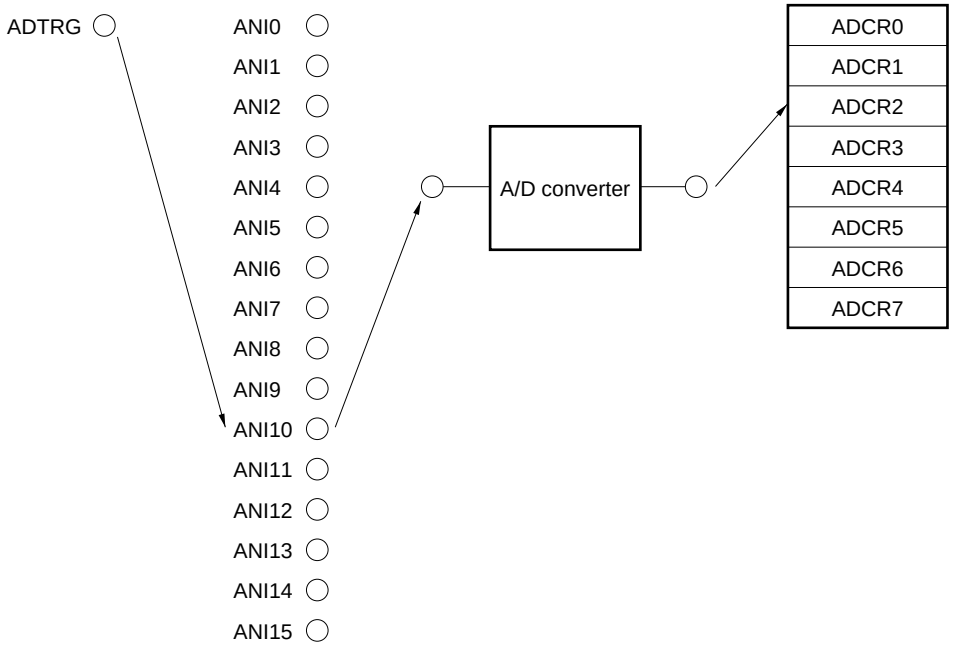
While the CE bit of the ADM0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRG pin.

This mode is suitable for applications which read out the result in each A/D conversion.

Table 11-8. Correspondence between Analog Input Pin and ADCRn Register (1-Buffer Mode (External Trigger Select 1-Buffer))

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANI0/ANI8	ADCR0
ADTRG signal	ANI1/ANI9	ADCR1
ADTRG signal	ANI2/ANI10	ADCR2
ADTRG signal	ANI3/ANI11	ADCR3
ADTRG signal	ANI4/ANI12	ADCR4
ADTRG signal	ANI5/ANI13	ADCR5
ADTRG signal	ANI6/ANI14	ADCR6
ADTRG signal	ANI7/ANI15	ADCR7

Figure 11-15. Example of 1-Buffer Mode (External Trigger Select 1-Buffer) Operation (ANI10)



(2) 4-buffer mode (External trigger select 4-buffer)

The A/D converter converts one analog input four times using the ADTRG signal as a trigger and stores the results in four ADCRn registers (refer to **Table 11-9**, **Figure 11-16**). The INTAD interrupt is generated and conversion ends when the four A/D conversions end.

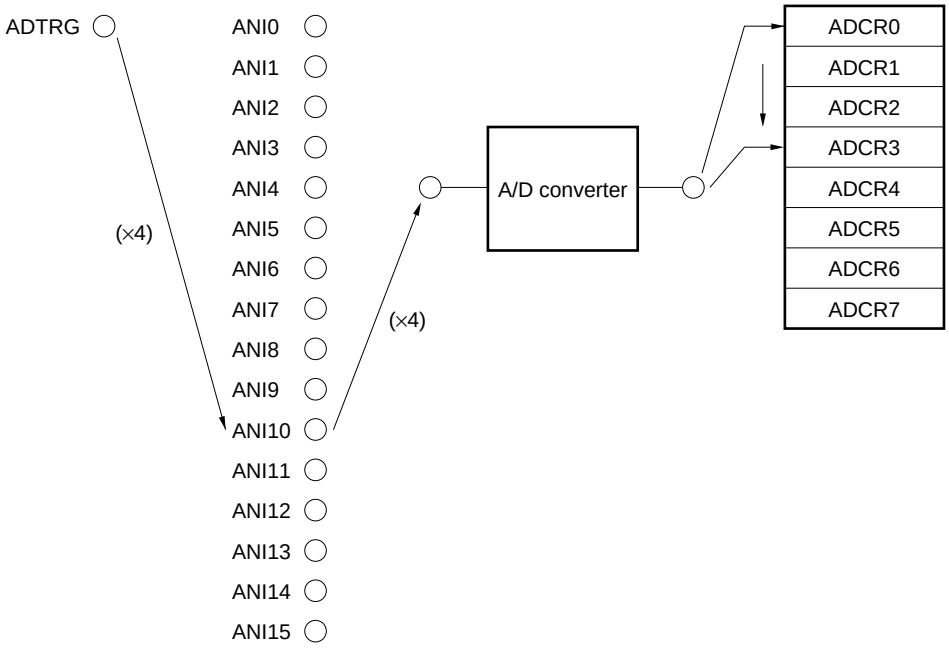
While the CE bit of the ADM0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRG pin.

This mode is suitable for applications which calculate the average of the A/D conversion result.

Table 11-9. Correspondence between Analog Input Pin and ADCRn Register (4-Buffer Mode (External Trigger Select 4-Buffer))

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANI0 to ANI3/ANI8 to ANI11	ADCR0 (First time) ADCR1 (Second time) ADCR2 (Third time) ADCR3 (Fourth time)
ADTRG signal	ANI4 to ANI7/ANI12 to ANI15	ADCR4 (First time) ADCR5 (Second time) ADCR6 (Third time) ADCR7 (Fourth time)

Figure 11-16. Example of 4-Buffer Mode (External Trigger Select 4-Buffer) Operation (ANI10)



11.7.2 Scan mode operation

The analog inputs from ANI0/ANI8 to the analog input specified with the ADM0 register are selected sequentially and converted to digital when triggered by the ADTRG signal. The A/D conversion results are stored in the ADCRn register corresponding to the analog input (refer to **Table 11-10**, **Figure 11-17**). When the conversion of all the specified analog inputs ends, the INTAD interrupt is generated and A/D conversion is ended.

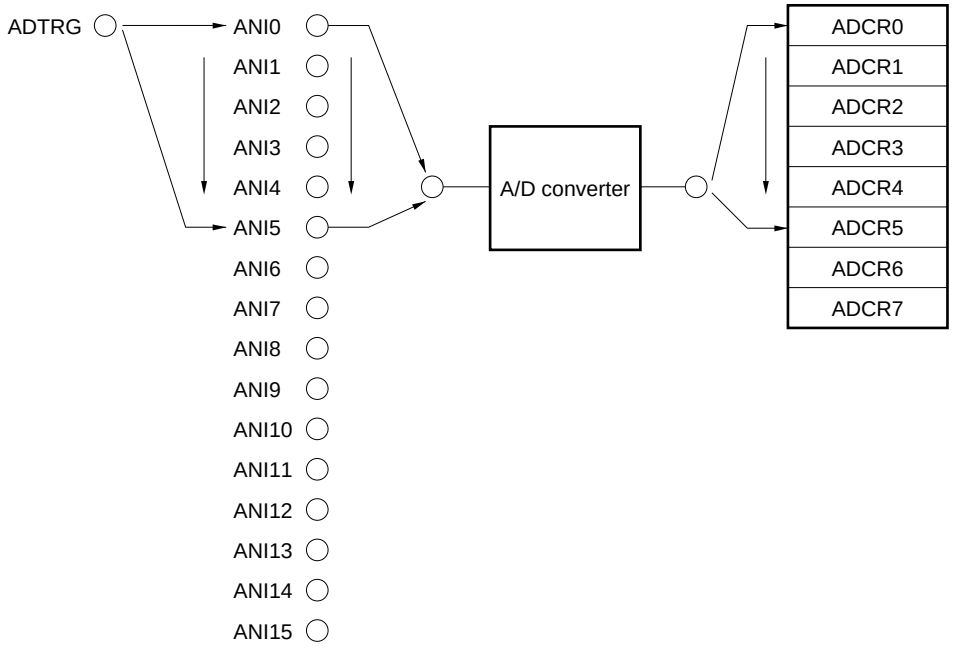
While the CE bit of the ADM0 register is 1, A/D conversion is restarted every time a trigger is input from the ADTRG pin.

This mode is suitable for applications which always monitor two or more analog inputs.

Table 11-10. Correspondence between Analog Input Pin and ADCRn Register (Scan Mode (External Trigger Scan))

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANI0/ANI8	ADCR0
ADTRG signal	ANI1/ANI9	ADCR1
ADTRG signal	ANI2/ANI10	ADCR2
ADTRG signal	ANI3/ANI11	ADCR3
ADTRG signal	ANI4/ANI12	ADCR4
ADTRG signal	ANI5/ANI13	ADCR5
ADTRG signal	ANI6/ANI14	ADCR6
ADTRG signal	ANI7/ANI15	ADCR7

Figure 11-17. Example of Scan Mode (External Trigger Scan) Operation (ANI0 to ANI5)



11.8 Cautions Regarding Operations

(1) Stop of conversion operations

When 0 is written to the CE bit of the ADM0 register during conversion, conversion stops and the conversion results are not stored in the ADCRn register (n = 0 to 7).

(2) Interval of the external/timer trigger

Set the interval (input time interval) of the trigger during the external or timer trigger mode longer than the conversion time specified by the FR2 to FR0 bits of the ADM1 register.

When 0 < interval ≤ conversion operation time

When the next external trigger or timer trigger is input during conversion, conversion stops and conversion starts according to the last external or timer trigger input.

When conversion operations are stopped, the conversion results are not stored in the ADCRn register (n = 0 to 7). However, the number of triggers input are counted, and when an interrupt is generated, the value at which conversion ended is stored in the ADCRn register.

(3) Operation in the standby mode

<1> HALT mode

Continues A/D conversion operations. When canceled by NMI input, the ADM0/ADM1 register and ADCRn register hold the value (n = 0 to 7).

<2> IDLE mode, STOP mode

As clock supply to the A/D converter is stopped, no conversion operations are performed. When canceled using NMI input, the ADM0/ADM1 register and the ADCRn register hold the value (n = 0 to 7). However, when these modes are set during conversion, conversion stops. At this time, if canceled using the NMI input, the conversion operation resumes, but the conversion result written to the ADCRn register will become undefined.

In the IDLE and STOP modes, A/D conversion operation is also stopped to reduce the power consumption. To further reduce current consumption, set the voltage of the AV_{REF} to V_{SS}.

(4) Compare coincide interrupt in the timer trigger mode

The coincidence interrupt of the compare register becomes the A/D conversion start trigger and conversion operations are started. At this time, the coincidence interrupt of the compare register also functions as the coincidence interrupt of the compare register for the CPU. To prevent generation of the coincidence interrupt of the compare register for the CPU, set interrupt disable using the interrupt mask bit (CC8MK3) of the interrupt control register (CC8IC3).

(5) Input range of ANI0 to ANI15

Keep the input voltage of the ANI0 through ANI15 pins to within the rated range. If a voltage greater than AV_{REF} or lower than AV_{SS} (even within the range of the absolute maximum ratings) is input to a channel, the converted value of the channel becomes undefined. Moreover, the values of the other channels may also be affected.

(6) 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 writing of ADCR.

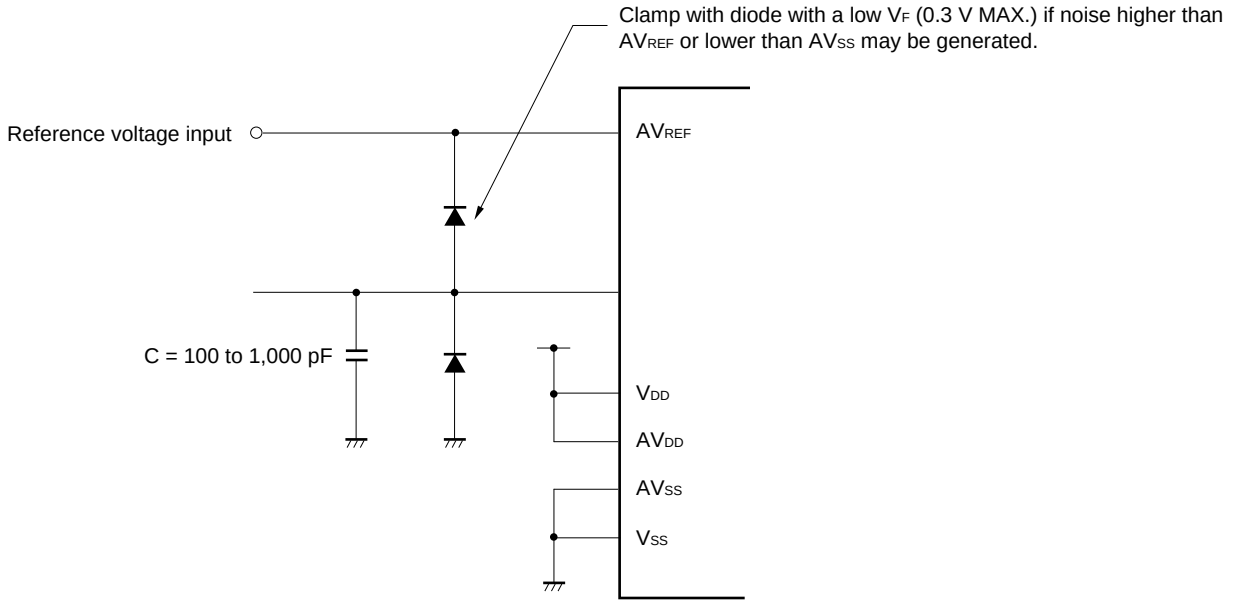
<3> Conflict between writing of ADCR and writing A/D converter mode register 1 (ADM1) or analog input channel specification register (ADS)

When ADM1 or ADS write is performed immediately after ADCR write following A/D conversion end, the conversion result is written to the ADCR register, but the timing is such that the conversion end interrupt request signal (INTAD) is not generated.

(7) Countermeasures against noise

To keep the resolution of 10 bits, prevent noise from being superimposed on the AV_{REF} and ANI0 to ANI15 pins. The higher the output impedance of the analog input source, the heavier the influence of noise. To lower noise, connecting an external capacitor as shown in Figure 11-18 is recommended.

Figure 11-18. Handling of Analog Input Pin



(8) ANI0 to ANI15

The analog input (ANI0 to ANI15) pins are multiplexed with port pins.

To execute A/D conversion with any of ANI0 to ANI15 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.

(9) Input impedance of AV_{REF} pin

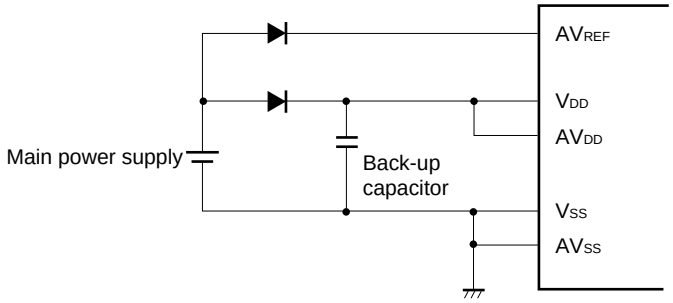
A series resistor string is connected between the AV_{REF} and AV_{SS} pins.

If the output impedance of the reference voltage source is too high, the series resistor string between the AV_{REF} and AV_{SS} pins are connected in parallel, increasing the error of the reference voltage.

(10) AV_{DD} pin

The AV_{DD} pin is the power supply pin of the analog circuit, and also supplies power to the input circuit of ANI0 to ANI15. Even in an application where a back-up power supply is used, therefore, be sure to apply the same voltage as the V_{DD} pin to the AV_{DD} pin as shown in Figure 11-19.

Figure 11-19. Handling of AV_{DD} Pin



CHAPTER 12 DMA FUNCTIONS

12.1 Functions

The DMA (Direct Memory Access) controller transfers data between internal RAM and peripheral I/Os based on DMA requests sent from on-chip peripheral hardware (such as the serial interface, timer, or A/D converter).

This product includes six independent DMA channels that can transfer data in 8-bit and 16-bit units. The maximum number of transfers is 256 (when transferring data in 8-bit units).

After a DMA transfer has occurred a specified number of times, DMA transfer completion interrupt requests (INTDMA0 to INTDMA5) are output individually from the various channels.

The priority levels of the DMA channels are fixed as follows for simultaneous generation of multiple DMA transfer requests.

DMA0 > DMA1 > DMA2 > DMA3 > DMA4 > DMA5

12.2 Transfer Completion Interrupt Request

After a DMA transfer has occurred a specified number of times and the TCn bit in the corresponding DMA channel control register (DCHCn) has been set to 1, a DMA transfer completion interrupt request (INTDMA0 to INTDMA5) occurs on each channel in relation to the interrupt controller. (n = 0 to 5)

12.3 Control Registers

12.3.1 DMA peripheral I/O address registers 0 to 5 (DIOA0 to DIOA5)

These registers are used to set the peripheral I/O register's address for DMA channel n.

These registers can be read/written in 16-bit units.

Figure 12-1. DMA Peripheral I/O Address Registers 0 to 5 (DIOA0 to DIOA5)

After reset: Undefined R/W Address: DIOA0 FFFFF180H DIOA3 FFFFF1B0H
DIOA1 FFFFF190H DIOA4 FFFFF1C0H
DIOA2 FFFFF1A0H DIOA5 FFFFF1D0H

	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, DIOAm, DRAm, DBCn, DCHCm, CORCN, CORRQ, CORADm, xxICx (interrupt control register), ISPR, TM8, CC8m, TMC8a, TOC8, TOVS, TM9, CM90, CM91, CP9m, TMC90, TMC91, EDVa, EDVCa, EVS (n: 0 to 5, m: 0 to 3, a: 0 to 2, x: Refer to 5.3.4 Interrupt control register (xxICn))

12.3.2 DMA internal RAM address registers 0 to 5 (DRA0 to DRA5)

These registers are used to set the on-chip RAM address for DMA channel n. An address is incremented after each transfer is completed, when the DADn bit of the DCHDn register is 0. The incrementation value is “1” during 8-bit transfers and “2” during 16-bit transfers (n = 0 to 5).

These registers can be read/written in 16-bit units.

Figure 12-2. DMA Internal RAM Address Registers 0 to 5 (DRA0 to DRA5)

After reset: Undefined R/W Address: DRA0 FFFFF182H DRA3 FFFFF1B2H

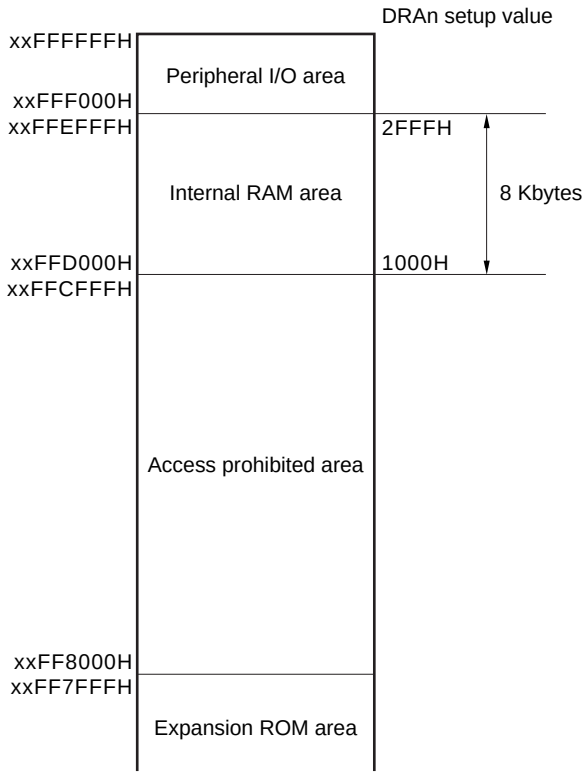
DRA1 FFFFF192H DRA4 FFFFF1C2H

DRA2 FFFFF1A2H DRA5 FFFFF1D2H

	15	14	13													0
DRAn	0	0	RAn13 to RAn0													

(n = 0 to 5)

Figure 12-3. Correspondence between DRAn Setup Value and Internal RAM Area (μPD703039, 703039Y, 703041, and 703041Y)



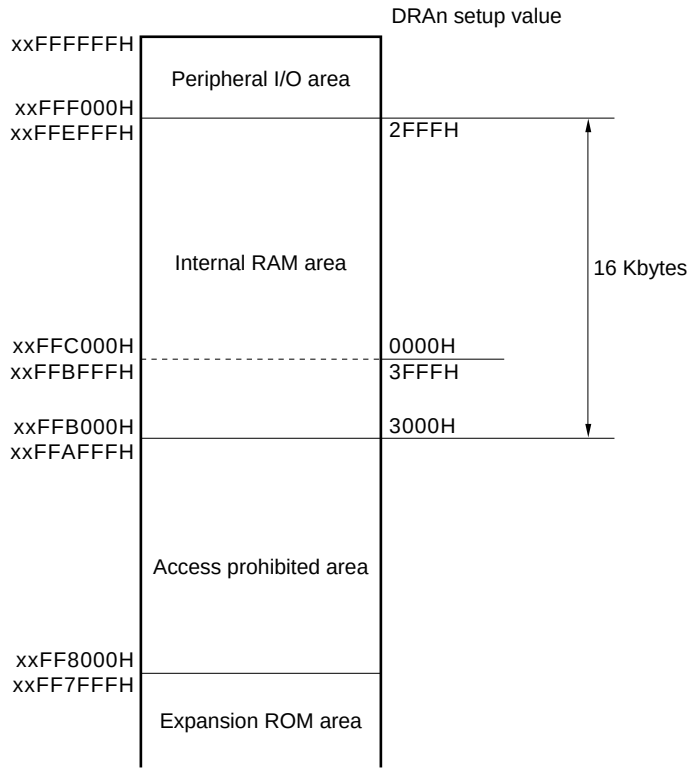
Notes 1. Do not set the odd-number addresses during 16-bit transfer (DSn = 1 in DCHCn register).

2. Set a value between 1000H and 2FFFH in internal RAM address register n of DMA (DRAn). No other value must be set. The following settings are also prohibited because the setup value is incremented to 3000H or above.

- Setting 2FFFH in the DRAn register when DDADn bit = 0 and DSn bit = 0 in the DCHCn register
- Setting 2FFEH or 2FFFH in the DRAn register when DDADn bit = 0 and DSn bit = 1 in the DCHCn register

Remark n = 0 to 5

Figure 12-4. Correspondence between DRAn Setup Value and Internal RAM Area (μPD703040, 703040Y, 70F3040, and 70F3040Y)



Caution Do not set the odd-number addresses during 16-bit transfer (DSn = 1 in DCHCn register).

12.3.3 DMA byte count registers 0 to 5 (DBC0 to DBC5)

These are 8-bit registers that are used to set the number of transfers for DMA channel n.

The remaining number of transfers is retained during the DMA transfers.

A value of 1 is decremented once per transfer if the transfer is a byte (8-bit) transfer, and a value of 2 is decremented once per transfer if the transfer is a 16-bit transfer. The transfers are ended 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. For a 16-bit transfer, the setting of bit 0 is ignored and bit 0 is set to 0 after decrement.

These registers can be read/written in 8-bit units.

Figure 12-5. DMA Byte Count Registers 0 to 5 (DBC0 to DBC5)

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.

12.3.4 DMA channel control registers 0 to 5 (DCHC0 to DCHC5)

These registers are used to control the DMA transfer operation mode for DMA channel n.
These registers can be read/written in 1-bit or 8-bit units.

Figure 12-6. DMA Channel Control Registers 0 to 5 (DCHC0 to DCHC5) (1/2)

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

Channel n	TTYPn1	TTYPn0	Setting of Activation Source for DMA Transfer
0	0	0	INTCSI0/INTIIC0[Note 2]
	0	1	INTCSI1/INTSR0
	1	0	INTAD
	1	1	INTTM00
1	0	0	INTCSI0/INTIIC0[Note 2]
	0	1	INTST0
	1	0	INTP6
	1	1	INTTM10
2	0	0	INTCSI1/INTSR0
	0	1	INTP0
	1	0	INTCP80/INTCM80
	1	1	INTAD
3	0	0	INTCSI3/INTSR1
	0	1	INTCSI2
	1	0	INTCM90
	1	1	INTTM4

Notes 1. TCn is set to 1 when a specified number of transfers are completed, and is cleared to 0 when a write instruction is executed.

2. INTIIC0 is available only in the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

Figure 12-6. DMA Channel Control Registers 0 to 5 (DCHC0 to DCHC5) (2/2)

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

Channel n	TTYPn1	TTYPn0	Setting of Activation Source for DMA Transfer
4	0	0	INTST1
	0	1	INTCSI4
	1	0	INTAD
	1	1	INTTM2
5	0	0	INTCSI3/INTSR1
	0	1	INTCSI4
	1	0	INTCSI2
	1	1	INTTM6

TDIRn	Transfer Direction Control Between Peripheral I/Os and Internal RAM[Note 1]
0	From internal RAM to peripheral I/Os
1	From peripheral I/Os to internal RAM

DSn	Control of Transfer Data Size for DMA Transfer[Note 1]
0	8-bit transfer
1	16-bit transfer

ENn	Control of DMA Transfer Enable/Disable Status[Note 2]
0	Disable
1	Enable (reset to 0 after DMA transfer is completed)

Notes 1. Make sure that the transfer format conforms to the peripheral I/O register specifications (access-enabled data size, read/write, etc.) for the DMA peripheral I/O address registers (DIOAn).

2. After the specified number of transfer is completed, this bit is cleared to 0.

CHAPTER 13 REAL-TIME OUTPUT FUNCTION (RTO)

13.1 Function

The real-time output function transfers pre-set data to real-time output buffer registers n (RTBLn and RTBHn), and then transfers this data with hardware to an external device via the output latches, upon the occurrence of an external interrupt or external trigger. The pins through which the data is output to an external device constitute a port called a real-time output port.

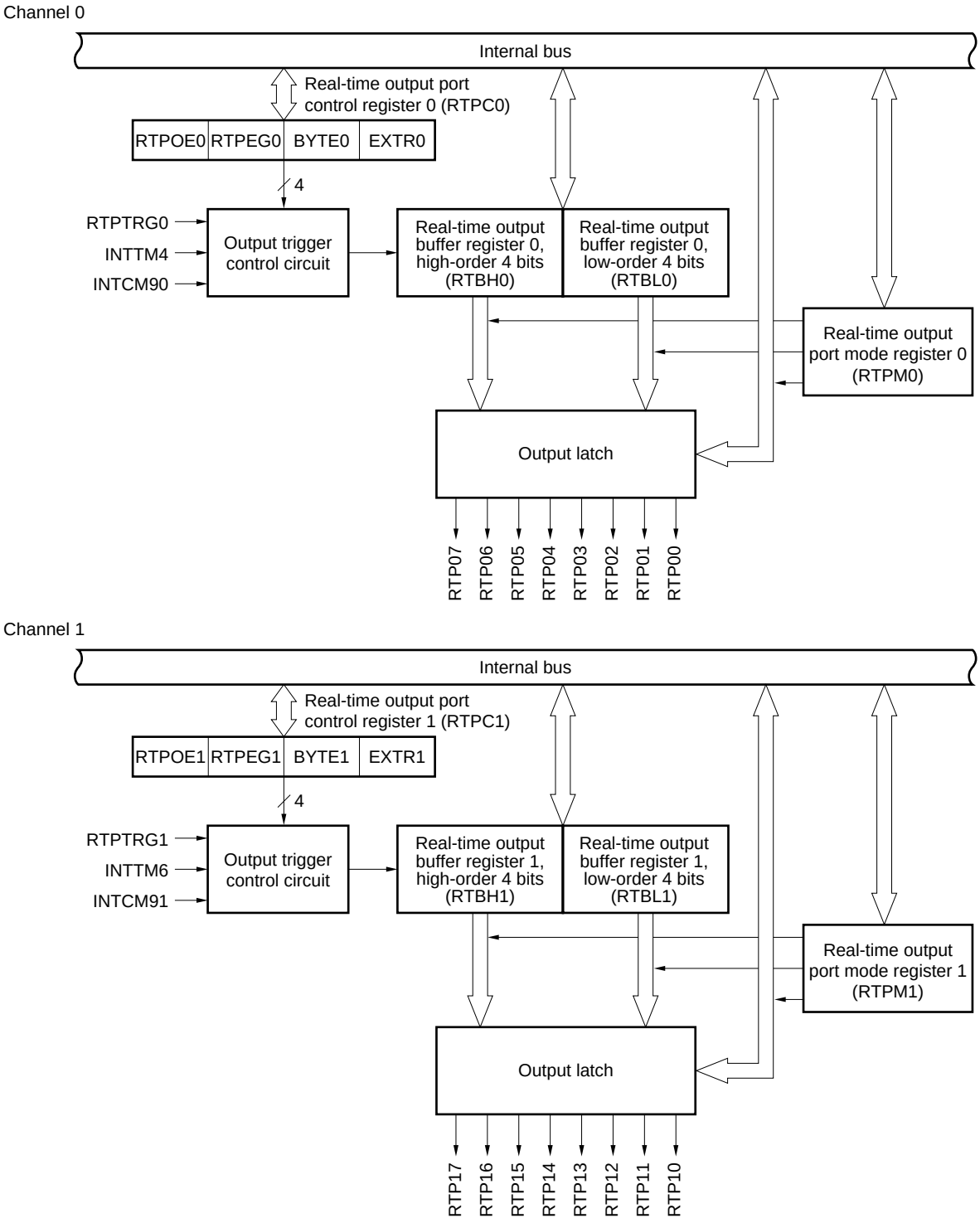
Because RTO can output signals without jitter, it is suitable for controlling a stepping motor.

The real-time output port can be set in port mode or real-time output port mode in 1-bit units.

Figure 13-1 shows the block diagram of RTO.

Remark n = 0, 1

Figure 13-1. Block Diagram of RTO



13.2 Configuration

RTO consists of the following hardware.

Table 13-1. Configuration of RTO

Item	Configuration
Registers	Real-time output buffer registers n (RTBLn, RTBHn)
Control registers	Real-time output port mode register n (RTPMn) Real-time output port control register n (RTPCn)

Remark n = 0, 1

(1) Real-time output buffer registers n (RTBLn, RTBHn)

RTBLn and RTBHn are 4-bit registers that hold output data in advance.

These registers are mapped to independent addresses in the special function register (SFR) area as shown in Figure 13-2.

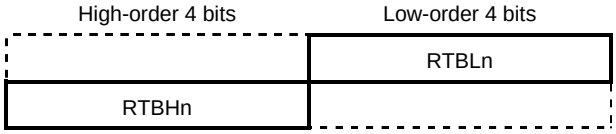
If an operation mode of 4 bits × 2 channels is specified, data can be individually set to RTBLn and RTBHn. The data of both the registers can be read all at once by specifying the address of either of the registers.

If an operation mode of 8 bits × 1 channel is specified, 8-bit data can be set to both RTBLn and RTBHn respectively by writing the data to either of the registers. The data of both the registers can be read all at once by specifying the address of either of the registers.

Figure 13-2 shows the configuration of RTBLn and RTBHn, and Table 13-2 shows the operation to be performed when RTBLn and RTBHn are manipulated.

Remark n = 0, 1

Figure 13-2. Configuration of Real-Time Output Buffer Registers n



Remark n = 0, 1

Table 13-2. Operation When Real-Time Output Buffer Registers n Are Manipulated

Operation Mode	Register to Be Manipulated	Read[Note 1]		Write[Note 2]	
		High-Order 4 Bits	Low-Order 4 Bits	High-Order 4 Bits	Low-Order 4 Bits
4 bits × 2 channels	RTBLn	RTBHn	RTBLn	Invalid	RTBLn
	RTBHn	RTBHn	RTBLn	RTBHn	Invalid
8 bits × 1 channel	RTBLn	RTBHn	RTBLn	RTBHn	RTBLn
	RTBHn	RTBHn	RTBLn	RTBHn	RTBLn

Notes 1. Only the bits set in real-time output port mode register n (RTPMn) can be read. If a bit set in the port mode is read, 0 is read.

2. Set output data to RTBLn and RTBHn after setting the real-time output port until the real-time output trigger is generated.

Remark n = 0, 1

13.3 RTO Control Registers

RTO is controlled by using the following two types of registers:

- Real-time output port mode registers 0 and 1 (RTPM0 and RTPM1)
- Real-time output port control registers 0 and 1 (RTPC0 and RTPC1)

(1) Real-time output port mode registers 0 and 1 (RTPM0 and RTPM1)

These registers select real-time output port mode or port mode in 1-bit units.

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

$\overline{\text{RESET}}$ input sets RTPMn to 00H.

Figure 13-3. Real-Time Output Port Mode Registers 0 and 1 (RTPM0 and RTPM1)

After reset: 00H R/W Address: FFFFF234H, FFFFF23CH

	7	6	5	4	3	2	1	0
RTPMn	RTPMn7	RTPMn6	RTPMn5	RTPMn4	RTPMn3	RTPMn2	RTPMn1	RTPMn0

RTPMm	Selection of Real-Time Output Port (m = 0 to 7)
0	Port mode
1	Real-time output port mode

Cautions 1. Set a port pin to the output mode when it is used as a real-time output port pin.

2. Data cannot be set to the output latch for a port pin set as a real-time output port pin. To set an initial value, therefore, set the data to the output latch before setting the port pin to the real-time output port mode.

Remark n = 0, 1

(2) Real-time output port control registers 0 and 1 (RTPC0 and RTPC1)

These registers set the operation mode and output trigger of the real-time output port.

The relationship between the operation mode and output trigger of the real-time output port is as shown in Table 13-3.

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

$\overline{\text{RESET}}$ input sets RTPCn to 00H.

Figure 13-4. Real-Time Output Port Control Registers 0 and 1 (RTPC0 and RTPC1)

After reset: 00H R/W Address: FFFFF236H, FFFFF23EH

	7	6	5	4	3	2	1	0
RTPCn	RTPOEn	RTPEGn	BYTEn	EXTRn	0	0	0	0

RTPOEn	Control of Operation of Real-Time Output Port
0	Disables operation[Note]
1	Enables operation

RTPEGn	Valid Edge of RTPTRGn
0	Falling edge
1	Rising edge

BYTEn	Operation Mode of Real-Time Output Port
0	4 bits × 2 channels
1	8 bits × 1 channel

EXTRn	Control of Real-Time Output by RTPTRGn Signal
0	Does not use RTPTRGn as real-time output trigger
1	Uses RTPTRGn as real-time output trigger

Note RTPn0 to RTPn7 output 0 if the real-time output operation is disabled (RTPOEn = 0).

Remark n = 0, 1

Table 13-3. Operation Mode and Output Trigger of Real-Time Output Port (Channel 0)

BYTE0	EXTR0	Operation Mode	RTBH0 → Port Output	RTBL0 → Port Output
0	0	4 bits × 2 channels	INTTM4	INTCM90
	1		INTCM90	RTPTRG0
1	0	8 bits × 1 channel	INTCM90	
	1		RTPTRG0	

Table 13-4. Operation Mode and Output Trigger of Real-Time Output Port (Channel 1)

BYTE1	EXTR1	Operation Mode	RTBH1 → Port Output	RTBL1 → Port Output
0	0	4 bits × 2 channels	INTTM6	INTCM91
	1		INTCM91	RTPTRG1
1	0	8 bits × 1 channel	INTCM91	
	1		RTPTRG1	

13.4 Operation

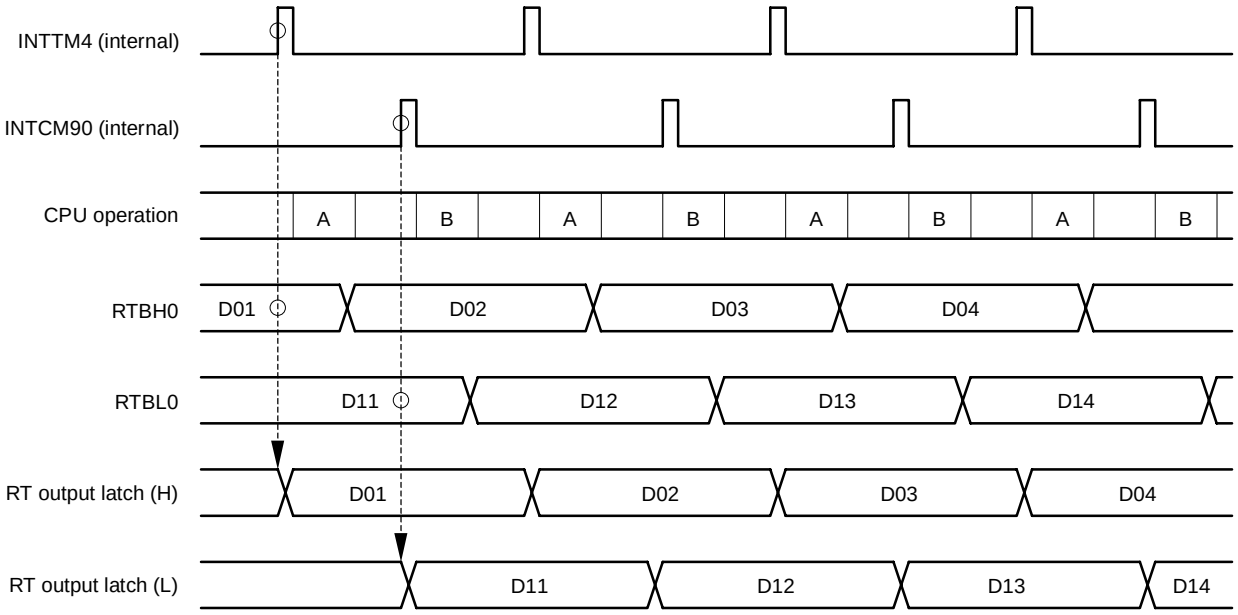
If the real-time output operation is enabled by setting bit 7 (RTPOEn) of real-time output port control register n (RTPCn) to 1, the data of the real-time output buffer registers (RTBHn and RTBLn) is transferred to the output latch in synchronization with the generation of the selected transmit trigger (set by EXTRn and BYTEn[Note]). Of the transferred data, only the data of the bits specified in the real-time output port by the real-time output port mode register n (RTPMn) is output from the bits of RTPn0 to RTPn7. The bits specified in the port mode by RTPMn output 0.

If the real-time output operation is disabled by clearing RTPOEn to 0, RTPn0 to RTPn7 output 0 regardless of the setting of RTPMn.

Note EXTRn: Bit 4 of real-time output port control register n (RTPCn)
BYTEn: Bit 5 of real-time output port control register n (RTPCn)

Remark n = 0, 1

Figure 13-5. Example of Operation Timing of RTO (When EXTR0 = 0, BYTE0 = 0)



A: Software processing by interrupt request input to INTTM4 (RTBH0 write)

B: Software processing by interrupt request input to INTCM90 (RTBL0 write)

13.5 Usage

(1) Disable the real-time output operation.
Clear bit 7 (RTPOEn) of the real-time output port control register n (RTPCn) to 0.

(2) Initialization
- Set the initial value to the output latch.
- Specify the real-time output port mode or port mode in 1-bit units.
  Set the real-time output port mode register n (RTPMn).
- Selects a trigger and valid edge.
  Set bits 4, 5, and 6 (EXTRn, BYTEn, and RTPEGn) of RTPCn.
- Set the initial value that is the same as the output latch to the real-time output buffer registers n (RTBHn and RTBLn).

(3) Enable the real-time output operation.
Set RTPOEn to 1.

(4) Set the output latch of the port to 0 and the next output to RTBHn and RTBLn until the selected transfer trigger is generated.

(5) Set the next real-time output value to RTBHn and RTBLn by interrupt servicing corresponding to the selected trigger.

Remark n = 0, 1

13.6 Cautions

(1) Before performing initialization, disable the real-time output operation by clearing bit 7 (RTPOEn) of the real-time output port control register n (RTPCn) to 0.

(2) Once the real-time output operation is disabled (RTPOEn = 0), be sure to set the initial value that is the same as the output latch to the real-time output buffer registers (RTBHn and RTBLn) before enabling the real-time output operation (RTPOEn = 0 $\rightarrow$ 1).

[MEMO]

CHAPTER 14 PWM FUNCTION

14.1 Outline

- PWMn: 4 channels
- 12- to 16-bit PWM output port
- Main pulse + additional pulse configuration
  Main pulse 4/5/6/7/8 bits
  Additional pulse 8 bits
- Repeat frequency: 78 kHz, 1.2 MHz (f_{PWMC} = 20 MHz, f_{PWMC} = PWM operation clock frequency)
- Pulse width overwrite frequency selection: Each one pulse/256 pulse
- Active level of the PWM output pulse can be selected.
- Operation clock: Can be selected from f_{XX}, $f_{XX}/2$, $f_{XX}/4$, and $f_{XX}/8$. (f_{XX} is the internal system clock)

Remark n = 0 to 3

14.2 Configuration

Figure 14-1 shows the configuration of the output circuit of PWMn.

(1) Prescaler

Divides the frequency of f_{XX} and generates PWM operational clock (f_{PWMC}). Prescaler output is selected by the PWPn0/PWPn1 bit of the PWPRn register.

(2) Reload control

Controls the reload of the modulo values of x-bit down counter and the 8-bit counter.

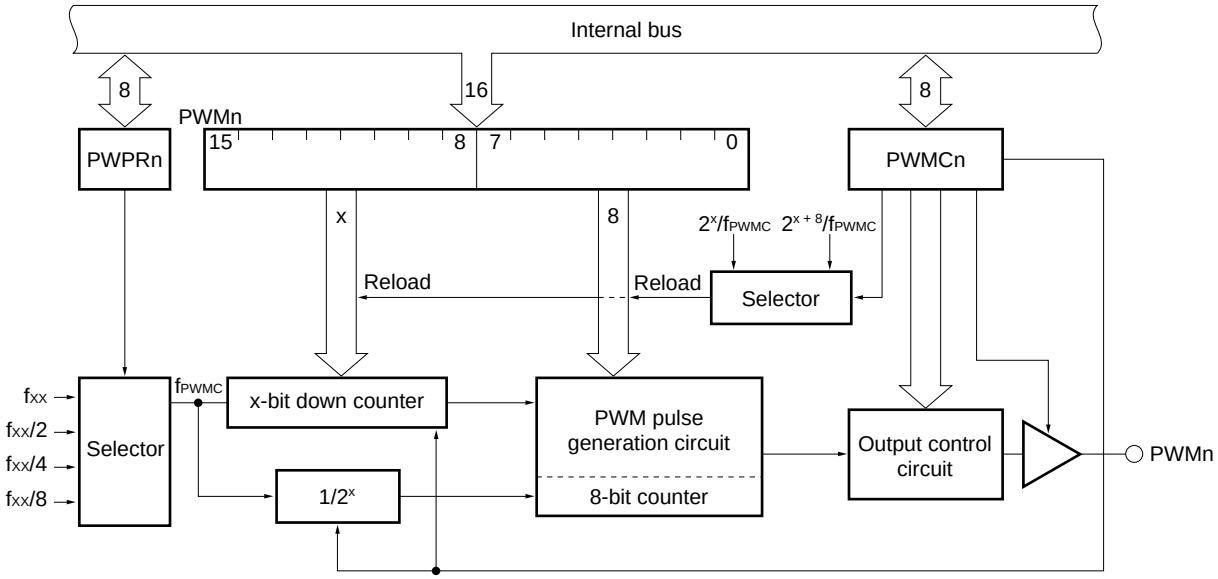
$2^x/f_{PWMC}$ or $2^{x+8}/f_{PWMC}$ is selected by the SYNn bit of the PWMCn register for the reload timing (PWM pulse width rewrite cycle).

(3) x-bit down counter

Controls the output timings of the main pulse.

The value of the modulo H register is loaded to this counter by the reload signal generated in the reload controls and decremented by PWM operational clock (f_{PWMC}).

Figure 14-1. Block Diagram of PWM Function



Remarks 1. n = 0 to 3

x = 4 to 8 (specified with the PMPn2 to PMPn0 bits)

2. f_{PWMC} = PWM operation clock frequency

3. f_{XX} = Main system clock frequency

14.3 Control Registers

(1) PWM control registers 0 to 3 (PWMC0 to PWMC3)

These registers are used to control the PWM operation and to specify the active level of the output and the bit length of the main pulse.

The registers can be read and written in 1-bit or 8-bit units. The contents of the register can be changed even while PWM is operating (PWMEn = 1).

Figure 14-2. PWM Control Registers 0 to 3 (PWMC0 to PWMC3)

After reset: 05H R/W Address: FFFFF0E0H, FFFFF0E8H, FFFFF0F0H, FFFFF0F8H

	7	6	5	4	3	2	1	0
PWMCn	PMPn2	PMPn1	PMPn0	0	0	SYNn	PWMEn	PALVn

PMPn2	PMPn1	PMPn0	Bit Length of the x-Bit Down Counter (Main Pulse)
0	0	0	8 bits
0	0	1	7 bits
0	1	0	6 bits
0	1	1	5 bits
1	0	0	4 bits
Other than above			Setting prohibited

SYNn	PWN Pulse Width Rewrite Period Specification
0	Large period (at intervals of 256 PWM cycles ($2^{x+8}/f_{PWMC}$))
1	Small period (at intervals of 1 PWM cycle ($2^{x}/f_{PWMC}$))

PWMEn	PWM Operation/Stop
0	Stop
1	Operating

PALVn	PWM Active Level Specification
0	Active low
1	Active high

Remark n = 0 to 3

x: Bit number specified by the PRPn2 to PMPn0 bits

(2) PWM prescaler registers 0 to 3 (PWPR0 to PWPR3)

These registers select the PWM operating clock frequency (f_{PWMC}).

The registers can be read and written in 1-bit or 8-bit units. The contents of the registers should be changed when the PWMEn bit of the PWMCn register is held to 0. If the contents are changed while PWMEn is held to 1, unpredictable results may occur.

Figure 14-3. PWM Prescaler Registers 0 to 3 (PWPR0 to PWPR3)

After reset: 00H R/W Address: FFFFF0E4H, FFFFF0ECH, FFFFF0F4H, FFFFF0FCH

	7	6	5	4	3	2	1	0
PWPRn	0	0	0	0	0	0	PWPn1	PWPn0

PWPn1	PWPn0	Operating Clock Frequency (f_{PWMC})
0	0	f_{XX}
0	1	$f_{XX}/2$
1	0	$f_{XX}/4$
1	1	$f_{XX}/8$

Remarks 1. f_{XX}: System clock

2. n = 0 to 3

(3) PWM modulo registers 0 to 3 (PWM0 to PWM3)

The 16-bit registers determine the width of the PWM pulse. The registers can be read and written in 16-bit units. The register consists of these two parts:

<1> Modulo H register (bits 8 to 15)

This register indicates the bit number specified by the PMPn2 to PMPn0 bits of the PWMCn register. This value becomes the precision of main pulse generation. The bit number of this register also determines the pulse width rewrite time.

If the PMPn2 to PMPn0 bits select 4 to 7 bits for the counter, the remaining high-order bits should be set to 0.

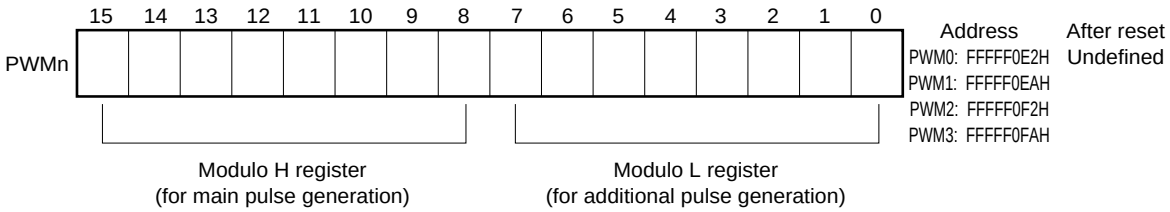
<2> Modulo L register (bits 0 to 7)

The value of this register determines the timing at which an additional pulse is added for fine adjustment (refer to **Figure 14-6**).

The value of this register becomes undefined upon $\overline{\text{RESET}}$ input. Set the data with the initialization program before enabling PWM output.

The value from 0000H to FFFFH can be set to the PWMn register, and PWM output also changes linearly. When 0000H is set, inactive level is retained. When FFFFH is set, one additional pulse ($1/f_{PWMC}$) becomes inactive by one rewrite cycle ($2^{16}/f_{PWMC}$) (refer to **Figure 14-7**).

Figure 14-4. PWM Modulo Registers 0 to 3 (PWM0 to PWM3)



Remark n = 0 to 3

14.4 Operation

14.4.1 Basic operations of PWM

The duty of the PWM pulse output is determined as follows by the value set to the modulo H register of the PWM modulo register (PWMn: n = 0 to 3).

$$\text{Duty of PWM pulse output} = \frac{(\text{Value of modulo H register})^{\text{Note 1}} + 1^{\text{Note 2}}}{2^x}$$

Notes 1. $0 \leq$ (Value of modulo H register) $\leq 2^x - 1$

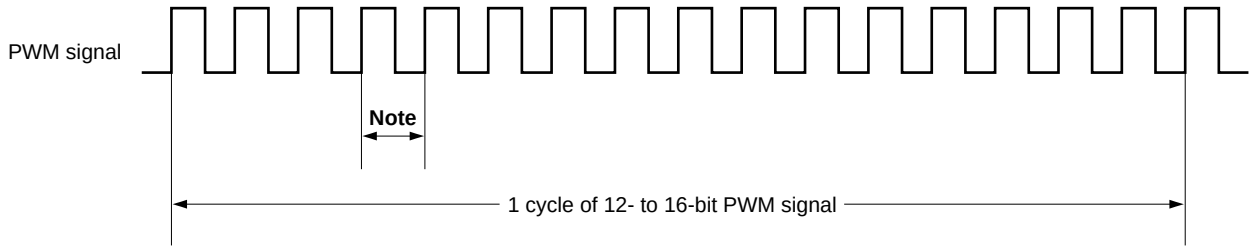
2. With additional pulse

Remark x = 4 to 8

The repeat frequency of the PWM pulse output is the frequency of the 2^x frequency division (= $f_{PWMC}/2^x$) of the PWM clock (f_{PWMC}) of f_{XX} to $f_{XX}/4$ set by the PWM prescaler register (PWPRn), and the minimum pulse width is $1/f_{PWMC}$.

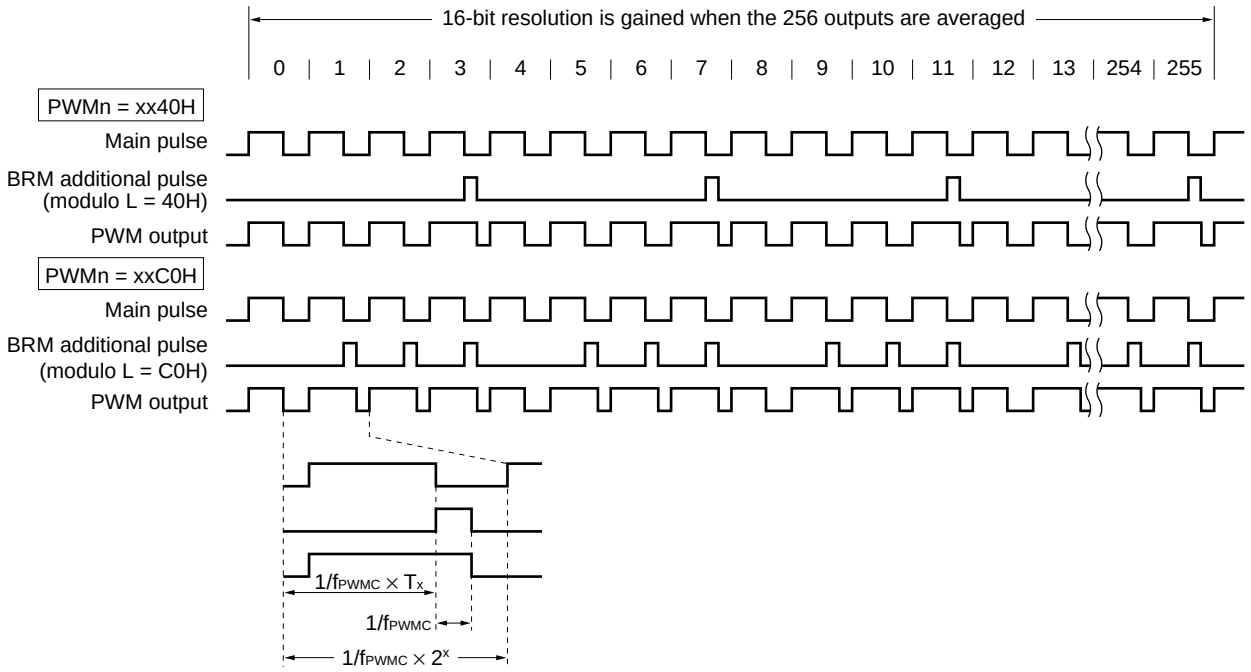
The PWM pulse output realizes 12- to 16-bit resolution by repeatedly outputting the PWM signal with 4- to 8-bit resolution and $f_{PWMC}/2^x$ repeat frequency for 256 times. The PWM pulse signal with 12- to 16-bit resolution is realized in 256 cycles by controlling the addition of the additional pulse ($1/f_{PWMC}$) to the PWM pulse with 4- to 8-bit resolution determined by the modulo H register according to the value of the modulo L register in 1-cycle units.

Figure 14-5. Basic Operations of PWM



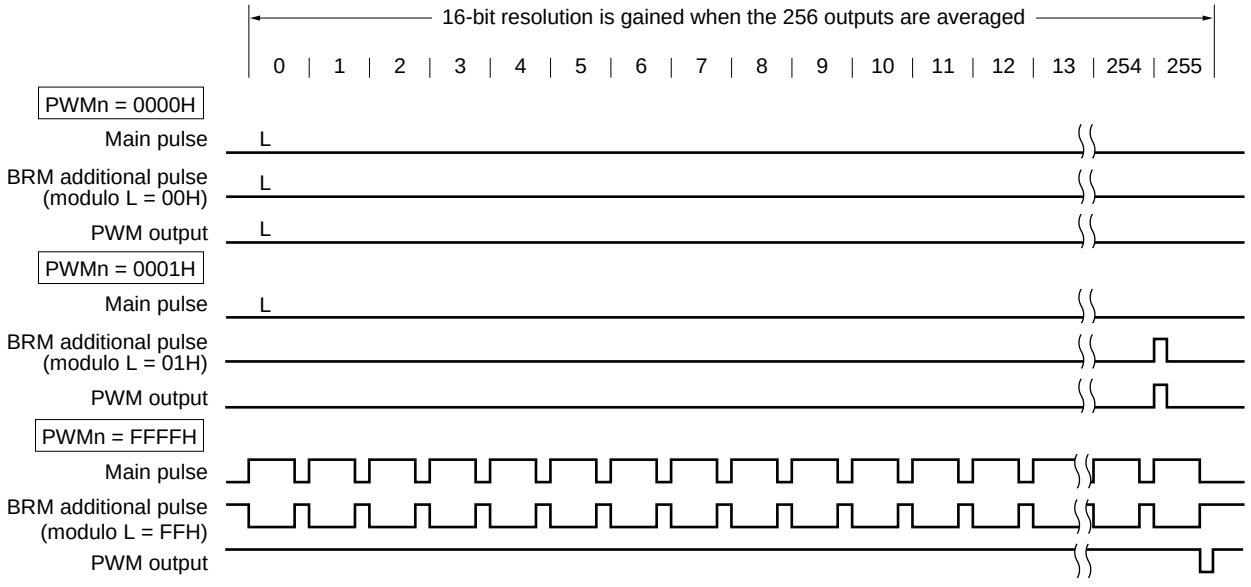
Note PWM pulse: 1 cycle 4- to 8-bit resolution

Figure 14-6. Example of PWM Output by Main Pulse and Additional Pulse



Remark f_{PWMC}: PWM operation clock frequency
T_X: Value of modulo H register
x: Number of bits of modulo H register (selected with PMPn2 to PMPn0 bits)
Active level: High level

Figure 14-7. Example of PWM Output Operation



Remark Condition: Number of bits of modulo H register = 8 bits, 16-bit accuracy, active level = high level

14.4.2 Enabling/disabling PWM operation

To output the PWM pulse, the PWMEn bit of the PWM control register (PWMCn) is set (1) after setting data to the PWM prescaler register (PWPRn) and the PWM modulo register (PWMn) (n = 0 to 3).

Thereby, PWM pulse with active level specified by the PALVn bit of the PWMCn register is output from the PWM output pin.

When the PWMEn bit of the PWMCn register is cleared (0), the PWM output unit immediately stops the PWM output operation, and the PWM output pin becomes inactive.

(1) Setting when PWM operation starts

When the PWMEn bit of the PWMCn register is set, PWMn goes into operation status. However, the PWM pin maintains the port mode status even after the operation status is set until the reload signal of the PWMn register is generated. In addition, the value of the PWMn register is not loaded to the x-bit down counter. Therefore, when the pulse width rewrite timing is set to 2^{x+8} (large cycle: SYNn bit = 0), operation starts in $2^{x+8}/f_{PWMC}$ max. after the PWMEn bit is set. The SYNn bit of the PWMCn register can be rewritten even during PWM output.

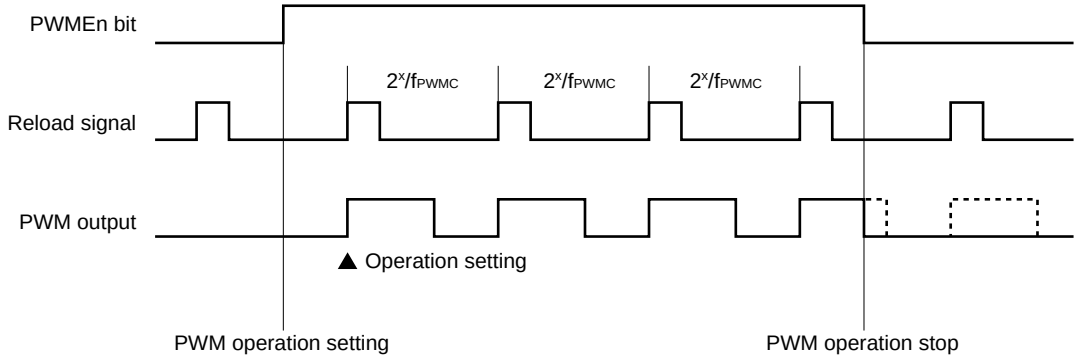
Initialize the following registers before starting PWMn operation.

- PMC10 register: Setting of the control mode
- PWMn register: Setting of the pulse width
- PWPRn register: Specification of the operational clock of the PWM output circuit
- PWMCn register: Specification of the PWM pulse width rewrite cycle, specification of active level of PWM pin, PWM operation control, and selection of the number of bits of the main pulse

(2) Setting when PWM operation stops

When the PWMEn bit of the PWMCn register is reset, PWM operation immediately stops.

Figure 14-8. Operation Timing of PWM



14.4.3 Specification of active level of PWM pulse

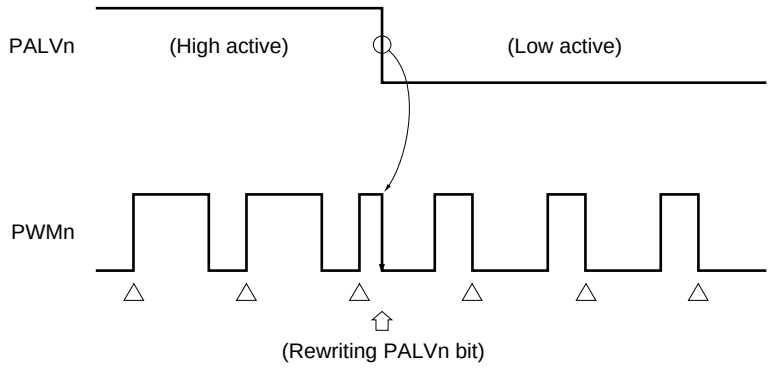
The PALVn bit of the PWM control register (PWMCn) specifies the active level of the PWM pulse output from the PWM output pin (n = 0 to 3).

When the PALVn bit is set (1), a pulse with high active level is output, and when it is cleared (0), a pulse with low active level is output.

When the PALVn bit is rewritten, the active level of the PWM output immediately changes. Figure 14-9 shows the active level setting and the pin status of the PWM output.

The active level of the PWM output can be changed by manipulating the PALVn bit, regardless of the setting of the PWMEn bit (enabling/disabling PWM).

Figure 14-9. Setting of Active Level of PWM Output



Remark PWMEn = 1 (n = 0 to 3)

14.4.4 Specification of PWM pulse width rewrite cycle

Starting PWM output and changing the PWM pulse width are performed in synchronization either with each $2^{(x+8)}$ cycles ($2^{(x+8)}/f_{PWMC}$) of the PWM pulse or with each 1 cycle ($2^x/f_{PWMC}$) of the PWM pulse. The specification of the PWM pulse width rewrite cycle is performed with the SYNn bit of the PWMCn register (n = 0 to 3).

When the SYNn bit is cleared (0), the pulse width is changed at every $2^{(x+8)}$ cycles ($2^{(x+8)}/f_{PWMC}$) of the PWM pulse. Therefore, it will take $2^{(x+8)}$ clocks max. before the pulse with the width corresponding to the data written to the PWMn register is output.

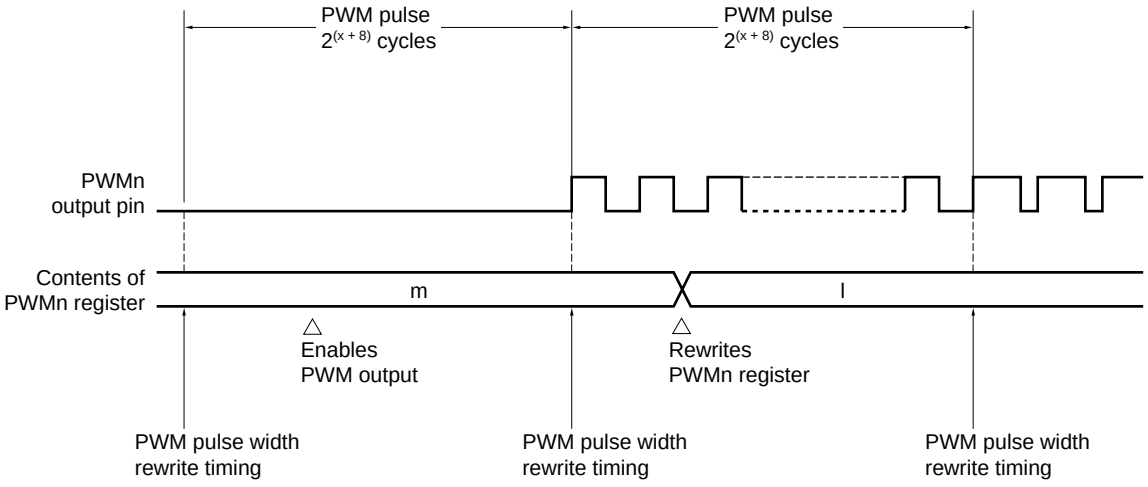
Figure 14-10 shows an example of the PWM output timing.

On the other hand, when the SYNn bit is set (1), the pulse width is changed at every 1 cycle of the PWM pulse ($2^x/f_{PWMC}$). In this case, it will take 2^x clocks max. before the pulse with the width corresponding to the data written to the PWMn register is output.

When the PWM pulse rewriting cycle is specified as every $2^x/f_{PWMC}$ (when the SYNn bit is set (1)), the accuracy of the PWM pulse gained is x bits or more and (x + 8) bits or less, which is lower than the accuracy when the rewriting cycle is specified as $2^{(x+8)}/f_{PWMC}$. However, the response is improved because the repeat frequency is increased.

Figure 14-11 shows an example of the PWM output timing when the rewrite timing is $2^x/f_{PWMC}$.

Figure 14-10. Example 1 of PWM Output Timing (PWM Pulse Width Rewrite Cycle $2^{(x+8)}/f_{PWMC}$)



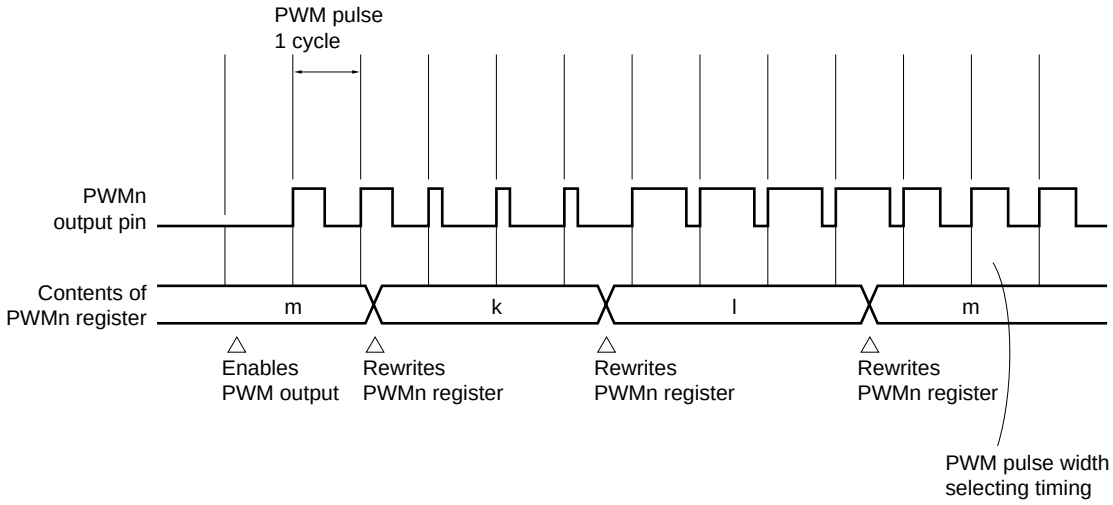
Cautions 1. The pulse width is rewritten at every 256 cycles of the PWM pulse.

2. The accuracy of the PWM pulse is (x + 8) bits.

Remarks 1. m and l are the contents of the PWMn register.

2. n = 0 to 3

Figure 14-11. Example 2 of PWM Output Timing (PWM Pulse Width Rewrite Cycle $2^{x}/f_{PWMC}$)



Cautions 1. The pulse width is rewritten at every 1 cycle of the PWM pulse.

2. The accuracy of the PWM pulse is x bits or more and (x + 8) bits or less.

Remarks 1. k, l, and m are the contents of the PWMn register**.**

2. n = 0 to 3

14.4.5 Repetition frequency

The repetition frequency of the PWMn is shown below (n = 0 to 3).

Main Pulse	Additional Pulse	Repetition Frequency	Pulse Width Rewrite Cycle	
			Large Cycle (SYNn Bit = 0)	Small Cycle (SYNn Bit = 1)
4 bits	8 bits	$f_{PWMC}/16$	$f_{PWMC}/2^{12}$	$f_{PWMC}/2^{4}$
5 bits	8 bits	$f_{PWMC}/32$	$f_{PWMC}/2^{13}$	$f_{PWMC}/2^{5}$
6 bits	8 bits	$f_{PWMC}/64$	$f_{PWMC}/2^{14}$	$f_{PWMC}/2^{6}$
7 bits	8 bits	$f_{PWMC}/128$	$f_{PWMC}/2^{15}$	$f_{PWMC}/2^{7}$
8 bits	8 bits	$f_{PWMC}/256$	$f_{PWMC}/2^{16}$	$f_{PWMC}/2^{8}$

f_{PWMC}: Select from f_{XX}, $f_{XX}/2$, $f_{XX}/4$, and $f_{XX}/8$ by the PWPRn register.

[MEMO]

CHAPTER 15 Vsync/Hsync SEPARATOR

15.1 Outline

A Vsync/Hsync separator separates the Vsync and Hsync signals from the Csync signal.

The circuit can separate a Csync signal in both NTSC and PAL format.

Through the adoption of an up/down counter, the Vsync/Hsync separator (digital noise elimination circuit) can accurately detect the Vsync and Hsync signals even if the Csync signal has narrow noise synchronized with the count timing.

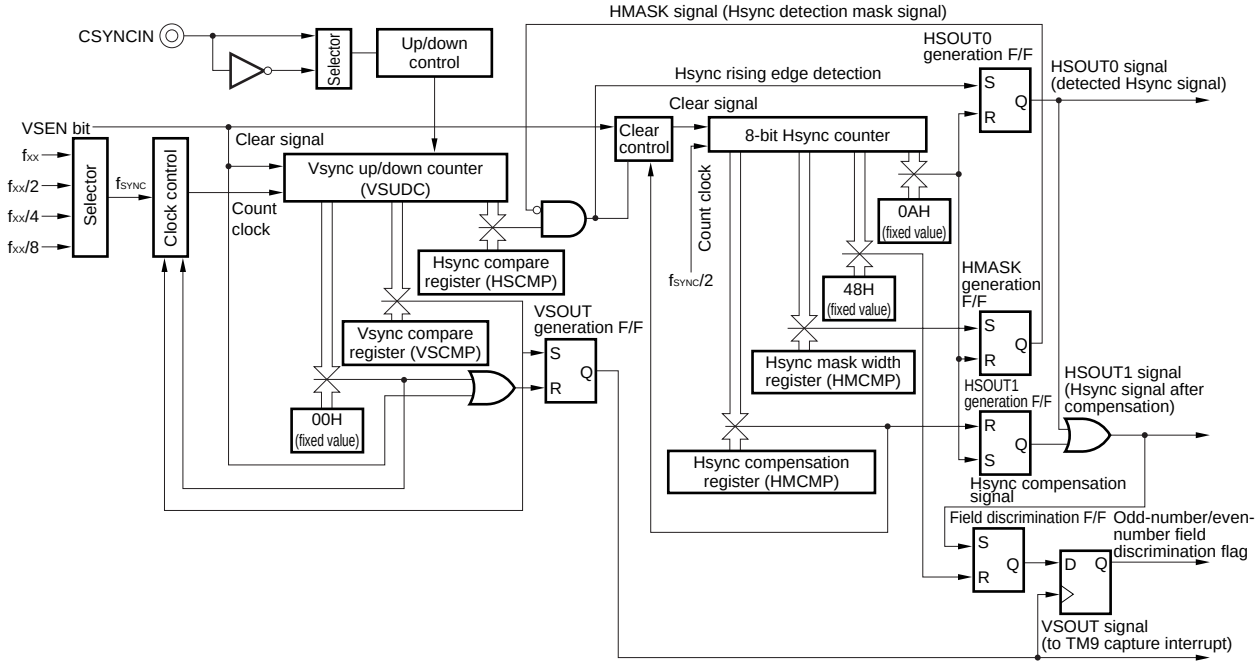
The Vsync/Hsync signal detection timing, mask period, and other settings can be optimized for applications by tuning the up/down counter source clock and the settings of the compare registers.

The detected Vsync signal (VSOUT) and Hsync signal (HSOUT0), as well as the compensated Hsync signal (HSOUT1), can be output to external pins.

The Vsync/Hsync separator has the following functions:

- Separating the Vsync signal from the Csync signal
- Separating the Hsync signal from the Csync signal
- Generating the Hsync signal for itself
- Discriminating between even-number and odd-number fields

Figure 15-1. Block Diagram of Vsync/Hsync Separator



Remark f_{sync}: Vsync/Hsync separator operating clock frequency

15.2 Configuration

The Vsync/Hsync separator consists of the following hardware:

Table 15-1. Configuration of Vsync/Hsync Separator

Item	Configuration
Registers	Vsync up/down counter (VSUDC) Vsync compare register (VSCMP) Hsync compare register (HSCMP) Hsync mask width register (HMCMP) Hsync compensation register (HCCMP)
Control register	Vsync control register (VSC)

(1) Vsync up/down counter (VSUDC)

The Vsync up/down counter is an 8-bit up/down counter.

The counter counts according to f_{sync} (Vsync/Hsync separator operating clock frequency) selected by the VSCK0 and VSCK1 bits of the Vsync control register (VSC). The level of the CSYNCIN signal determines whether the counter counts up or down (active level: up, inactive level: down).

The counter can be read only by an 8-bit manipulation instruction. If the counter is read while an operation is in progress, the f_{sync} input is temporarily stopped[Note], and the current count is read.

$\overline{\text{RESET}}$ input sets the counter to 00H.

Note The count clock (f_{sync}) is masked for a period of $2/f_{CPU}$ (f_{CPU}: CPU operating clock frequency).

(2) Vsync compare register (VSCMP)

The Vsync compare register is an 8-bit compare register.

If the contents of the Vsync compare register (VSCMP) match the contents of the Vsync up/down counter (VSUDC), a match signal is generated. The VSOUT generation flip-flop is set, and the VSOUT signal goes high.

The VSCMP setup value is the count-up limit for VSUDC.

VSCMP is set by an 8-bit manipulation instruction.

$\overline{\text{RESET}}$ input sets VSCMP to 00H.

Cautions 1. Do not rewrite VSCMP while the Vsync/Hsync separator is operating.
2. After setting VSCMP, enable the operation (VSEN = 1).

(3) Hsync compare register (HSCMP)

The Hsync compare register (HSCMP) is an 8-bit compare register.

If the contents of VSUDC match the contents of HSCMP while the Vsync up/down counter (VSUDC) is counting up, a match signal is generated. If the contents match during down counting, no match signal is generated.

VSUDC continues counting up or down even once the contents match the contents of HSCMP.

$\overline{\text{RESET}}$ input sets HSCMP to 00H.

HSCMP is set by an 8-bit manipulation instruction.

Cautions 1. Do not rewrite HSCMP while the Vsync/Hsync separator is operating.
2. After setting HSCMP, enable the operation (VSEN = 1).

(4) 8-bit Hsync counter

The 8-bit Hsync counter is an 8-bit interval counter.

The counter counts according to $f_{sync}/2$, or a half of f_{sync} (Vsync/Hsync separator operating clock frequency) selected by the VSCK0 and VSCK1 bits of the Vsync control register (VSC).

The counter is ready once the first Vsync signal is detected after the VSEN bit of VSC is set to 1. When a first Hsync signal is subsequently detected, the counter starts counting up. (The counter is held to 00H until the first Vsync signal is detected.)

When the Hsync signal is detected or when the contents match the contents of the Hsync compensation register (HCCMP), the counter is cleared and started.

If the VSEN bit is cleared to 0, the counter is held to 00H.

The 8-bit Hsync counter cannot be read nor written.

$\overline{\text{RESET}}$ input sets the counter to 00H.

(5) Hsync mask width register (HMCMP)

The Hsync mask width register (HMCMP) is an 8-bit compare register.

If the contents of HMCMP match the contents of the 8-bit Hsync counter, the Hsync mask signal (H_MASK) is cleared to 0.

HMCMP has both master and slave sections. Only the master section of HMCMP can be read or written. The contents of the slave register are compared with the contents of the 8-bit Hsync counter.

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

$\overline{\text{RESET}}$ input sets the register to 00H.

When any of the following conditions is satisfied, the data is transferred from the master section to the slave section:

- When the VSEN bit of VSC is set to 1
- When a compensated Hsync signal (HSOUT1) is detected (master-to-slave transfer is not performed when the first HSOUT1 signal is detected)

(6) Hsync compensation register (HCCMP)

The Hsync compensation register (HCCMP) is an 8-bit compare register.

When the contents of the Hsync counter match the contents of HCCMP, the counter is cleared and started. The match signal causes the HSOUT1 generation flip-flop to be set and the Hsync signal to be self-generated.

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

$\overline{\text{RESET}}$ input sets the register to 00H.

Cautions 1. Do not rewrite HCCMP while the Vsync/Hsync separator is operating.

2. After setting HCCMP, enable the operation (VSEN = 1).

15.3 Control Register

The Vsync/Hsync separator is controlled by the Vsync control register (VSC).

(1) Vsync control register (VSC)

This register controls the operation of the Vsync/Hsync separator.

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

$\overline{\text{RESET}}$ input sets VSC to 00H.

Figure 15-2. Vsync Control Register (VSC)

After reset: 00H R/W Address: FFFFF0DAH

	7	6	5	4	3	2	1	0
VSC	VSEN	0	0	VOUT	VALV	VFLG	VSCK1	VSCK0

VSCK1	VSCK0	Count Clock (f_{sync}[Note]) Selection
0	0	f_{XX}
0	1	$f_{XX}/2$
1	0	$f_{XX}/4$
1	1	Setting prohibited

VFLG	Odd-Number/Even-Number Field Discrimination
0	Odd-number field
1	Even-number field

VALV	CSYNCIN Signal Active Level Control
0	Normal
1	Inverted CSYNCIN signal is input.

VOUT	Vsync/Hsync Signal Output Control
0	No signal is output from the pins.
1	Vsync signal (VSOUT), Hsync signal before compensation (HSOUT0), and compensated Hsync signal (VSOUT1) are output from the pins.

VSEN	Vsync/Hsync Separator Stop/Operation Control
0	Stop
1	Operation

Note f_{sync} should be set to about 4 MHz.

Remark f_{sync}: Vsync/Hsync separator operating clock frequency

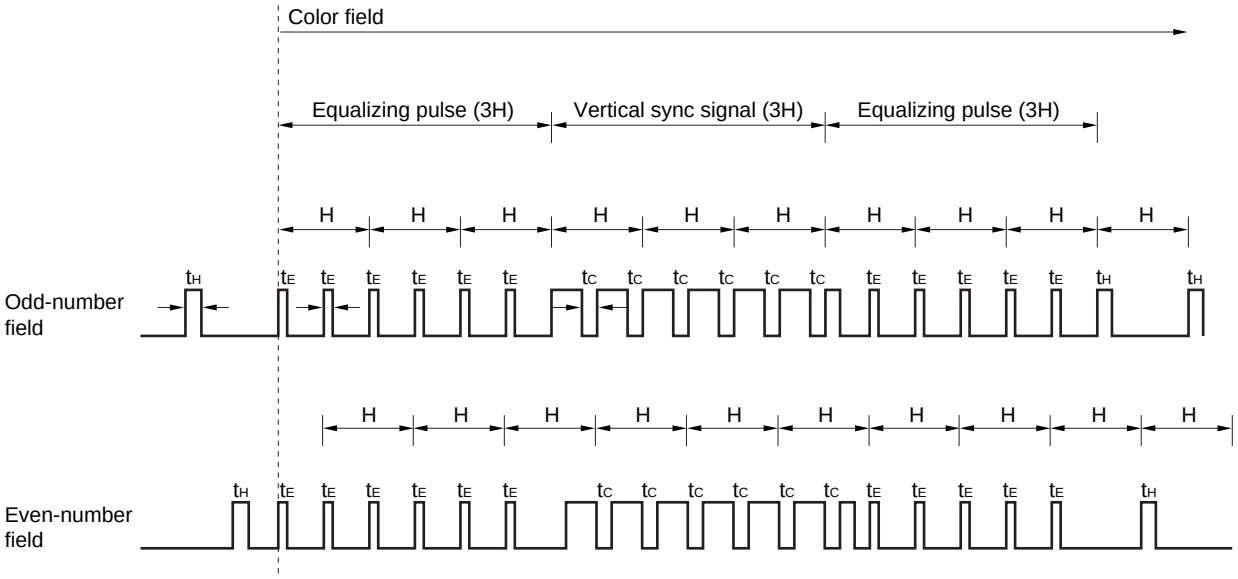
15.4 Operation

The Vsync/Hsync separator is designed to operate optimally when f_{sync} (Vsync/Hsync separator operating clock frequency) is about 4 MHz. Set f_{sync} to about 4 MHz by setting the VSCK1 and VSCK0 bits of the Vsync control register (VSC).

15.4.1 Format of Csync signal

Figure 15-3 shows the format of the Csync signal.

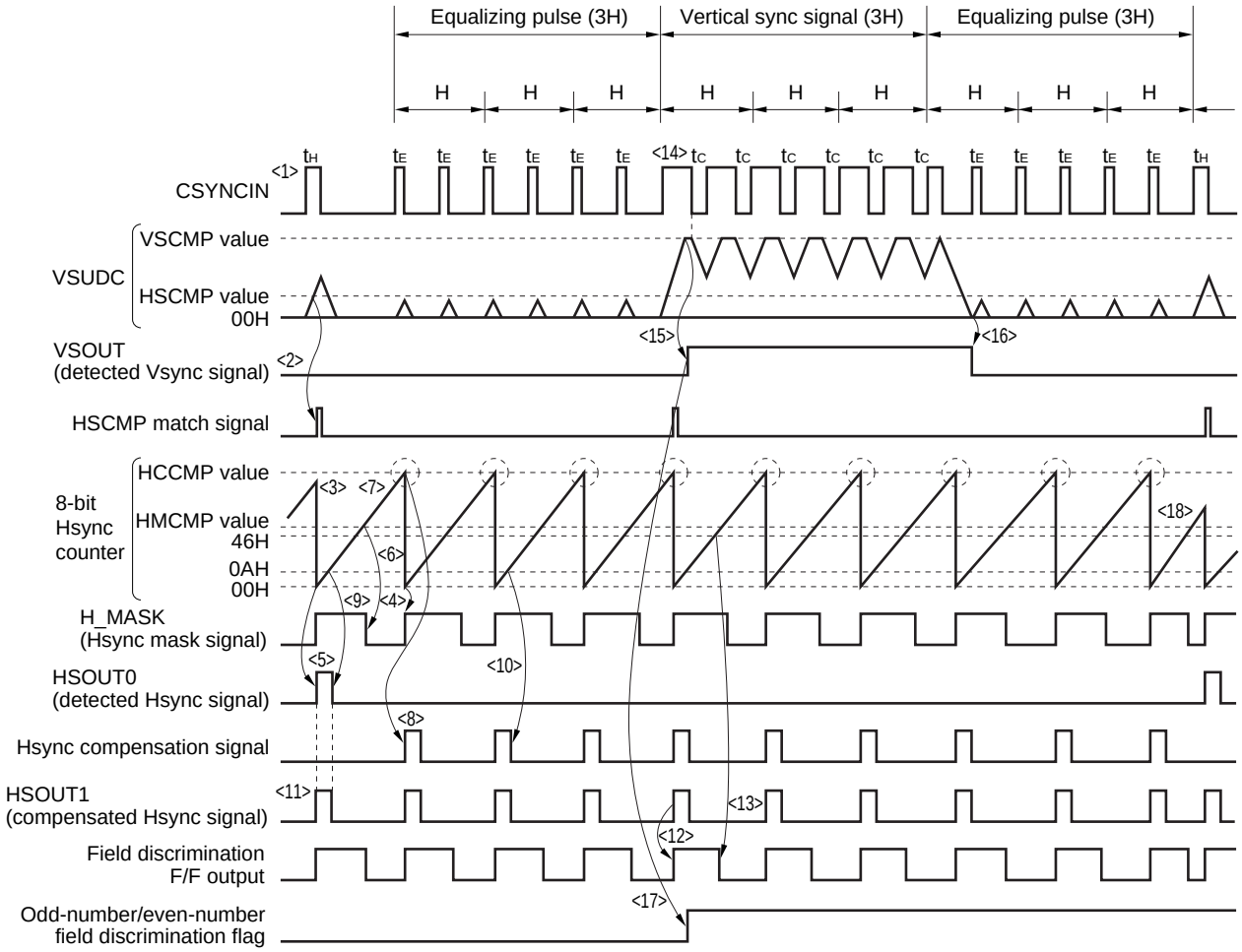
Figure 15-3. Format of Csync Signal



Remark H: Horizontal sync signal: 63.55 μs (NTSC)
64 μs (PAL)
t_H: Pulse width of horizontal sync signal: 4.7 μs
t_E: Equalizing pulse width: 2.5 μs
t_C: Serrated pulse width: 4.4 μs

15.4.2 Basic operation with odd-number field

Figure 15-4. Odd-Number Field



Remark Refer to Table 15-2 for <1> to <18>.

Table 15-2. Operation with Odd-Number Field

Operation		Condition
<1>	VSUDC counting up VSUDC counting down	Signal (1) is input to the CSYNCIN pin. Signal (0) is input to the CSYNCIN pin.
<2>	HSCMP match signal generation[Note]	VSUDC up-count value matches HSCMP value.
<3>	8-bit Hsync counter clearing and starting	HSCMP match signal is generated, and H_MASK signal is set to 0.
<4>	Setting H_MASK signal to 1	8-bit Hsync counter is set to 01H.
<5>	Setting HSOUT0 to 1	8-bit Hsync counter is set to 01H.
<6>	Setting H_MASK signal to 0	8-bit Hsync counter value matches HMCMP value.
<7>	8-bit Hsync counter clearing and starting	HCCMP match signal is generated, and H_MASK signal is set to 0 (when Hsync signal cannot be detected).
<8>	Setting Hsync compensation signal to 1	8-bit Hsync counter is cleared and started in <7>.
<9>	Setting HSOUT0 to 0	8-bit Hsync counter is set to 0AH.
<10>	Setting Hsync compensation signal to 0	8-bit Hsync counter is set to 0AH.
<11>	HSOUT1 signal generation	HSOUT0 signal and Hsync compensation signal are ORed.
<12>	Setting field discrimination F/F signal to 1	HSOUT1 signal is set to 1.
<13>	Setting field discrimination F/F signal to 0	8-bit Hsync counter is set to 46H.
<14>	VSUDC counting up VSUDC counting down	Vsync signal (1) is input to the CSYNCIN pin. Vsync signal (0) is input to the CSYNCIN pin.
<15>	Setting VSOUT signal to 1	VSUDC up-count value matches VSCMP value.
<16>	Clearing VSOUT signal to 0	After VSUDC up-count value matches VSCMP value, VSUDC counts down to 00.
<17>	Setting field discrimination F/F to 1	VSOUT signal is set to 1.

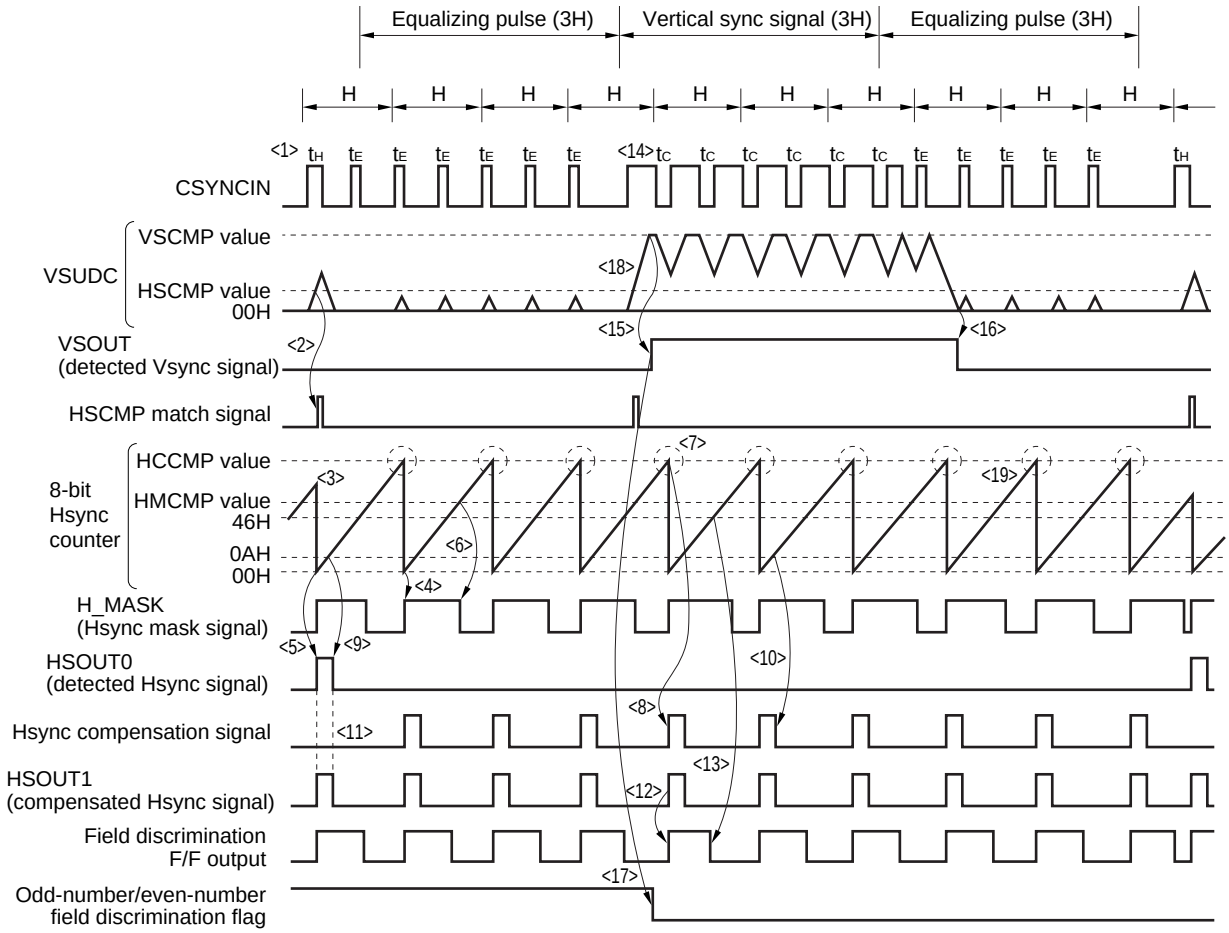
Note The generation is masked while the H_MASK signal is held to 1.

Cautions 1. If the value of the HCCMP register is too large, the timing of the H_MASK signal would be shifted to mask a real Hsync signal (<18>).

2. While an equalizing pulse or vertical sync signal is present, the Hsync signal is not present. During this time, the Hsync signal is self-generated.

15.4.3 Basic operation with even-number field

Figure 15-5. Even-Number Field



Remark Refer to Table 15-3 for <1> to <19>.

Table 15-3. Operation with Even-Number Field

Operation		Condition
<1>	VSUDC counting up VSUDC counting down	Signal (1) is input to the CSYNCIN pin. Signal (0) is input to the CSYNCIN pin.
<2>	HSCMP match signal generation[Note]	VSUDC up-count value matches HSCMP value.
<3>	8-bit Hsync counter clearing and starting	HSCMP match signal is generated, and H_MASK signal is set to 0.
<4>	Setting H_MASK signal to 1	8-bit Hsync counter is set to 01H.
<5>	Setting HSOUT0 to 1	8-bit Hsync counter is set to 01H.
<6>	Setting H_MASK signal to 0	8-bit Hsync counter value matches HMCMP value.
<7>	8-bit Hsync counter clearing and starting	HCCMP match signal is generated, and H_MASK signal is set to 0 (when Hsync signal cannot be detected).
<8>	Setting Hsync compensation signal to 1	8-bit Hsync counter is cleared and started in <7>.
<9>	Setting HSOUT0 to 0	8-bit Hsync counter is set to 0AH.
<10>	Setting Hsync compensation signal to 0	8-bit Hsync counter is set to 0AH.
<11>	HSOUT1 signal generation	HSOUT0 signal and Hsync compensation signal are ORed.
<12>	Setting field discrimination F/F signal to 1	HSOUT1 signal is set to 1.
<13>	Setting field discrimination F/F signal to 0	8-bit Hsync counter is set to 46H.
<14>	VSUDC counting up VSUDC counting down	Vsync signal (1) is input to the CSYNCIN pin. Vsync signal (0) is input to the CSYNCIN pin.
<15>	Setting VSOUT signal to 1	VSUDC up-count value matches VSCMP value.
<16>	Clearing VSOUT signal to 0	After VSUDC up-count value matches VSCMP value, VSUDC counts down to 00.
<17>	Setting field discrimination F/F to 0	VSOUT signal is set to 1.

Note The generation is masked while the H_MASK signal is held to 1.

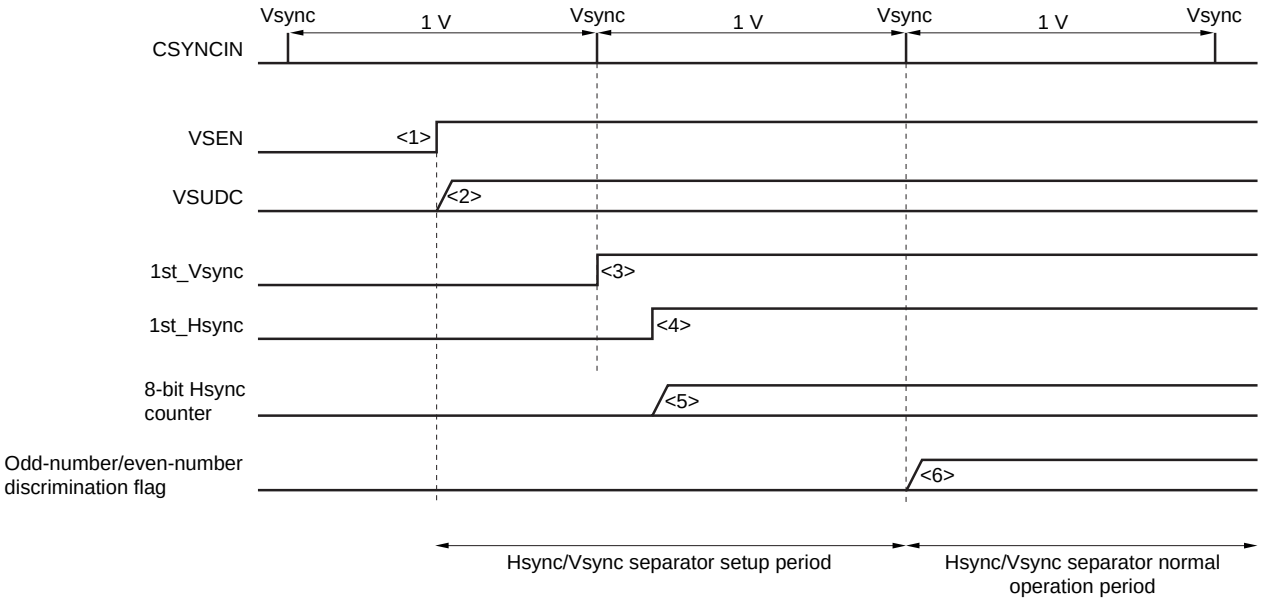
Cautions 1. While VSUDC is counting up from 00H to the VSCMP value (<18>), the HSCMP match signal is generated when the value matches the contents of the HSCMP register. Because the signal is not a real Hsync signal, the HMCMP register should be set to such a value that the HSCMP match signal is masked by the H_MASK signal.

2. If the value of the HCCMP register is too large, the timing of the H_MASK signal would be shifted to mask a real Hsync signal (<19>).

3. While an equalizing pulse or vertical sync signal is present, the Hsync signal is not present. During this time, the Hsync signal is self-generated.

15.4.4 Operation at activation

Figure 15-6. Operation at Activation



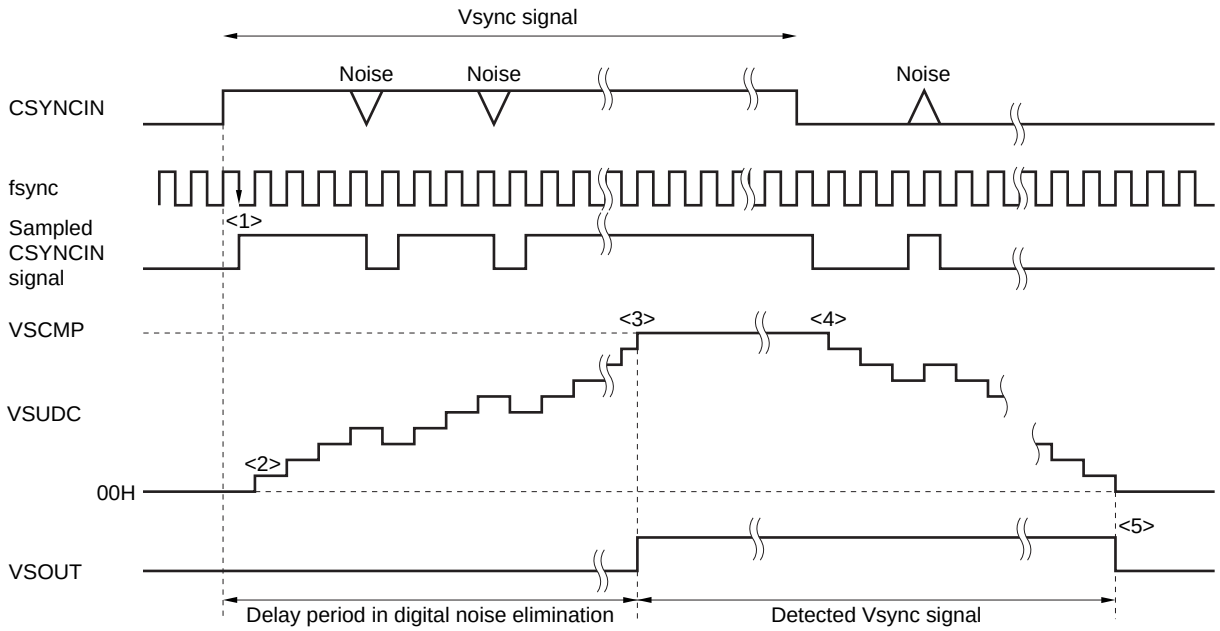
1. The VSEN bit of the Vsync control register (VSC) is set to 1 (<1>).
   VSUDC starts counting (<2>).
   Vsync signal separation starts.
2. If the Vsync signal is detected, the Vsync signal detection signal (1st_Vsync signal) becomes active (<3>).
   If the Hsync signal is detected, the Hsync signal detection signal (1st_Hsync signal) becomes active (<4>).
   The 8-bit Hsync counter starts operating (<5>), and Hsync signal separation starts.
3. When a second Vsync signal is detected, odd-number/even-number field discrimination starts (<6>).

Cautions 1. As shown in Figure 15-6, at least two Vsync signals should be detected before the Vsync/Hsync separator operates normally.

2. At least one compensated Hsync signal (HSOUT1 signal) should be detected between the first and second Vsync signals.

15.4.5 Vsync signal separation

Figure 15-7. Vsync Signal Separation



1. The signal input from the CSYNCIN pin is sampled at the f_{sync} clock timing (falling edge of f_{sync}) (<1>). If the sampled CSYNCIN signal is 1, the Vsync up/down counter (VSUDC) counts up. If the sampled signal is 0, the counter counts down.
2. The counting up or down of VSUDC is controlled according to the sampled CSYNCIN signal. Using f_{sync} as the count clock, VSUDC counts up or down in synchronization with the rising edge of f_{sync} (<2>).
3. If the contents of VSUDC match the contents of the Vsync compare register (VSCMP) (<3>), a match signal is generated, and the VSOUT signal is set to 1. The upper limit of the VSUDC value is the VSCMP value. Even if the CSYNCIN signal is set to 1, the VSUDC value does not exceed the value set in VSCMP.
4. At the end of the Vsync period, the CSYNCIN signal is set to 0 for a longer period. VSUDC counts down (<4>) to 00H after a certain period (<5>). When VSUDC is set to 00H, a 00H match signal is generated, and the VSOUT signal is cleared to 0. While VSUDC is held to 00H, no underflow occurs, even if the CSYNCIN signal is set to 0. The value 00H is held. If the CSYNCIN signal is set to 1, VSUDC starts counting up again.
5. The VSOUT signal can be selected as a capture source of the TM9 capture register (CP91). Meanwhile, interrupt INTCP91 occurs at an edge (specifiable) of VSOUT.

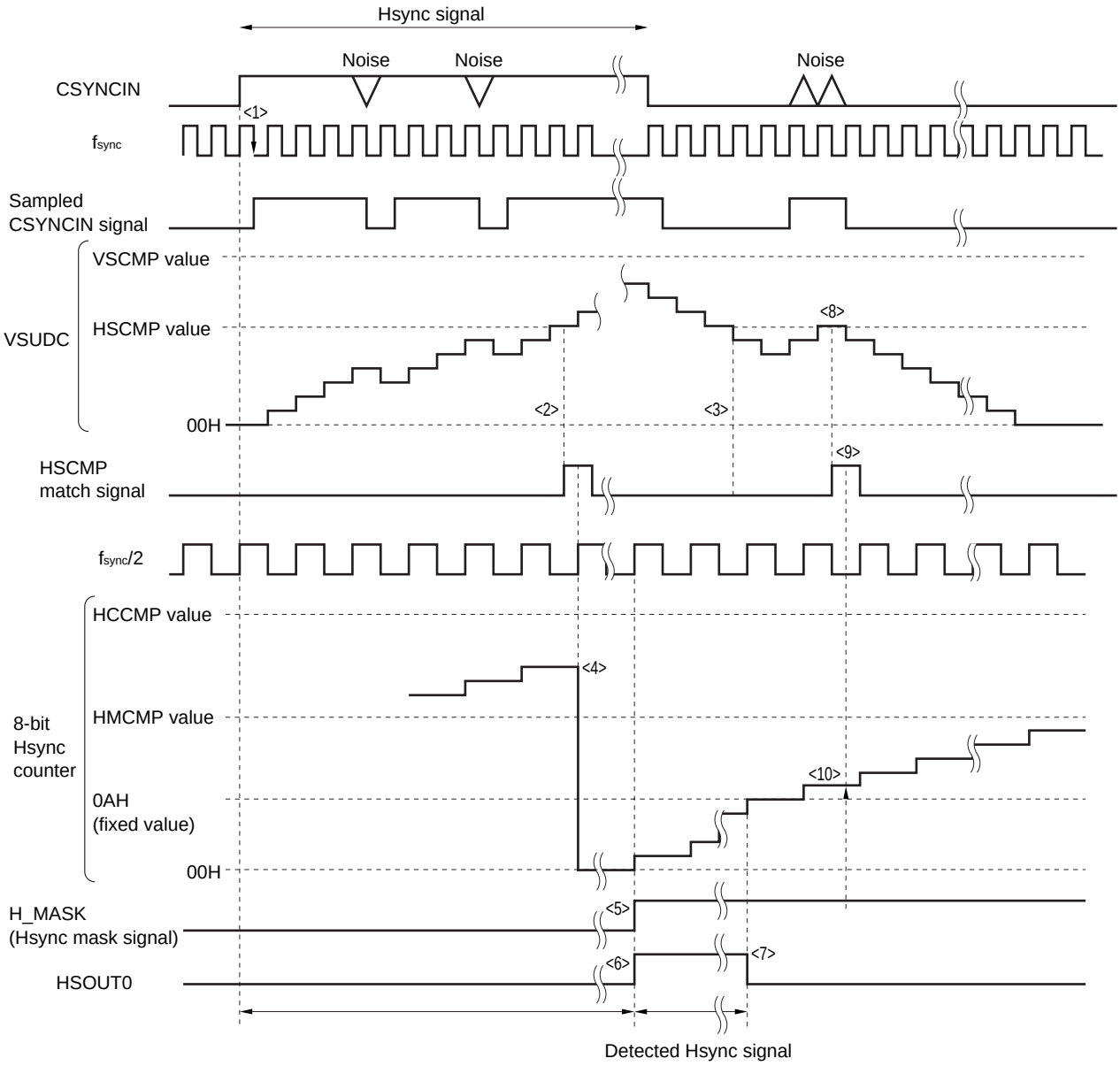
Caution For the purpose of digital noise elimination, the Vsync signal is detected after the following delay from the actual Vsync signal:

$$\text{Vsync signal detection delay} \fallingdotseq \text{(VSCMP setup value)}/f_{sync} + \text{Noise period}$$

Remark f_{sync}: Vsync/Hsync separator operating clock frequency

15.4.6 Hsync signal separation

Figure 15-8. Hsync Signal Separation



1. The signal input from the CSYNCIN pin is sampled at the f_{sync} clock timing (falling edge of f_{sync}) (<1>). If the sampled CSYNCIN signal is 1, the Vsync up/down counter (VSUDC) counts up. If the sampled signal is 0, the counter counts down.
2. If the contents of VSUDC match the contents of the Hsync compare register (HSCMP) while VSUDC is counting up, a match signal is generated (<2>). VSUDC continues counting up or down without being affected by the HSCMP setup value.

   If the contents of VSUDC match the contents of HSCMP while VSUDC is counting down, no match signal is generated (<3>).

3. If a match signal indicating that VSUDC matches HSCMP is generated when the Hsync mask signal (H_MASK) is held to 0, the 8-bit Hsync counter is cleared and started (<4>). The HSOUT0 signal is set to 1 (<5>). At the same time, the Hsync mask signal (H_MASK) is set to 1 (<6>).
4. When the 8-bit Hsync counter is set to 0AH, a 0AH match signal is generated, and the HSOUT0 signal is cleared to 0 (<7>). Consequently, the active width of the detected Hsync signal is always $9/f_{sync}$ (4.5 μs when f_{sync} is 4 MHz).
5. If noise causes VSUDC to match HSCMP (<8>), the Hsync detection signal is set to 1 (<9>). Meanwhile, the Hsync mask signal (H_MASK) masks the Hsync detection signal. Accordingly, the 8-bit Hsync counter is neither cleared nor started, and the count-up operation can continue (<10>).

Cautions 1. For the purpose of digital noise elimination, the Hsync signal is detected after the following delay from the actual Hsync signal:

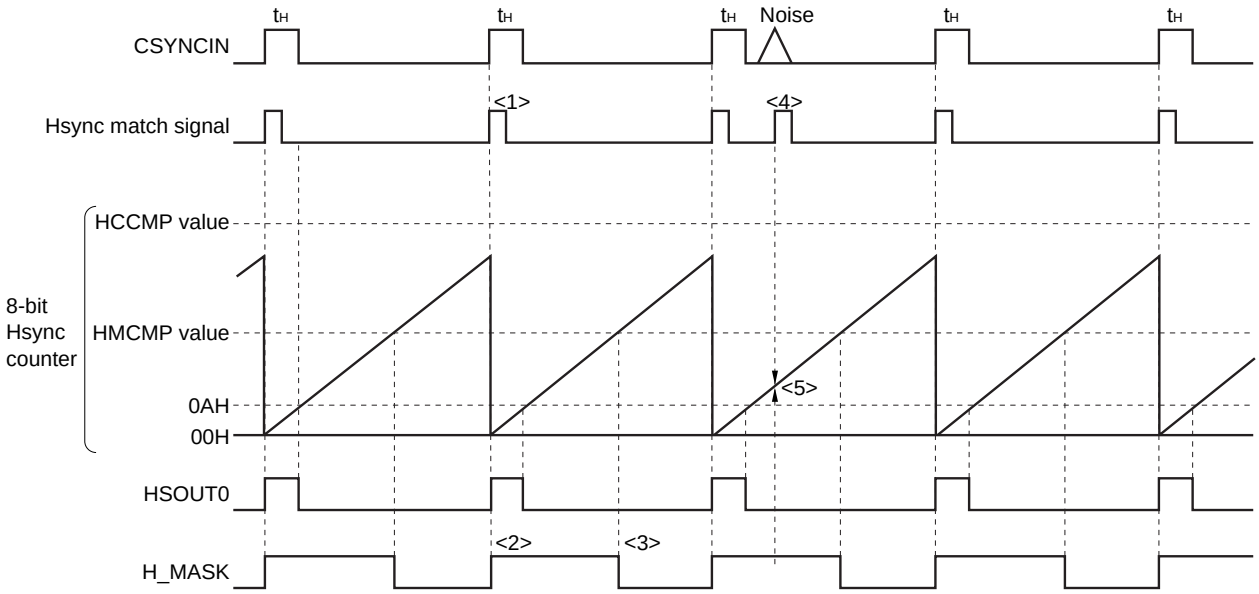
Hsync signal detection delay $\fallingdotseq$ (HSCMP setup value + 2)/f_{sync} + Noise period

2. Normal Hsync separation operation starts when a Vsync signal is detected after the Vsync/Hsync separator is activated (VSEN bit is set to 1).

Remark f_{sync}: Vsync/Hsync separator operating clock frequency

15.4.7 Hsync signal mask operation

Figure 15-9. Hsync Signal Mask Operation

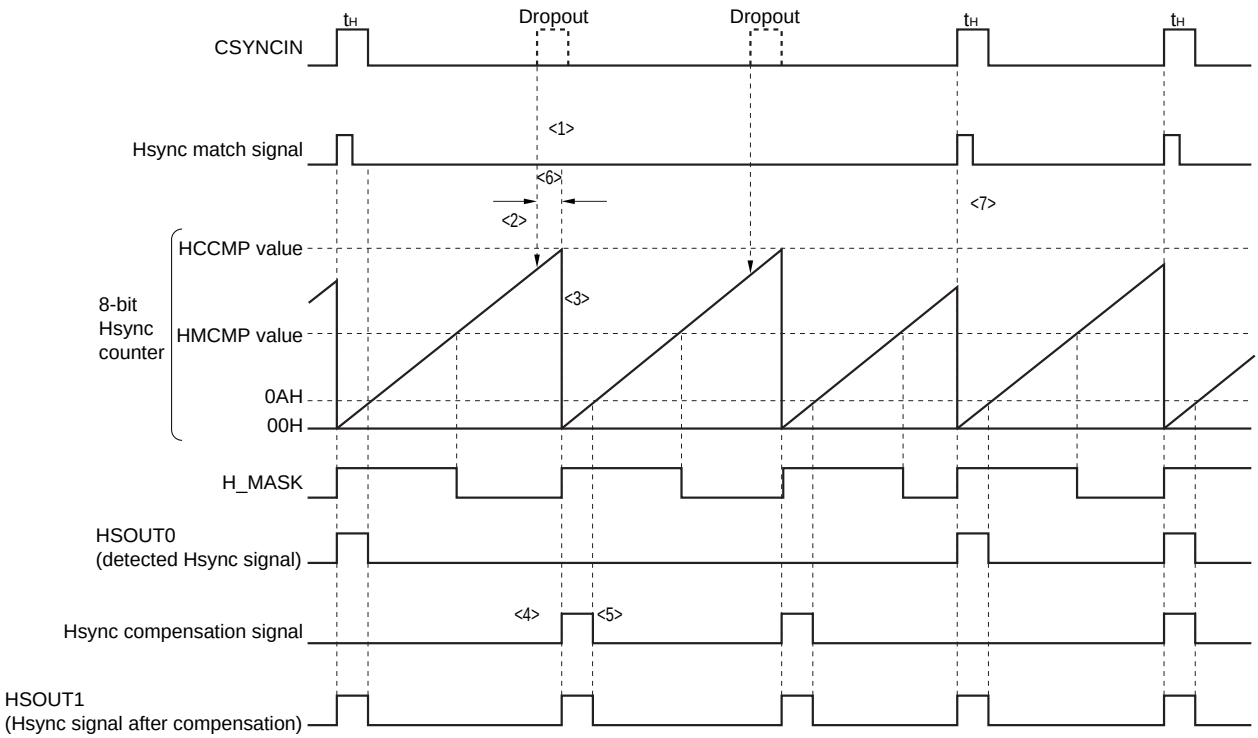


1. After the reset is cancelled, the H_MASK signal is held to 0. When the detection of the Hsync signal subsequently makes the Hsync match signal active (<1>), the 8-bit Hsync counter is cleared and restarted.
2. When the 8-bit Hsync counter is set to 01H, the H_MASK signal is set to 1 (<2>).
3. When the 8-bit Hsync counter counts up to match the contents of the HMCMP register, the H_MASK signal is cleared to 0 (<3>).
4. Even if noise or the like causes the Hsync match signal to be generated (<4>) while the H_MASK signal is held to 1, the 8-bit Hsync counter is not cleared (<5>). The H_MASK signal masks the Hsync match signal.

Caution To perform normal operation, set the Hsync mask width register (HMCMP) to 0AH or a grater value.

15.4.8 Hsync signal self-generation

Figure 15-10. Hsync Signal Self-Generation



1. If the Hsync signal is absent, no Hsync match signal is generated (<1>). Accordingly, the 8-bit Hsync counter is not cleared and continues counting up (<2>).
2. When the contents of the 8-bit Hsync counter match the contents of the Hsync compensation compare register (HCCMP), the counter is cleared and starts counting up again (clear and start) (<3>). Meanwhile, the Hsync compensation signal is set (<4>). The HSOUT0 signal is not set.
3. When the 8-bit Hsync counter is set to 0AH, the Hsync compensation signal is cleared to 0 (<5>).
4. The HSOUT0 signal and Hsync compensation signal are ORed, and the HSOUT1 signal is generated.
5. The HSOUT0 signal is connected to the count clock sources of TM10 and TM11. The degree of absence of the Hsync signal can be monitored by selecting the HSOUT0 signal (Hsync signal before compensation) as the count clock of TM10 or TM11 and reading TM10 or TM11.

Cautions 1. A single compensation operation causes the self-generation signal to fall behind the real Hsync signal by the following period:

About (HCCMP setup value/f_{sync}) – (Horizontal synchronization signal cycle)

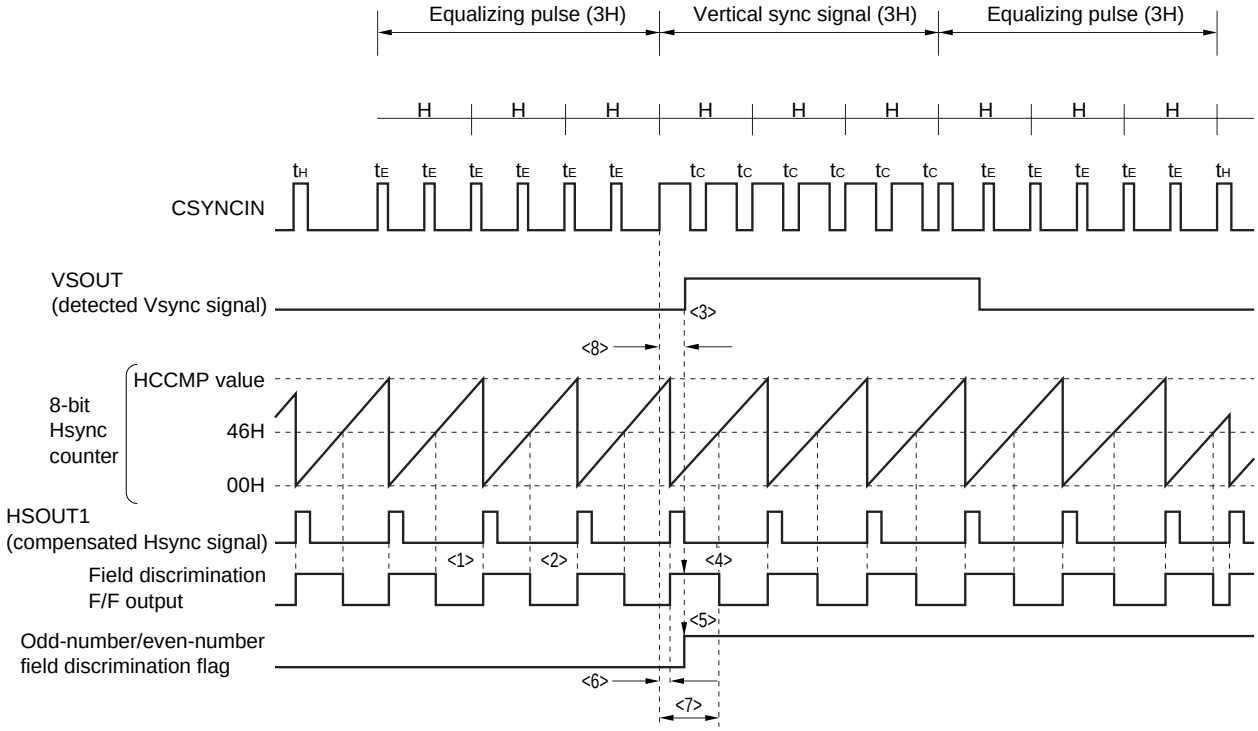
2. Consecutive dropouts of the Hsync signal cause the next Hsync signal (<7>) to encounter the H_MASK signal.

Remark The Hsync signal during the Vsync period is generated through self compensation.

15.4.9 Odd-number/even-number field discrimination

(1) Odd-number field discrimination

Figure 15-11. Odd-Number Field Discrimination



1. Field discrimination F/F is set to 1 by the HSOUT1 signal (self-compensated Hsync signal) (<1>).
2. When the 8-bit Hsync counter is set to 46H after that, the field discrimination F/F is cleared to 0 (<2>).
3. When the detection of the Vsync signal causes the VSOUT signal to be generated (<3>), the output value of the field discrimination F/F is latched at the rising edge of the VSOUT signal (<4>). The latched value becomes the odd-number/even-number field discrimination flag (<5>).
4. With an odd-number field, the odd-number/even-number field discrimination flag is set to 1. The flag can be monitored by the VFLG bit of the Vsync control register (VSC).

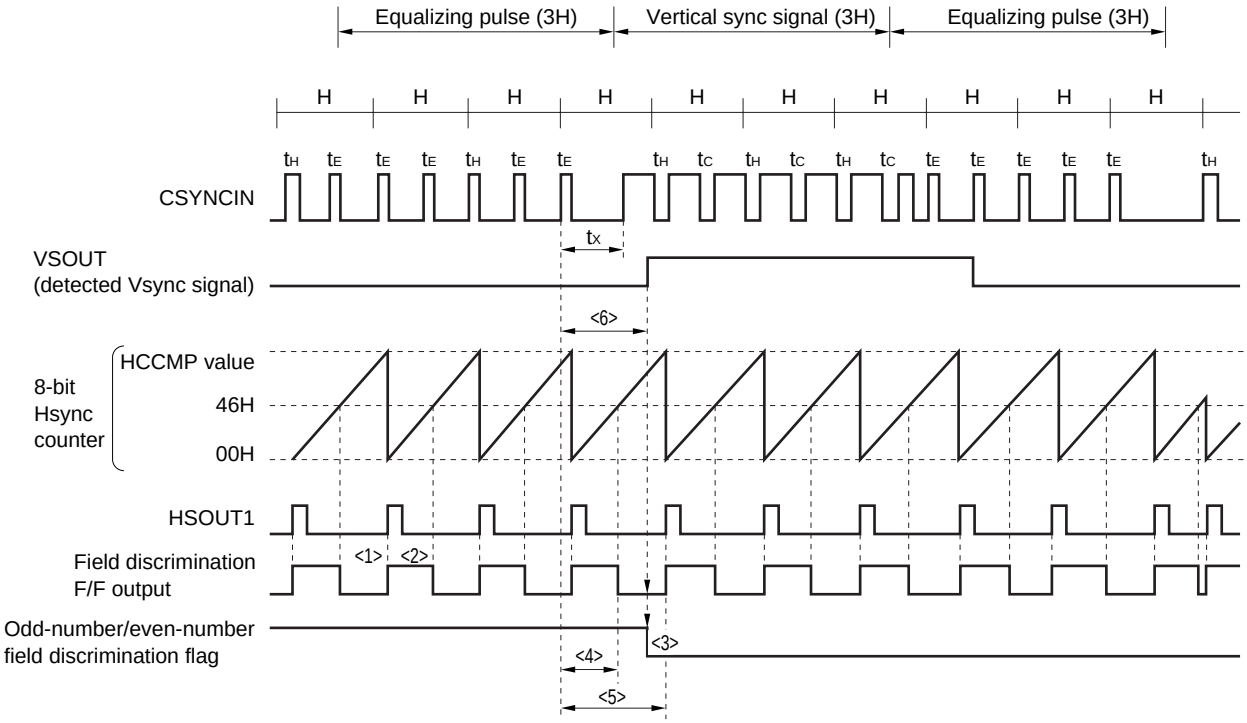
Cautions 1. To determine an odd-number field correctly, the Vsync signal (VSOUT signal) must be detected while the field discrimination F/F output is 1. Periods must be set to satisfy this condition: Period of <6> < Period of <8> < Period of <7>. To enable this, the VSCMP register should be set to a value within the following range, assuming that no noise occurs:

HSCMP + 3 < VSCMP < HSCMP + 141

2. The field discrimination flag starts operating normally after two Vsync signals are detected after the Vsync/Hsync separator is activated (VSEN bit is set to 1).

(2) Even-number field discrimination

Figure 15-12. Even-Number Field Discrimination



1. Field discrimination F/F is set to 1 by the HSOUT1 signal (self-compensated Hsync signal) (<1>).
2. When the 8-bit Hsync counter is subsequently set to 46H, the field discrimination F/F is cleared to 0 (<2>).
3. When the detection of the Vsync signal causes the VSOUT signal to be generated, the output value of the field discrimination F/F is latched at the rising edge of the VSOUT signal. The latched value becomes the odd-number/even-number field discrimination flag (<3>).
4. With an even-number field, the odd-number/even-number field discrimination flag is cleared to 0. The flag can be monitored by the VFLG bit of the Vsync control register (VSC).

Cautions 1. To determine an even-number field correctly, the Vsync signal (VSOUT signal) must be detected while the field discrimination F/F output is 0. Accordingly, periods must be set to satisfy this condition: Period of <4> < Period of <6> < Period of <5>. To enable this, the VSCMP register should be set to a value within the following range, assuming that no noise occurs:

$$HSCMP + 141 < t_x \times f_{sync} + VSCMP < HSCMP + 2 \times HCCMP + 1$$

2. The field discrimination flag starts operating normally after two Vsync signals are detected after the Vsync/Hsync separator is activated (VSEN bit is set to 1).

[MEMO]

CHAPTER 16 PORT FUNCTION

16.1 Port Configuration

The V850/SV1 includes 151 I/O ports from ports 0 to 19 (16 ports are input only).

There are three ports; I/O buffer power supplies, AV_{DD}, BV_{DD}, and V_{DD} that are described below.

Power Supply	Corresponded Pins	Usable Voltage Range
AV_{DD}	Port 7, port 8	2.7 V ≤ AV_{DD} ≤ 3.6 V
BV_{DD}	Port 4, port 5, port 6, port 9, CLKOUT, $\overline{WAIT}$	2.7 V ≤ BV_{DD} ≤ 3.6 V
V_{DD}	Port 0, port 1, port 2, port 3, port 10, port 11, port 12, port 13, port 14, port 15, port 16, port 17, port 18, port 19, $\overline{RESET}$	2.7 V ≤ V_{DD} ≤ 3.6 V

16.2 Port Pin Function

16.2.1 Port 0

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

A pull-up resistor can be connected in 1-bit units (software pull-up function).

When P00 to P04 are used as the NMI and INTP0 to INTP3 pins, noise is eliminated from these pins by an analog noise elimination circuit.

When P05 to P07 are used as the INTP4 to INTP6, ADTRG, and RTPTRG0 pins, noise is eliminated from these pins by a digital noise elimination circuit.

Figure 16-1. Port 0 (P0)

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, 7)
0	Outputs 0
1	Outputs 1

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

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

Port 0 includes the following alternate functions.

Table 16-1. Alternate Function of Port 0

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 0	P00	NMI	I/O	Yes	Analog noise elimination
	P01	INTP0			
	P02	INTP1			
	P03	INTP2			
	P04	INTP3			
	P05	INTP4/ADTRG			Digital noise elimination
	P06	INTP5/RTPTRG0			
	P07	INTP6			

Note Software pull-up function

(1) Function of P0 pins

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

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

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

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

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 0 (PU0).

When a reset is input, the settings are initialized to input mode. Also, the valid edge of each interrupt request becomes invalid (NMI and INTP0 to INTP6 do not function immediately after reset).

(2) Noise elimination

(a) Elimination of noise from NMI and INTP0 to INTP3 pins

An on-chip noise elimination circuit uses analog delay. Consequently, if a signal having a constant level is input for longer than a specified time to these pins, it is detected as a valid edge. Such edge detection occurs after the specified amount of time.

(b) Elimination of noise from INTP4 to INTP6, ADTRG, and RTPTRG0 pins

A digital noise elimination circuit is provided on chip.

This circuit uses digital sampling. A pin's input level is detected using a sampling clock (f_{XX}), and noise elimination is performed for the INTP4, INTP5, ADTRG, and RTPTRG0 pins if the same level is not detected three times consecutively. The noise-elimination width can be changed for the INTP6 pin (refer to **5.3.8 (3) Noise elimination of INTP6 pin**).

Cautions 1. If the input pulse width is 2 or 3 clock, whether it will be detected as a valid edge or eliminated as noise is undefined.

2. To ensure correct detection of pulses as pulses, constant-level input is required for 3 clocks or more.

3. If noise is occurring in synchronization with the sampling clock, it may not be recognized as noise. In such cases, attach a filter to the input pins to eliminate the noise.

4. Noise elimination is not performed when these pins are used as an ordinary input port.

(3) Control registers

(a) Port 0 mode register (PM0)

PM0 can be read/written in 1-bit or 8-bit units.

Figure 16-2. Port 0 Mode Register (PM0)

After reset: FFH R/W Address: FFFFF020H

	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

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

PU0 can be read/written in 1-bit or 8-bit units.

Figure 16-3. Pull-Up Resistor Option Register 0 (PU0)

After reset: 00H R/W Address: FFFFF0A0H

	7	6	5	4	3	2	1	0
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00

PU0n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(c) Rising edge specification register 0 (EGP0)

EGP0 can be read/written in 1-bit or 8-bit units.

Figure 16-4. Rising Edge Specification Register 0 (EGP0)

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

(d) Falling edge specification register 0 (EGN0)

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

Figure 16-5. Falling Edge Specification Register 0 (EGN0)

After reset: 00H R/W Address: FFFFF0C2H

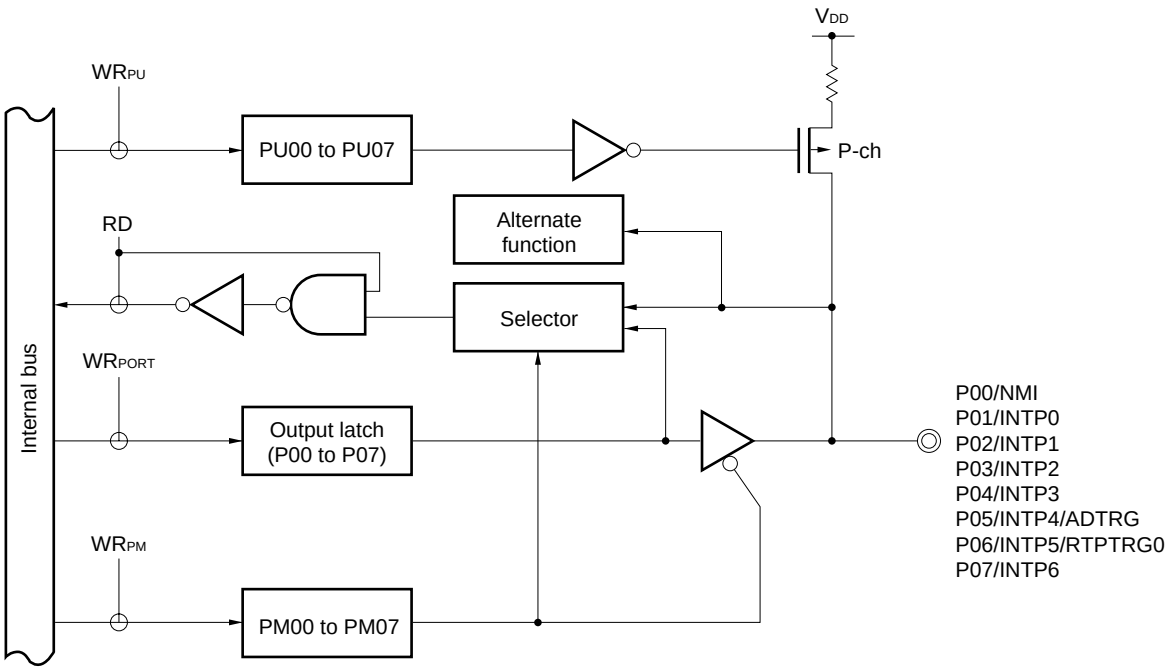
	7	6	5	4	3	2	1	0
EGN0	EGN07	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00

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 of port 0

Figure 16-6. Block Diagram of P00 to P07



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 0

WR: Write signal of port 0

16.2.2 Port 1

Port 1 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units.
A pull-up resistor can be connected in 1-bit units (software pull-up function).
P10 to P12, P14, and P15 are selectable as normal outputs or N-ch open-drain outputs.

Figure 16-7. Port 1 (P1)

After reset: 00H R/W Address: FFFFF002H

	7	6	5	4	3	2	1	0
P1	0	0	P15	P14	P13	P12	P11	P10

P1n	Control of Output Data (in Output Mode) (n = 0 to 5)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P1 register is read, the pin levels at that time are read. Writing to P1 writes the values to that register. This does not affect the input pins.
In output mode: When the P1 register is read, the P1 register's values are read. Writing to P1 writes the values to that register, and those values are immediately output.

Port 1 includes the following alternate functions. SDA0 and SCL0 pins are available only in the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

Table 16-2. Alternate Function of Port 1

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 1	P10	SI0/SDA0	I/O	Yes	Selectable as N-ch open-drain output
	P11	SO0			
	P12	$\overline{\text{SCK0}}$/SCL0			
	P13	SI1/RXD0			–
	P14	SO1/TXD0			Selectable as N-ch open-drain output
	P15	$\overline{\text{SCK1}}$/ASCK0			

Note Software pull-up function

(1) Function of P1 pins

Port 1 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 1 mode register (PM1).

In output mode, the values set to each bit are output to the port 1 register (P1). The port 1 function register (PF1) can be used to specify whether P10 to P12, P14, and P15 are normal outputs or N-ch open-drain outputs.

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

A pull-up resistor can be connected in 1-bit units when specified via the pull-up resistor option register 1 (PU1).

Clear the P1 and PM1 registers to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

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

(2) Control registers

(a) Port 1 mode register (PM1)

PM1 can be read/written in 1-bit or 8-bit units.

Figure 16-8. Port 1 Mode Register (PM1)

After reset: 3FH R/W Address: FFFFF022H

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

PM1n	Control of I/O Mode (n = 0 to 5)
0	Output mode
1	Input mode

(b) Pull-up resistor option register 1 (PU1)

PU1 can be read/written in 1-bit or 8-bit units.

Figure 16-9. Pull-Up Resistor Option Register 1 (PU1)

After reset: 00H R/W Address: FFFFF0A2H

	7	6	5	4	3	2	1	0
PU1	0	0	PU15	PU14	PU13	PU12	PU11	PU10

PU1n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 5)
0	Do not connect
1	Connect

(c) Port 1 function register (PF1)

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

Figure 16-10. Port 1 Function Register (PF1)

After reset: 00H R/W Address: FFFFF0B0H

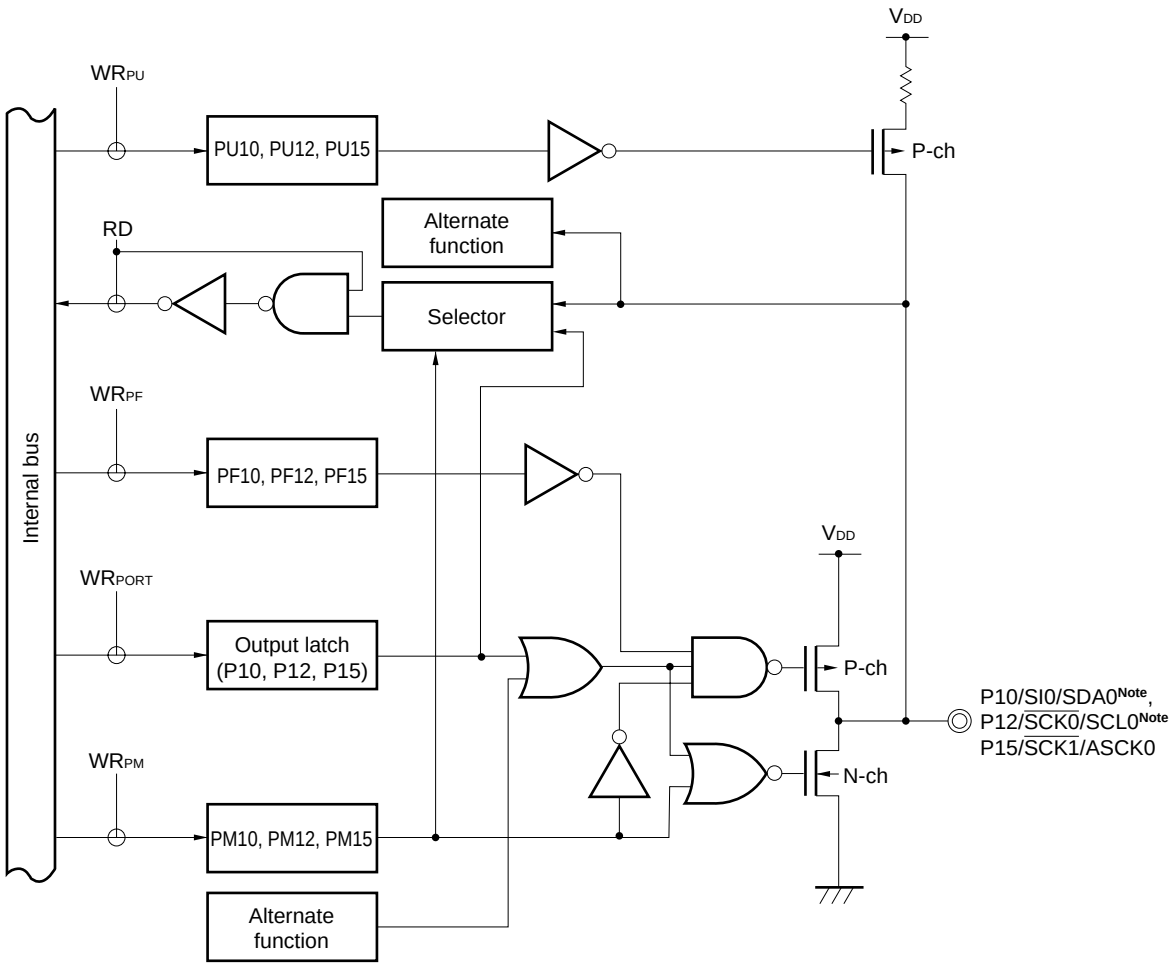
	7	6	5	4	3	2	1	0
PF1	0	0	PF15	PF14	0[Note]	PF12	PF11	PF10

PF1n	Control of Normal Output/N-ch Open-Drain Output (n = 0 to 2, 4, 5)
0	Normal output
1	N-ch open-drain output

Note Bit 3 is fixed as a normal output.

(3) Block diagram of port 1

Figure 16-11. Block Diagram of P10, P12, and P15



Note SDA0 and SCL0 pins are available only in the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

PU: Pull-up resistor option register

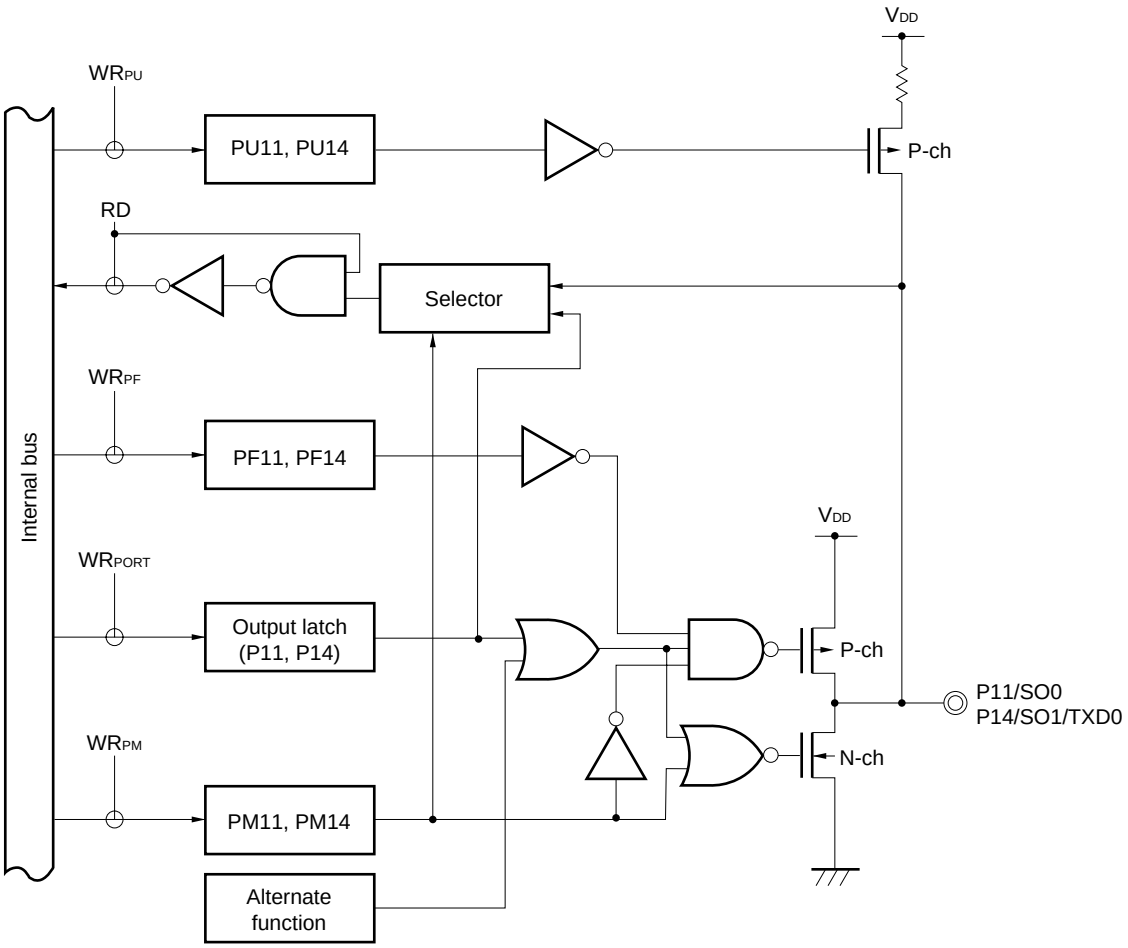
PF: Port function control register

PM: Port mode register

RD: Read signal of port 1

WR: Write signal of port 1

Figure 16-12. Block Diagram of P11 and P14



PU: Pull-up resistor option register

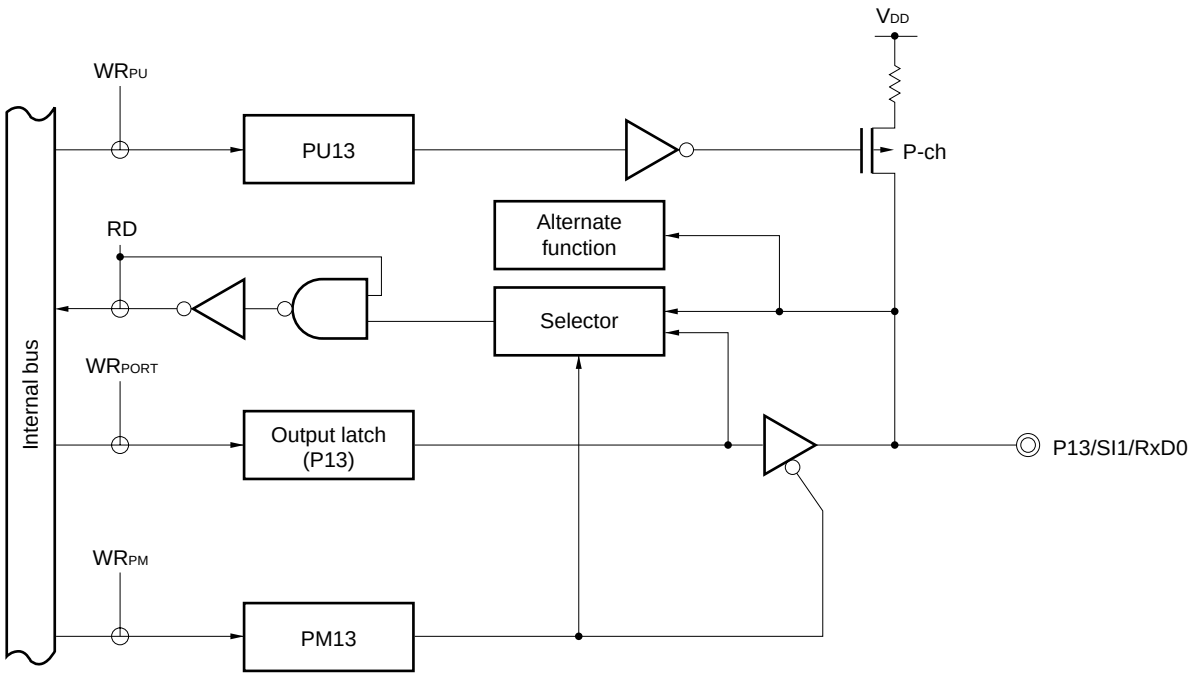
PF: Port function control register

PM: Port mode register

RD: Read signal of port 1

WR: Write signal of port 1

Figure 16-13. Block Diagram of P13



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 1

WR: Write signal of port 1

16.2.3 Port 2

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

A pull-up resistor can be connected in 1-bit units (software pull-up function).

P20, P21, P22, P24, and P25 are selectable as normal outputs or N-ch open-drain outputs.

When P26 and P27 are used as TI2 and TI3 pins, noise is eliminated from these pins by a digital noise elimination circuit.

Figure 16-14. Port 2 (P2)

After reset: 00H R/W Address: FFFFF004H

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23	P22	P21	P20

P2n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P2 register is read, the pin levels at that time are read. Writing to P2 writes the values to that register. This does not affect the input pins.

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

Port 2 includes the following alternate functions. SDA1 and SCL1 are available only in the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

Table 16-3. Alternate Function of Port 2

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 2	P20	SI2/SDA1	I/O	Yes	Selectable as N-ch open-drain output
	P21	SO2			
	P22	$\overline{\text{SCK2}}$/SCL1			
	P23	SI3/RXD1			–
	P24	SO3/TXD1			Selectable as N-ch open-drain output
	P25	$\overline{\text{SCK3}}$/ASCK1			
	P26	TI2/TO2			Digital noise elimination
	P27	TI3/TO3			

Note Software pull-up function

(1) Function of P2 pins

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

In output mode, the values set to each bit are output to the port 2 register (P2). The port 2 function register (PF2) can be used to specify whether P20, P21, P22, P24, and P25 are normal outputs or N-ch open-drain outputs.

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

A pull-up resistor can be connected in 1-bit units when specified via the pull-up resistor option register 2 (PU2).

When using the alternate function as TI2 and TI3 pins, noise elimination is provided by a digital noise elimination circuit (same as digital noise elimination circuit for port 0).

Clear the P2 and PM2 registers to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

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

(2) Control registers

(a) Port 2 mode register (PM2)

PM2 can be read/written in 1-bit or 8-bit units.

Figure 16-15. Port 2 Mode Register (PM2)

After reset: FFH R/W Address: FFFFF024H

	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

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

PU2 can be read/written in 1-bit or 8-bit units.

Figure 16-16. Pull-Up Resistor Option Register 2 (PU2)

After reset: 00H R/W Address: FFFFF0A4H

	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

PU2n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(c) Port 2 function register (PF2)

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

Figure 16-17. Port 2 Function Register (PF2)

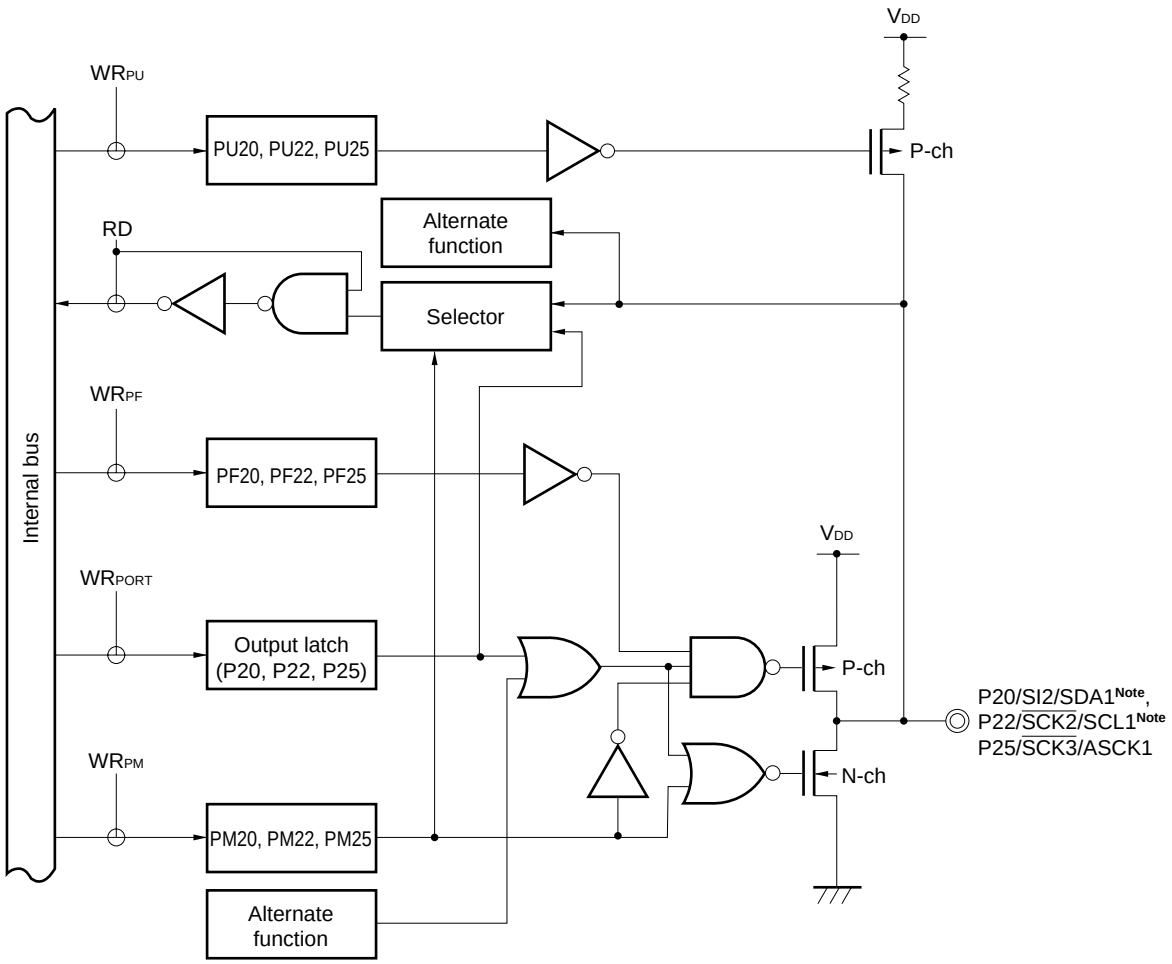
After reset: 00H R/W Address: FFFFF0B2H

	7	6	5	4	3	2	1	0
PF2	0	0	PF25	PF24	0	PF22	PF21	PF20

PF2n	Control of Normal Output/N-ch Open-Drain Output (n = 0 to 2, 4, 5)
0	Normal output
1	N-ch open-drain output

(3) Block diagram of port 2

Figure 16-18. Block Diagram of P20, P22, and P25



Note SDA1 and SCL1 pins are available only in the μPD703039Y, 703040Y, 703041Y, and 70F3040Y.

PU: Pull-up resistor option register

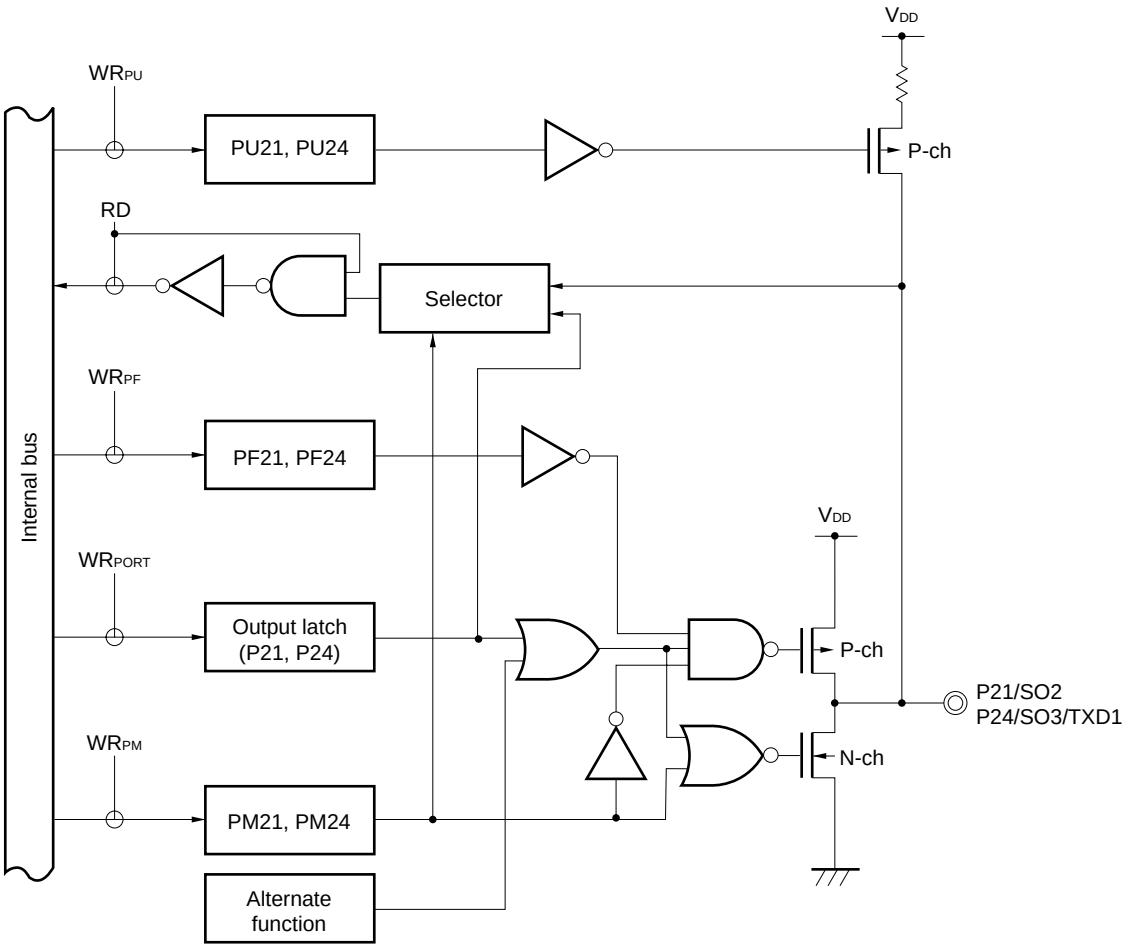
PF: Port function control register

PM: Port mode register

RD: Read signal of port 2

WR: Write signal of port 2

Figure 16-19. Block Diagram of P21 and P24



PU: Pull-up resistor option register

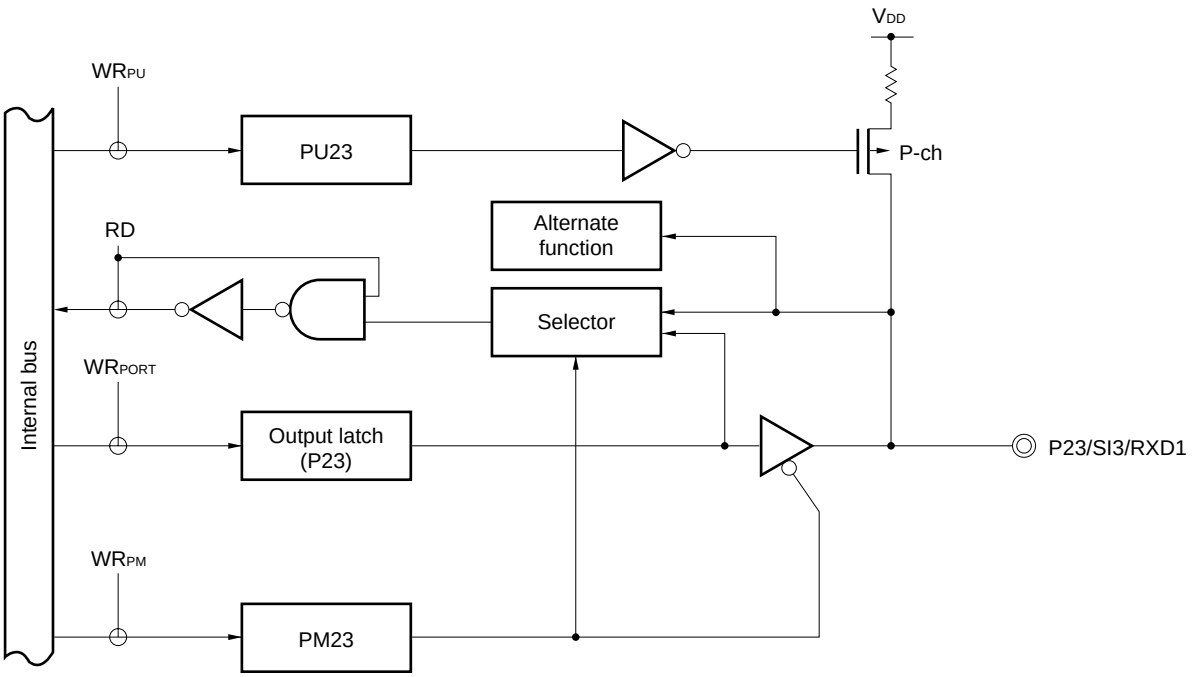
PF: Port function control register

PM: Port mode register

RD: Read signal of port 2

WR: Write signal of port 2

Figure 16-20. Block Diagram of P23



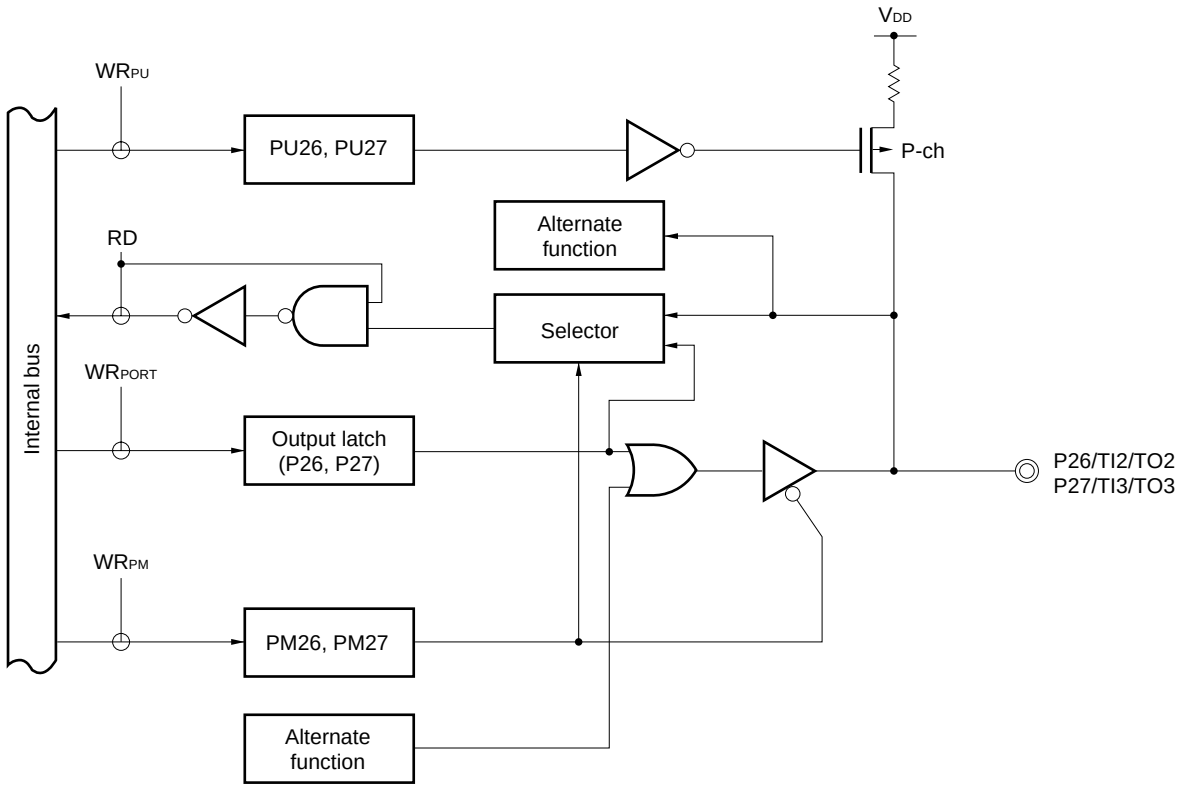
PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 2

WR: Write signal of port 2

Figure 16-21. Block Diagram of P26 and P27



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 2

WR: Write signal of port 2

16.2.4 Port 3

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

A pull-up resistor can be connected in 1-bit units (software pull-up function).

When P36 and P37 are used as the TI4 and TI5 pins, noise is eliminated from these pins by a digital noise elimination circuit.

Figure 16-22. Port 3 (P3)

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 the P3 register is read, the pin levels at that time are read. Writing to P3 writes the values to that register. This does not affect the input pins.

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

Port 3 includes the following alternate functions.

Table 16-4. Alternate Function of Port 3

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 3	P30	TI000	I/O	Yes	–
	P31	TI001			
	P32	TI010			
	P33	TI011			
	P34	TO0			
	P35	TO1			
	P36	TI4/TO4			Digital noise elimination
	P37	TI5/TO5			

Note Software pull-up function

(1) Function of P3 pins

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

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

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

A pull-up resistor can be connected in 1-bit units when specified via the pull-up resistor option register 3 (PU3).

When using the alternate function as TI4 and TI5 pins, noise elimination is provided by a digital noise elimination circuit (same as digital noise elimination circuit for port 0).

Clear the P3 and PM3 registers to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

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

(2) Control registers

(a) Port 3 mode register (PM3)

PM3 can be read/written in 1-bit or 8-bit units.

Figure 16-23. Port 3 Mode Register (PM3)

After reset: FFH R/W Address: FFFFF026H

	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

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

PU3 can be read/written in 1-bit or 8-bit units.

Figure 16-24. Pull-Up Resistor Option Register 3 (PU3)

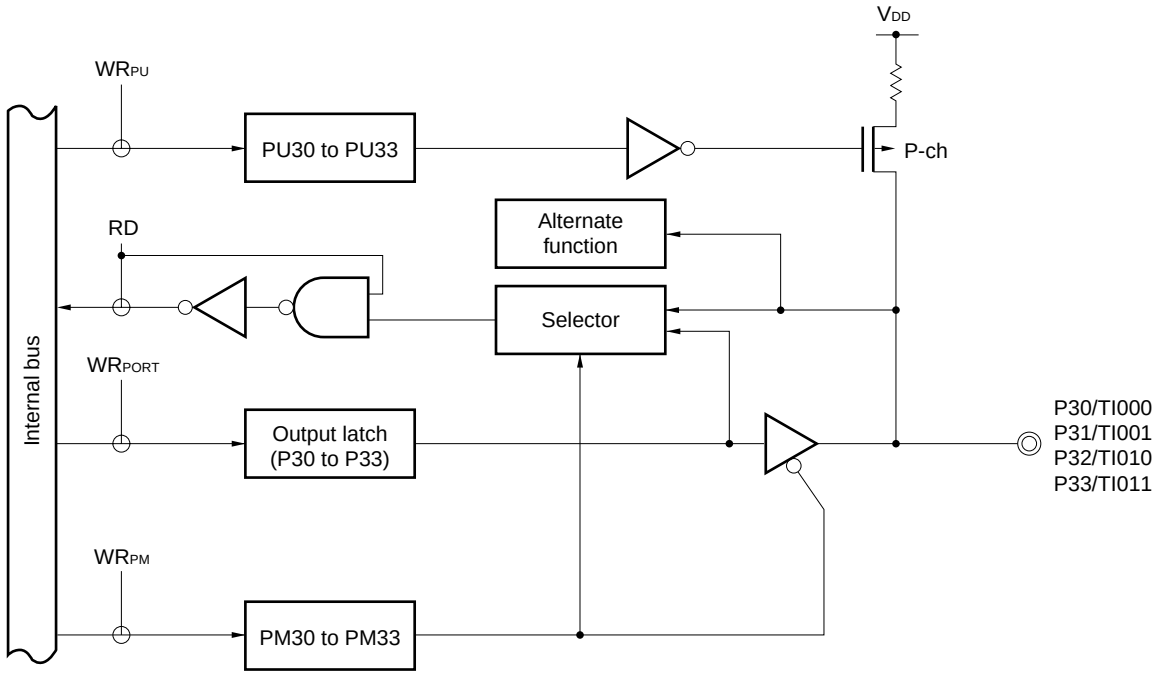
After reset: 00H R/W Address: FFFFF0A6H

	7	6	5	4	3	2	1	0
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(3) Block diagram of port 3

Figure 16-25. Block Diagram of P30 to P33



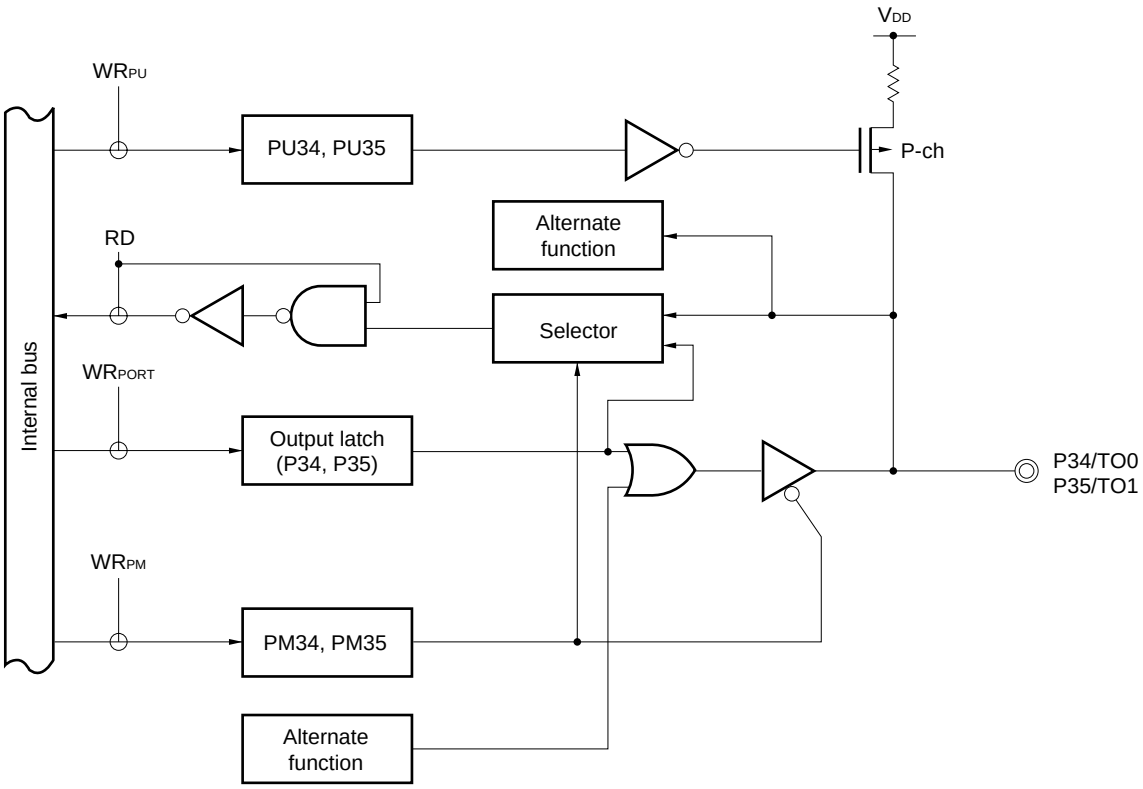
PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 3

WR: Write signal of port 3

Figure 16-26. Block Diagram of P34 and P35



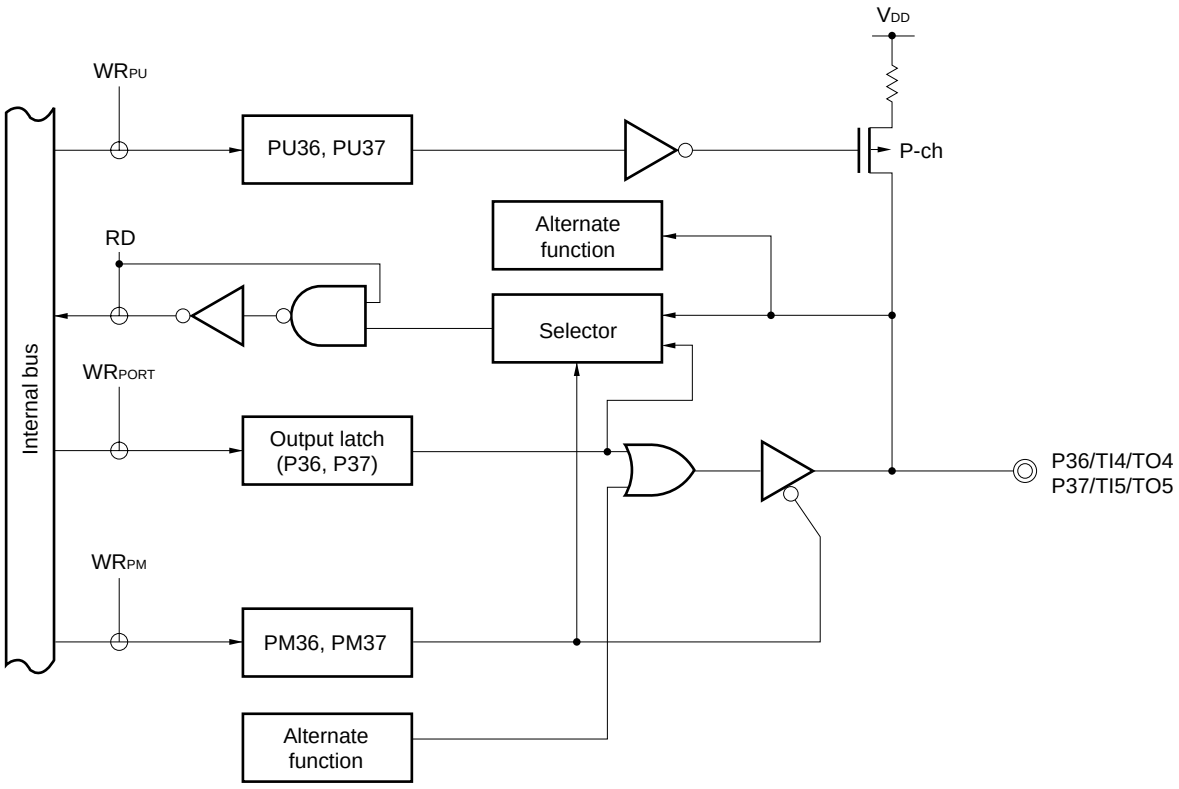
PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 3

WR: Write signal of port 3

Figure 16-27. Block Diagram of P36 and P37



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 3

WR: Write signal of port 3

16.2.5 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.

Figure 16-28. Ports 4 and 5 (P4 and P5)

After reset: 00H R/W Address: FFFFF008H, FFFFF00AH

	7	6	5	4	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnx	Control of Output Data (in Output Mode) (n = 4, 5, x = 0 to 7)
0	Outputs 0
1	Outputs 1

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

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

Ports 4 and 5 include the following alternate functions.

Table 16-5. Alternate Function 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 the port registers (P4 and P5).

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

A software pull-up function is not implemented.

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

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

(2) Control register

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

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

Figure 16-29. Port 4 Mode Register, Port 5 Mode Register (PM4, PM5)

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

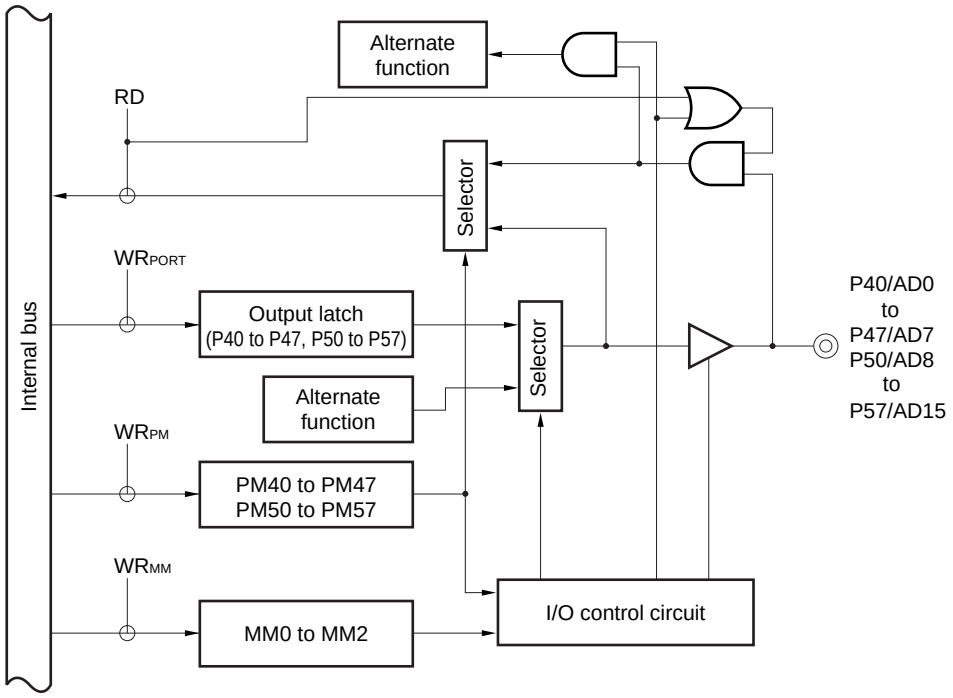
	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

(n = 4, 5)

PMnx	Control of I/O Mode (n = 4, 5, x = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagram of ports 4 and 5

Figure 16-30. Block Diagram of P40 to P47 and P50 to P57



PM: Port mode register

MM: Memory expansion mode register

RD: Read signal of ports 4 and 5

WR: Write signal of ports 4 and 5

16.2.6 Port 6

Port 6 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units.

Figure 16-31. Port 6 (P6)

After reset: 00H R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
P6	0	0	P65	P64	P63	P62	P61	P60

P6n	Control of Output Data (in Output Mode) (n = 0 to 5)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P6 register is read, the pin levels at that time are read. Writing to P6 writes the values to that register. This does not affect the input pins.

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

Port 6 includes the following alternate functions.

Table 16-6. Alternate Function of Port 6

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 6	P60	A16	I/O	No	–
	P61	A17			
	P62	A18			
	P63	A19			
	P64	A20			
	P65	A21			

Note Software pull-up function

(1) Function of P6 pins

Port 6 is a 6-bit I/O port for which I/O settings can be controlled in 1-bit units. I/O settings are controlled via the port 6 mode register (PM6).

In output mode, the values set to each bit are output to the port register (P6).

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

A software pull-up function is not implemented.

When using the alternate function as A16 to A21, set the pin functions via the memory expansion register (MM). This does not affect the PM6 register.

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

(2) Control register

(a) Port 6 mode register (PM6)

PM6 can be read/written in 1-bit or 8-bit units.

Figure 16-32. Port 6 Mode Register (PM6)

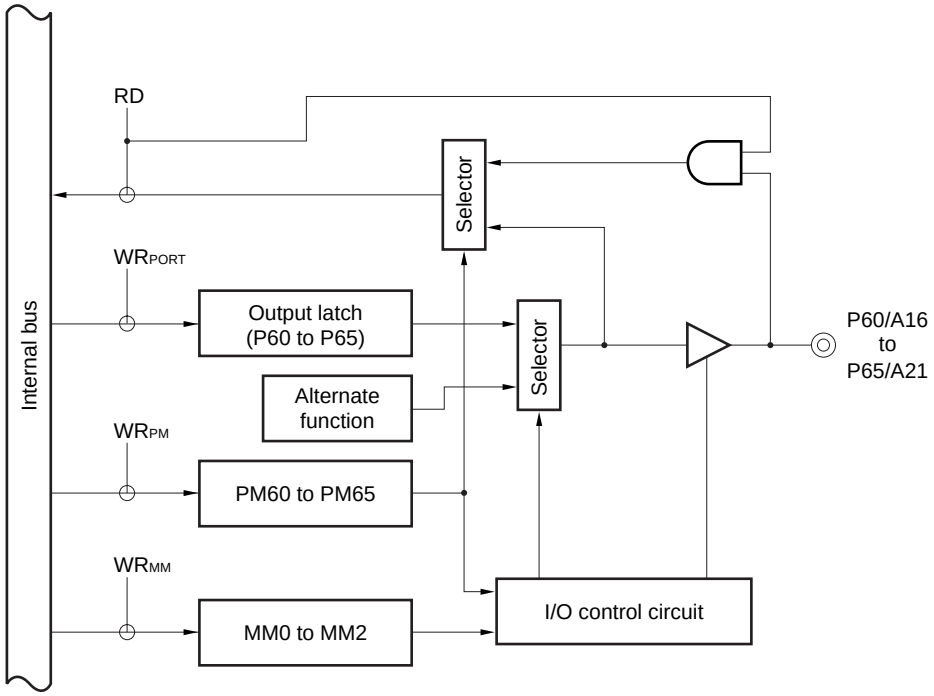
After reset: 3FH R/W Address: FFFFF02CH

	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 of port 6

Figure 16-33. Block Diagram of P60 to P65



PM: Port mode register

MM: Memory expansion mode register

RD: Read signal of port 6

WR: Write signal of port 6

16.2.7 Ports 7 and 8

Ports 7 and 8 are 8-bit input ports. Both ports are read-only and are accessible in 8-bit units or in 1-bit units.

Figure 16-34. Ports 7 and 8 (P7 and P8)

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	P87	P86	P85	P84	P83	P82	P81	P80

P8n	Pin Level (n = 0 to 7)
0/1	Read pin level of bit n

Ports 7 and 8 include the following alternate functions.

Table 16-7. Alternate Function 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			
	P84	ANI12			
	P85	ANI13			
	P86	ANI14			
	P87	ANI15			

Note Software pull-up function

(1) Functions of P7 and P8 pins

Ports 7 and 8 are 8-bit input-only ports.

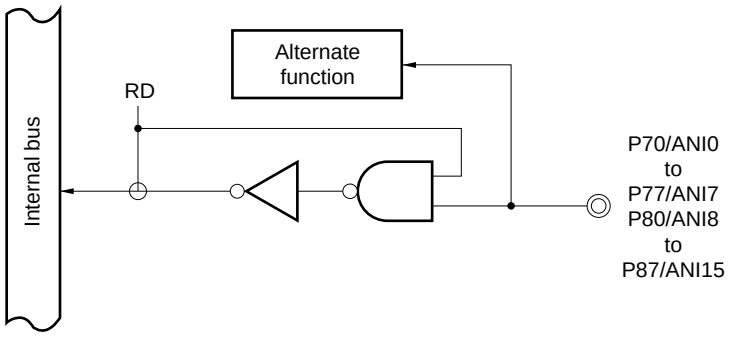
The pin statuses can be read by reading the port registers (P7 and P8). Data cannot be written to P7 or P8.

A software pull-up function is not implemented.

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

(2) Block diagram of ports 7 and 8

Figure 16-35. Block Diagram of P70 to P77 and P80 to P87



RD: Read signal of ports 7 and 8

16.2.8 Port 9

Port 9 is a 7-bit I/O port for which I/O settings can be controlled in 1-bit units.

Figure 16-36. Port 9 (P9)

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 the P9 register is read, the pin levels at that time are read. Writing to P9 writes the values to that register. This does not affect the input pins.

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

Port 9 includes the following alternate functions.

Table 16-8. Alternate Function of Port 9

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

Note Software pull-up function

(1) Function of P9 pins

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

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

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

A software pull-up function is not implemented.

When using the P9 for control signals in expansion mode, set the pin functions via the memory expansion mode register (MM).

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

(2) Control register

(a) Port 9 mode register (PM9)

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

Figure 16-37. Port 9 Mode Register (PM9)

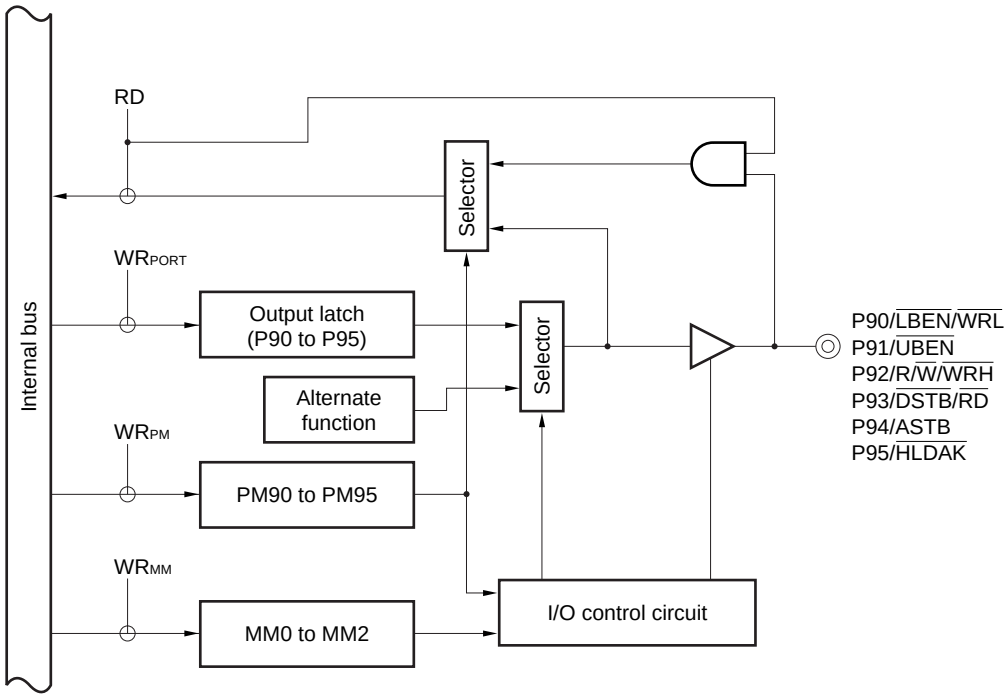
After reset: 7FH R/W Address: FFFFF032H

	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 of port 9

Figure 16-38. Block Diagram of P90 to P95



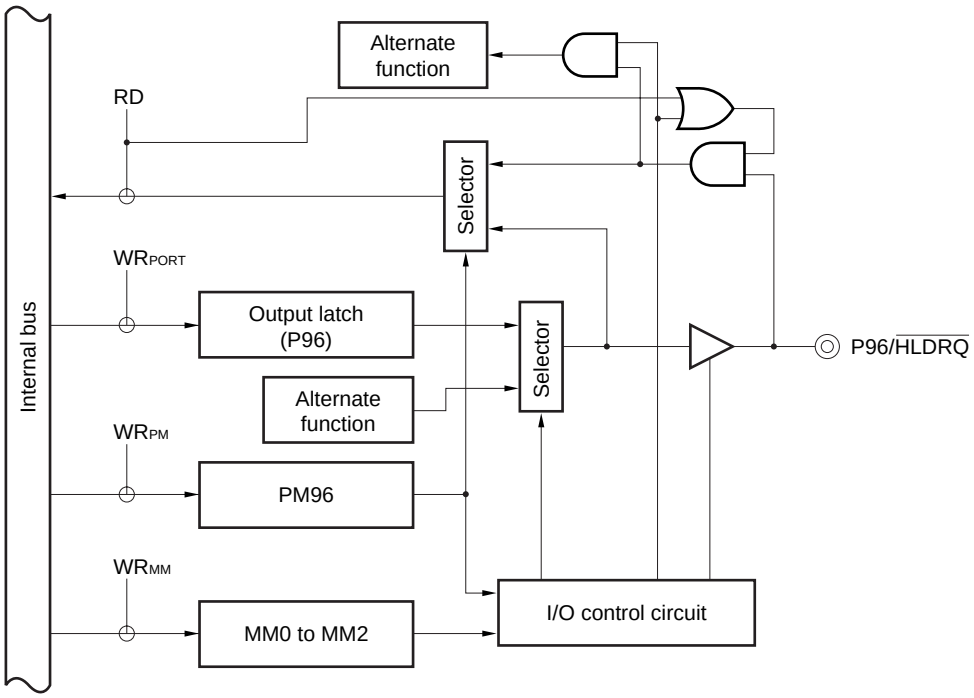
PM: Port mode register

MM: Memory expansion mode register

RD: Read signal of port 9

WR: Write signal of port 9

Figure 16-39. Block Diagram of P96



PM: Port mode register

MM: Memory expansion mode register

RD: Read signal of port 9

WR: Write signal of port 9

16.2.9 Port 10

Port 10 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units.
A pull-up resistor can be connected in 1-bit units (software pull-up function).
The pins in this port are selectable as normal outputs or N-ch open-drain outputs.

Figure 16-40. Port 10 (P10)

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 the P10 register is read, the pin levels at that time are read. Writing to P10 writes the values to that register. This does not affect the input pins.
In output mode: When the P10 register is read, the P10 register's values are read. Writing to P10 writes the values to that register, and those values are immediately output.

Port 10 includes the following alternate functions.

Table 16-9. Alternate Function of Port 10

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 10	P100	RTP00	I/O	Yes	Selectable as N-ch open-drain outputs
	P101	RTP01			
	P102	RTP02			
	P103	RTP03			
	P104	RTP04			
	P105	RTP05			
	P106	RTP06			
	P107	RTP07			

Note Software pull-up function

(1) Function of P10 pins

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

In output mode, the values set to each bit are output to the port register (P10). The port 10 function register (PF10) can be used to specify whether outputs are normal outputs or N-ch open-drain outputs.

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

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

Clear the P10 and PM10 registers to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

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

(2) Control register

(a) Port 10 mode register (PM10)

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

Figure 16-41. Port 10 Mode Register (PM10)

After reset: FFH R/W Address: FFFFF034H

	7	6	5	4	3	2	1	0
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100

PM10n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

(b) Pull-up resistor option register 10 (PU10)

PU10 can be read/written in 1-bit or 8-bit units.

Figure 16-42. Pull-Up Resistor Option Register 10 (PU10)

After reset: 00H R/W Address: FFFFF0A8H

	7	6	5	4	3	2	1	0
PU10	PU107	PU106	PU105	PU104	PU103	PU102	PU101	PU100

PU10n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(c) Port 10 function register (PF10)

PF10 can be read/written in 1-bit or 8-bit units.

Figure 16-43. Port 10 Function Register (PF10)

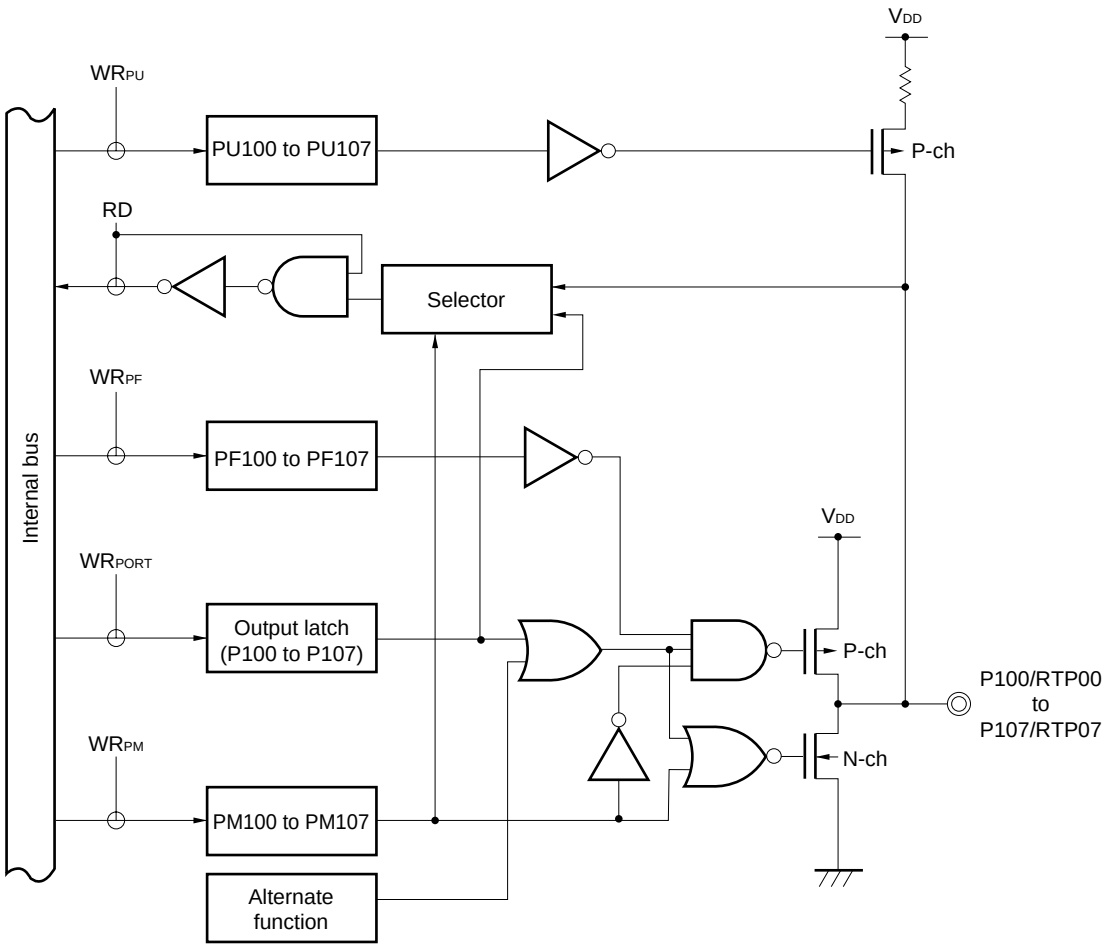
After reset: 00H R/W Address: FFFFF0B4H

	7	6	5	4	3	2	1	0
PF10	PF107	PF106	PF105	PF104	PF103	PF102	PF101	PF100

PF10n	Control of Normal Output/N-ch Open-Drain Output (n = 0 to 7)
0	Normal output
1	N-ch open-drain output

(3) Block diagram of port 10

Figure 16-44. Block Diagram of P100 to P107



PU: Pull-up resistor option register

PF: Port function control register

PM: Port mode register

RD: Read signal of port 10

WR: Write signal of port 10

16.2.10 Port 11

Port 11 is a 4-bit I/O port.
P11 can be read/written in 1-bit or 8-bit units.

Figure 16-45. Port 11 (P11)

After reset: 00H R/W Address: FFFFF016H

	7	6	5	4	3	2	1	0
P11	0	0	0	0	P113	P112	P111	P110

P11n	Control of Output Data (in Output Mode) (n = 0 to 3)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P11 register is read, the pin levels at that time are read. Writing to P11 writes the values to that register. This does not affect the input pins.
In output mode: When the P11 register is read, the P11 register's values are read. Writing to P11 writes the values to that register, and those values are immediately output.

Port 11 has no alternate functions.

Table 16-10. Port 11 (No Alternate Functions)

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 11	P110	–	I/O	No	–
	P111	–			
	P112	–			
	P113	–			

Note Software pull-up function

(1) Function of P11 pins

Port 11 is a 4-bit (total) port for which I/O settings can be controlled in 1-bit units.

In output mode, the values set to each bit (bit 0 to bit 3) are output to the port register (P11).

When using this port in input mode, the pin statuses can be read by reading the P11 register. Also, the P11 register (output latch) values can be read by reading the P11 register while in output mode (bit 0 to bit 3 only).

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

(2) Control register

(a) Port 11 mode register (PM11)

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

Figure 16-46. Port 11 Mode Register (PM11)

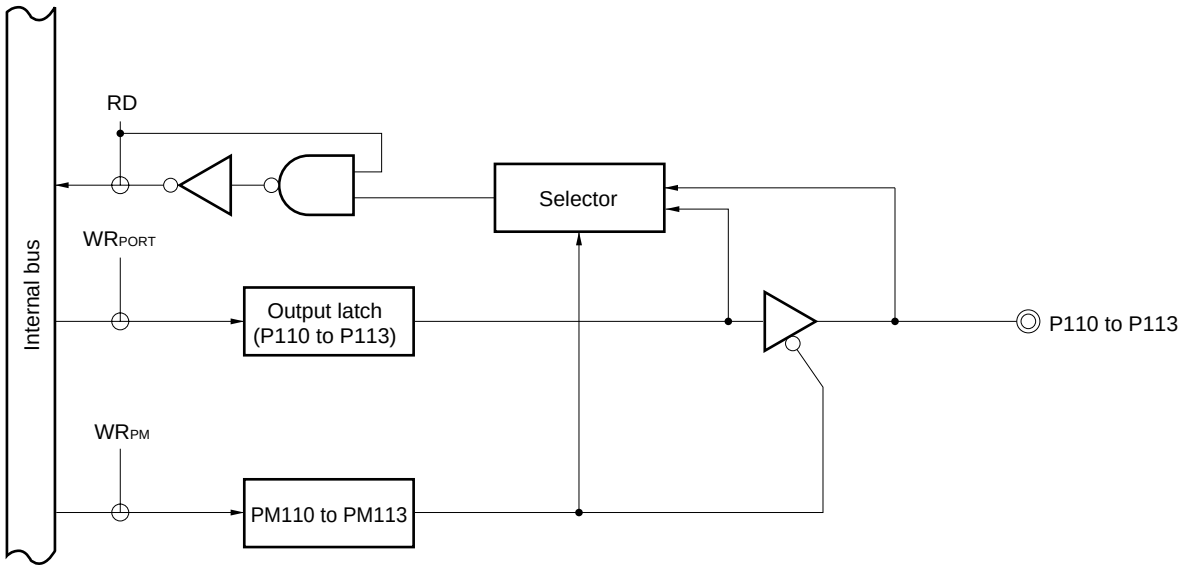
After reset: 1FH R/W Address: FFFFF036H

	7	6	5	4	3	2	1	0
PM11	0	0	0	1	PM113	PM112	PM111	PM110

PM11n	Control of I/O Mode (n = 0 to 3)
0	Output mode
1	Input mode

(3) Block diagram of port 11

Figure 16-47. Block Diagram of P110 to P113



PM: Port mode register

RD: Read signal of port 11

WR: Write signal of port 11

16.2.11 Port 12

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

P121 and P122 are selectable as the normal outputs or N-ch open-drain outputs.

When P124 to P127 are used as the TI6, TI7, TI10, and TI11 pins, noise is eliminated from these pins by a digital noise elimination circuit.

Figure 16-48. Port 12 (P12)

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 the P12 register is read, the pin levels at that time are read. Writing to P12 writes the values to that register. This does not affect the input pins.

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

Port 12 includes the following alternate functions.

Table 16-11. Alternate Function of Port 12

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 12	P120	SI4	I/O	No	–
	P121	SO4			Selectable as N-ch open-drain output
	P122	$\overline{\text{SCK4}}$			
	P123	CLO			–
	P124	TI6/TO6			Digital noise elimination
	P125	TI7/TO7			
	P126	TI10/TO10			
	P127	TI11/TO11			

Note Software pull-up function

(1) Function of P12 pins

Port 12 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 12 mode register (PM12).

In output mode, the value set for each bit is output to the port 12 register (P12). The port 12 function register (PF12) can be used to specify whether P121 and P122 are normal outputs or N-ch open-drain outputs.

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

When using the alternate function as the TI6, TI7, TI10, and TI11 pins, noise elimination is provided by a digital noise elimination circuit (same as the digital noise elimination circuit for port 0).

Clear the P12 and PM12 registers to 0 when using the alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

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

(2) Control registers

(a) Port 12 mode register (PM12)

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

Figure 16-49. Port 12 Mode Register (PM12)

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 12 function register (PF12)

PF12 can be read/written in 1-bit or 8-bit units.

Figure 16-50. Port 12 Function Register (PF12)

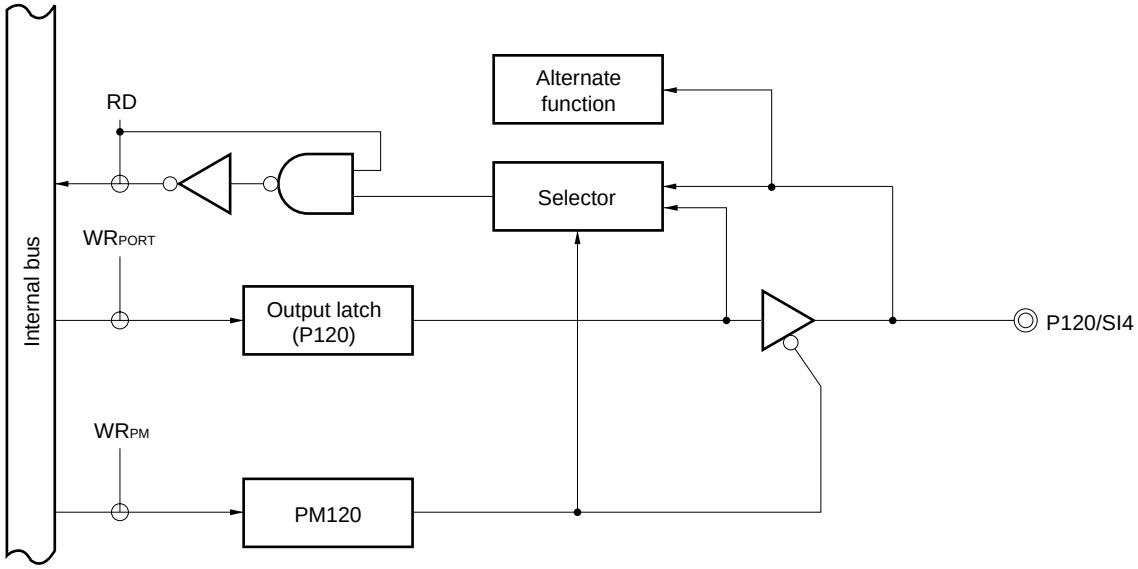
After reset: 00H R/W Address: FFFFF0B6H

	7	6	5	4	3	2	1	0
PF12	0	0	0	0	0	PF122	PF121	0

PF12n	Control of Normal Output/N-ch Open-Drain Output (n = 1, 2)
0	Normal output
1	N-ch open-drain output

(3) Block diagrams of port 12

Figure 16-51. Block Diagram of P120

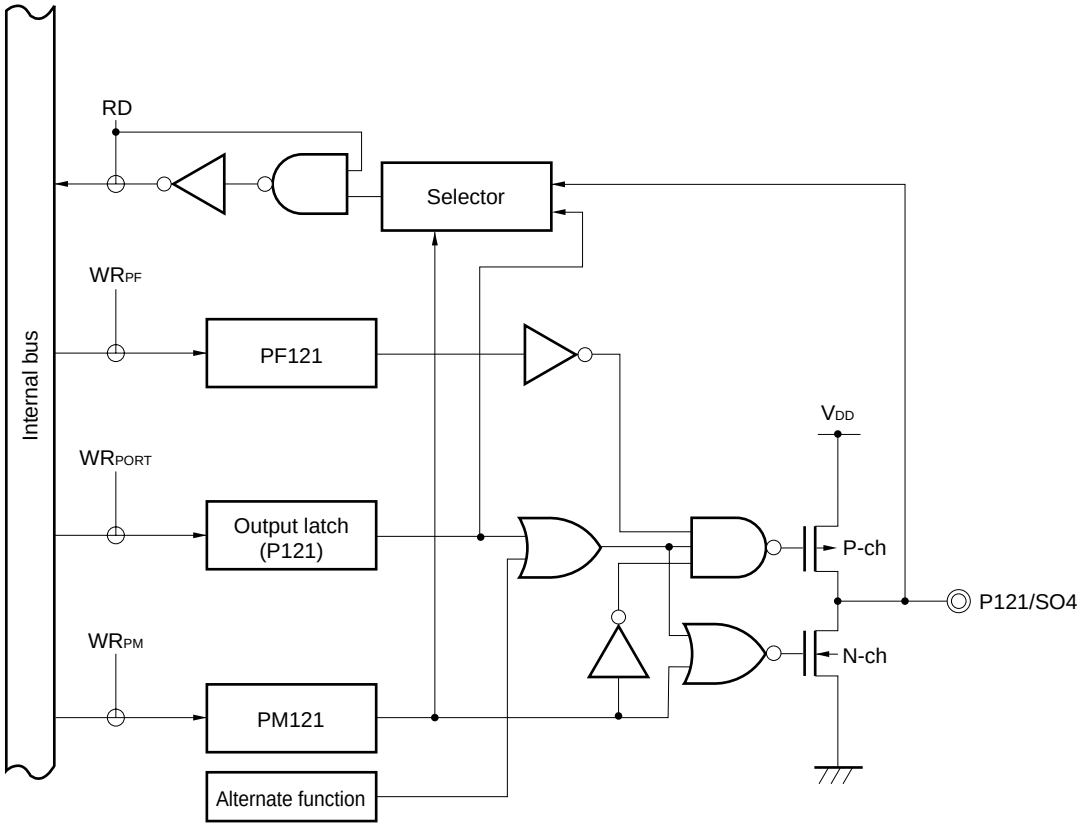


PM: Port mode register

RD: Read signal of port 12

WR: Write signal of port 12

Figure 16-52. Block Diagram of P121



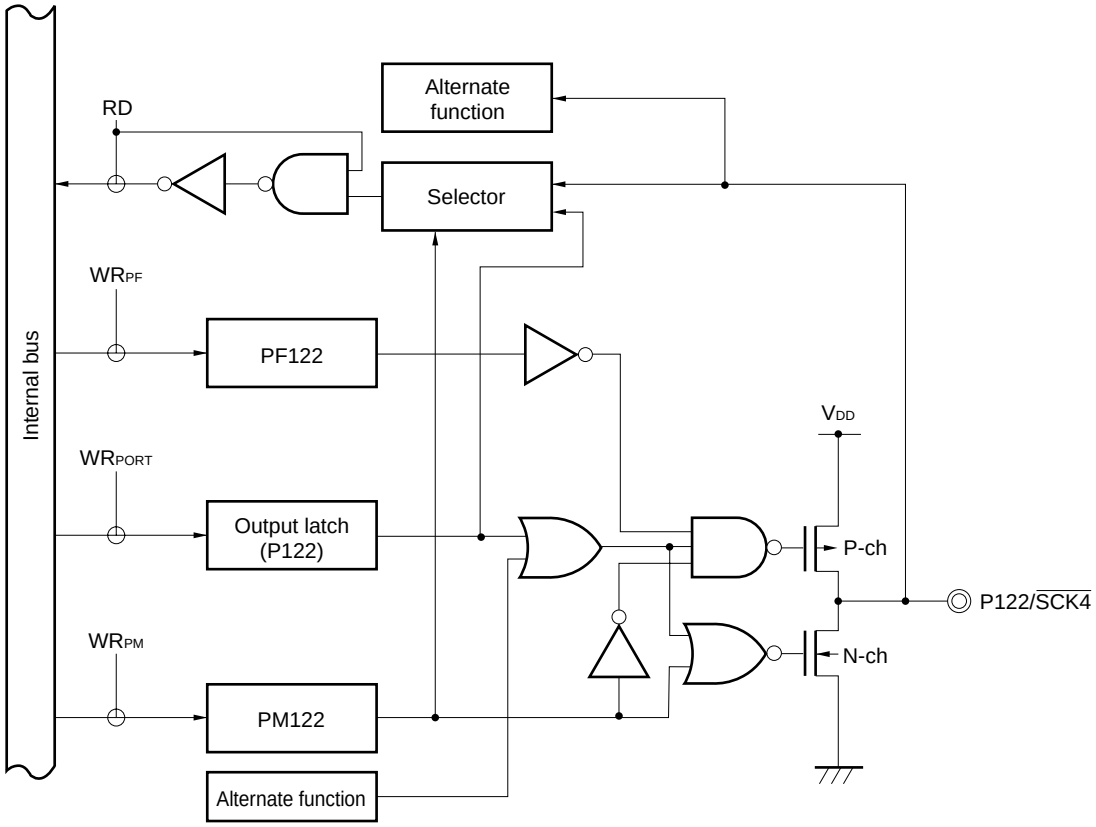
PF: Port function control register

PM: Port mode register

RD: Read signal of port 12

WR: Write signal of port 12

Figure 16-53. Block Diagram of P122



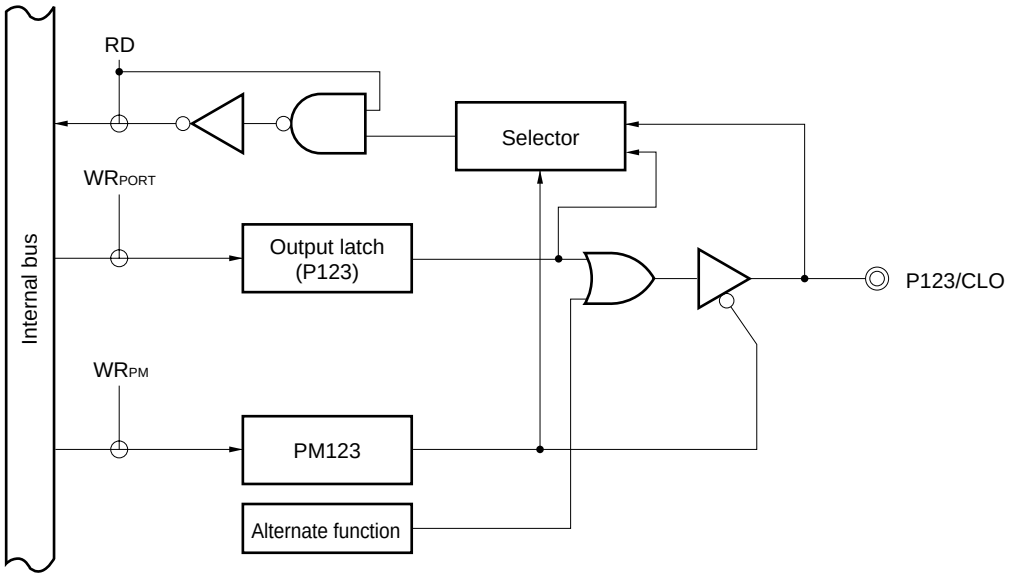
PF: Port function control register

PM: Port mode register

RD: Read signal of port 12

WR: Write signal of port 12

Figure 16-54. Block Diagram of P123

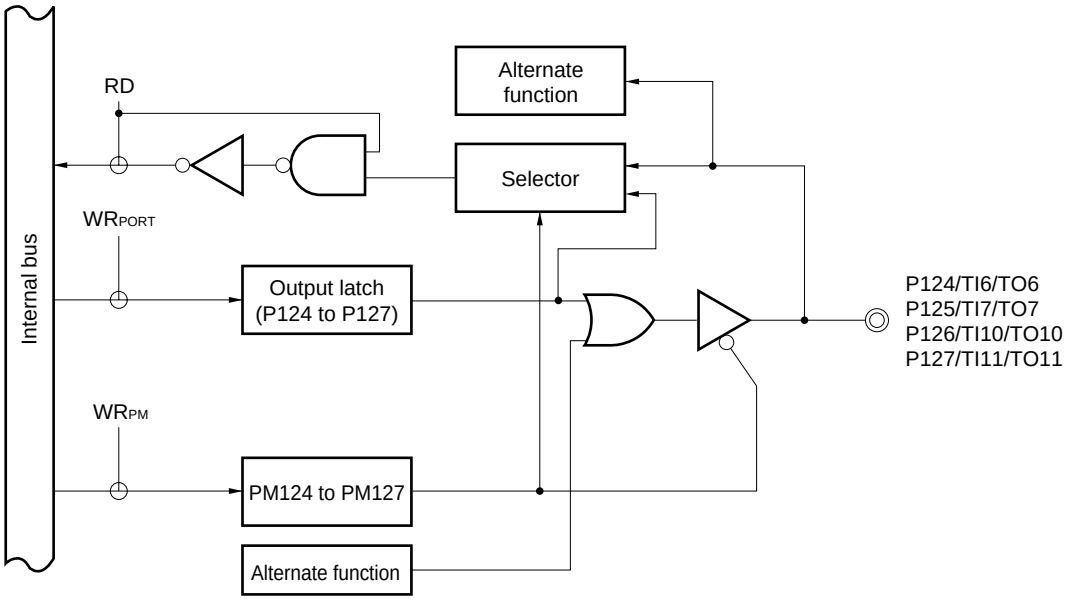


PM: Port mode register

RD: Read signal of port 12

WR: Write signal of port 12

Figure 16-55. Block Diagram of P124 to P127



PM: Port mode register

RD: Read signal of port 12

WR: Write signal of port 12

16.2.12 Port 13

Port 13 is an 8-bit I/O port.

P13 can be read/written in 1-bit or 8-bit units.

When P130 to P135 are used as INTCP80 to INTCP83, TI8 or INTTI8, and TCLR8 or INTTCLR8 pins, noise is eliminated from these pins by a digital noise elimination circuit.

Figure 16-56. Port 13 (P13)

After reset: 00H R/W Address: FFFFF01AH

	7	6	5	4	3	2	1	0
P13	P137	P136	P135	P134	P133	P132	P131	P130

P13n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

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

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

Port 13 includes the following alternate functions.

Table 16-12. Alternate Function of Port 13

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 13	P130	INTCP80	I/O	No	Digital noise elimination
	P131	INTCP81			
	P132	INTCP82			
	P133	INTCP83			
	P134	TI8/INTTI8			
	P135	TCLR8/INTTCLR8			
	P136	TO80			–
	P137	TO81			

Note Software pull-up function

(1) Function of P13 pins

Port 13 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. The I/O settings are controlled via the port 13 mode register (PM13).

In output mode, the value set for each bit is output to the port 13 register (P13).

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

When using the alternate function as the TI8, TCLR8, INTCP80 to INTCP83, INTTI8, and INTTCLR8 pins, noise elimination is provided by a digital noise elimination circuit (same as the digital noise elimination circuit for port 0).

Clear the P13 and PM13 registers to 0 when using the alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

The valid edge of INTCP80 to INTCP83, TI8/INTTI8, and TCLR8/INTTCLR8 are specified via rising edge specification register 2 (EGP2) and falling edge specification register 2 (EGN2).

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

(2) Control registers

(a) Port 13 mode register (PM13)

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

Figure 16-57. Port 13 Mode Register (PM13)

After reset: FFH R/W Address: FFFFF03AH

	7	6	5	4	3	2	1	0
PM13	PM137	PM136	PM135	PM134	PM133	PM132	PM131	PM130

PM13n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

(b) Rising edge specification register 2 (EGP2)

EGP2 can be read/written in 1-bit or 8-bit units.

Figure 16-58. Rising Edge Specification Register 2 (EGP2)

After reset: 00H R/W Address: FFFFF0C8H

	7	6	5	4	3	2	1	0
EGP2	0	0	EGP25	EGP24	EGP23	EGP22	EGP21	EGP20

EGP2n	Control of Rising Edge Validity
0	Interrupt request signal does not occur at rising edge
1	Interrupt request signal occurs at rising edge

n = 0: INTCP80
n = 1: INTCP81
n = 2: INTCP82
n = 3: INTCP83
n = 4: TCLR8/INTTCLR8
n = 5: TI8/INTTI8

(c) Falling edge specification register 2 (EGN2)

EGN2 can be read/written in 1-bit or 8-bit units.

Figure 16-59. Falling Edge Specification Register 2 (EGN2)

After reset: 00H R/W Address: FFFFF0CAH

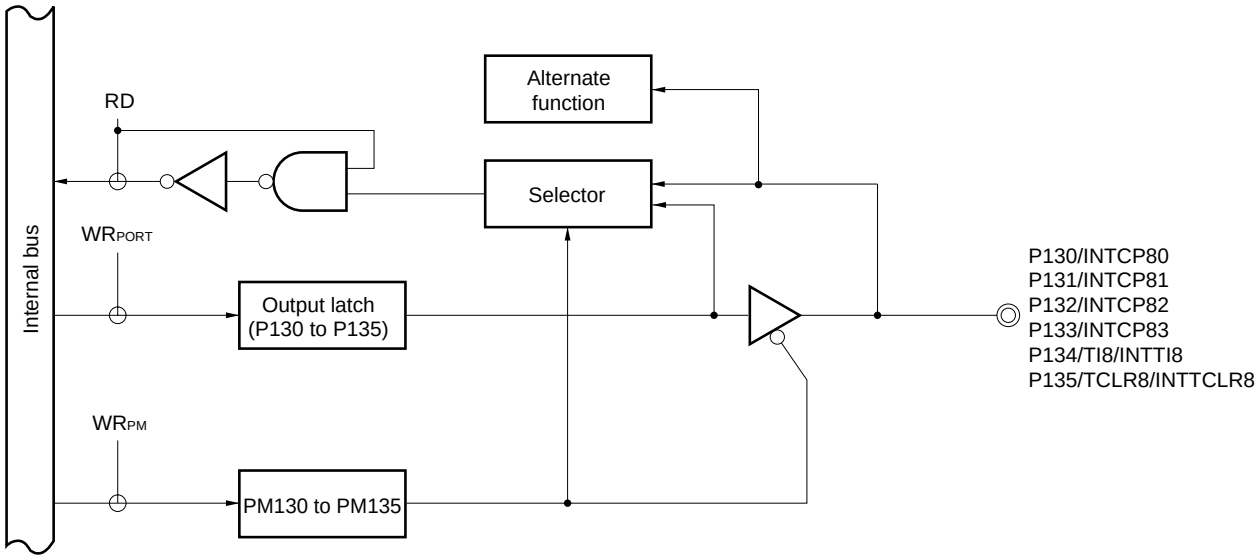
	7	6	5	4	3	2	1	0
EGN2	0	0	EGN25	EGN24	EGN23	EGN22	EGN21	EGN20

EGN2n	Control of Falling Edge Validity
0	Interrupt request signal does not occur at falling edge
1	Interrupt request signal occurs at falling edge

n = 0: INTCP80
n = 1: INTCP81
n = 2: INTCP82
n = 3: INTCP83
n = 4: TCLR8/INTTCLR8
n = 5: TI8/INTTI8

(3) Block diagrams of port 13

Figure 16-60. Block Diagram of P130 to P135

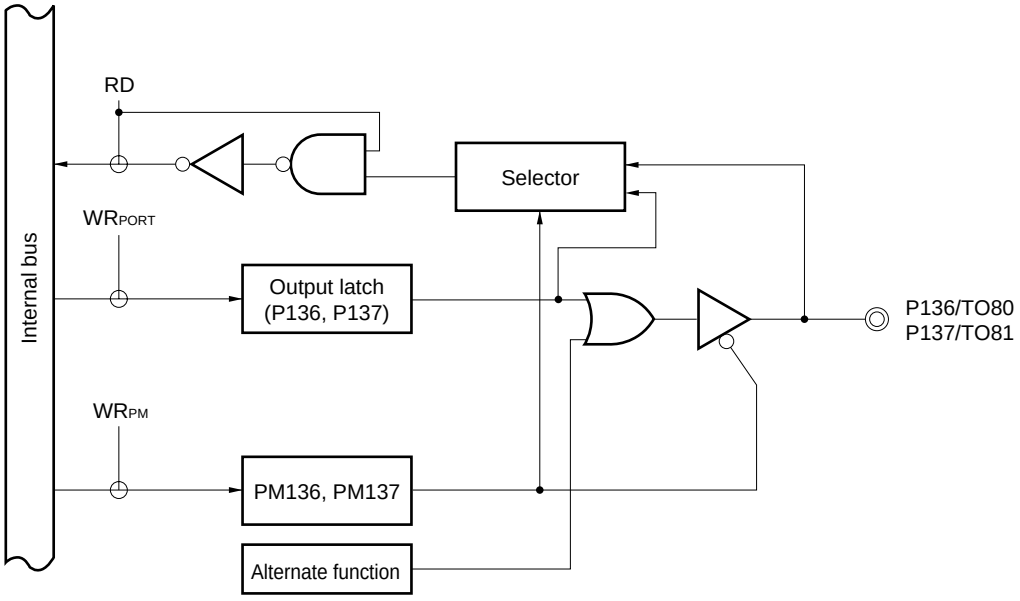


PM: Port mode register

RD: Read signal of port 13

WR: Write signal of port 13

Figure 16-61. Block Diagram of P136 and P137



PM: Port mode register

RD: Read signal of port 13

WR: Write signal of port 13

16.2.13 Port 14

Port 14 is an 8-bit I/O port.

P14 can be read/written in 1-bit or 8-bit units.

When P140 to P145 are used as the INTCP90 to INTCP93, TI9 or INTTI9, and RTPTRG1 pins, noise is eliminated from these pins by a digital noise elimination circuit.

Figure 16-62. Port 14 (P14)

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 the P14 register is read, the pin levels at that time are read. Writing to P14 writes the values to that register. This does not affect the input pins.

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

Port 14 includes the following alternate functions.

Table 16-13. Alternate Function of Port 14

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 14	P140	INTCP90	I/O	No	Digital noise elimination
	P141	INTCP91			
	P142	INTCP92			
	P143	INTCP93			
	P144	TI9/INTTI9			
	P145	RTPTRG1			
	P146	–			–
	P147	–			

Note Software pull-up function

(1) Function of P14 pins

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

In output mode, the value set for each bit is output to the port 14 register (P14).

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

When using the alternate function as the TI9, INTTI9, INTCP90 to INTCP93, and RTPTRG1 pins, noise elimination is provided by a digital noise elimination circuit (same as the digital noise elimination circuit for port 0).

Clear the P14 and PM14 registers to 0 when using alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.

The valid edge of INTCP90 to INTCP93, and TI9/INTTI9 are specified via rising edge specification register 3 (EGP3) and falling edge specification register 3 (EGN3).

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

(2) Control registers

(a) Port 14 mode register (PM14)

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

Figure 16-63. Port 14 Mode Register (PM14)

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

(b) Rising edge specification register 3 (EGP3)

EGP3 can be read/written in 1-bit or 8-bit units.

Figure 16-64. Rising Edge Specification Register 3 (EGP3)

After reset: 00H R/W Address: FFFFF0CCH

	7	6	5	4	3	2	1	0
EGP3	0	0	0	EGP34	EGP33	EGP32	EGP31	EGP30

EGP3n	Control of Rising Edge Validity
0	Interrupt request signal does not occur at rising edge
1	Interrupt request signal occurs at rising edge

n = 0: INTCP90
n = 1: INTCP91
n = 2: INTCP92
n = 3: INTCP93
n = 4: TI9/INTTI9

(c) Falling edge specification register 3 (EGN3)

EGN3 can be read/written in 1-bit or 8-bit units.

Figure 16-65. Falling Edge Specification Register 3 (EGN3)

After reset: 00H R/W Address: FFFFF0CEH

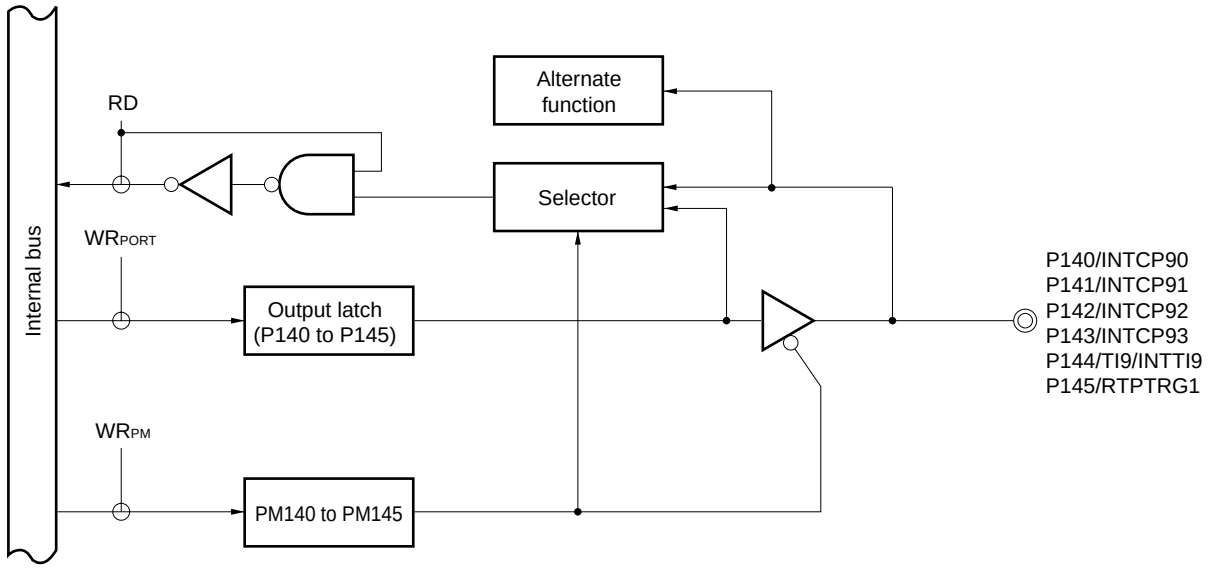
	7	6	5	4	3	2	1	0
EGN3	0	0	0	EGN34	EGN33	EGN32	EGN31	EGN30

EGN3n	Control of Falling Edge Validity
0	Interrupt request signal does not occur at falling edge
1	Interrupt request signal occurs at falling edge

n = 0: INTCP90
n = 1: INTCP91
n = 2: INTCP92
n = 3: INTCP93
n = 4: TI9/INTTI9

(3) Block diagrams of port 14

Figure 16-66. Block Diagram of P140 to P145

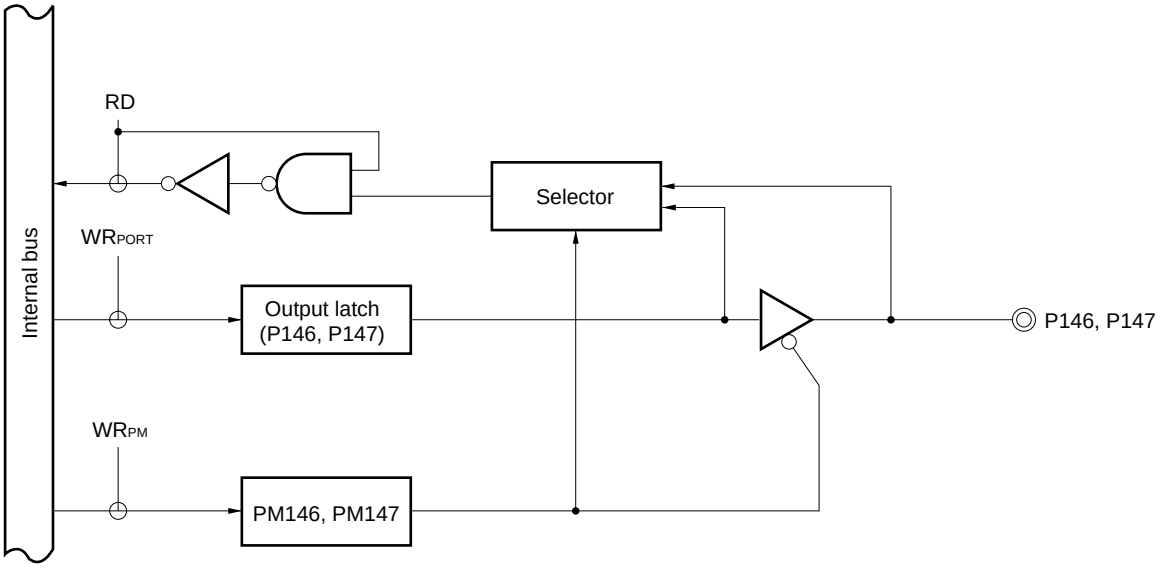


PM: Port mode register

RD: Read signal of port 14

WR: Write signal of port 14

Figure 16-67. Block Diagram of P146 and P147



PM: Port mode register

RD: Read signal of port 14

WR: Write signal of port 14

16.2.14 Port 15

Port 15 is an 8-bit I/O port.
P15 can be read/written in 1-bit or 8-bit units.

Figure 16-68. Port 15 (P15)

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 the P15 register is read, the pin levels at that time are read. Writing to P15 writes the values to that register. This does not affect the input pins.
In output mode: When the P15 register is read, the P15 register's values are read. Writing to P15 writes the values to that register, and those values are immediately output.

Port 15 includes the following alternate functions.

Table 16-14. Alternate Function of Port 15

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 15	P150	RTP10	I/O	No	
	P151	RTP11			
	P152	RTP12			
	P153	RTP13			
	P154	RTP14			
	P155	RTP15			
	P156	RTP16			
	P157	RTP17			

Note Software pull-up function

(1) Function of P15 pins

Port 15 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. The I/O settings are controlled via the port 15 mode register (PM15).
In output mode, the value set for each bit is output to the port 15 register (P15).
When using this port in input mode, the pin statuses can be read by reading the P15 register. Also, the P15 register (output latch) values can be read by reading the P15 register while in output mode.
Clear the P15 and PM15 registers to 0 when using the alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.
When a reset is input, the settings are initialized to input mode.

(2) Control register

(a) Port 15 mode register (PM15)

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

Figure 16-69. Port 15 Mode Register (PM15)

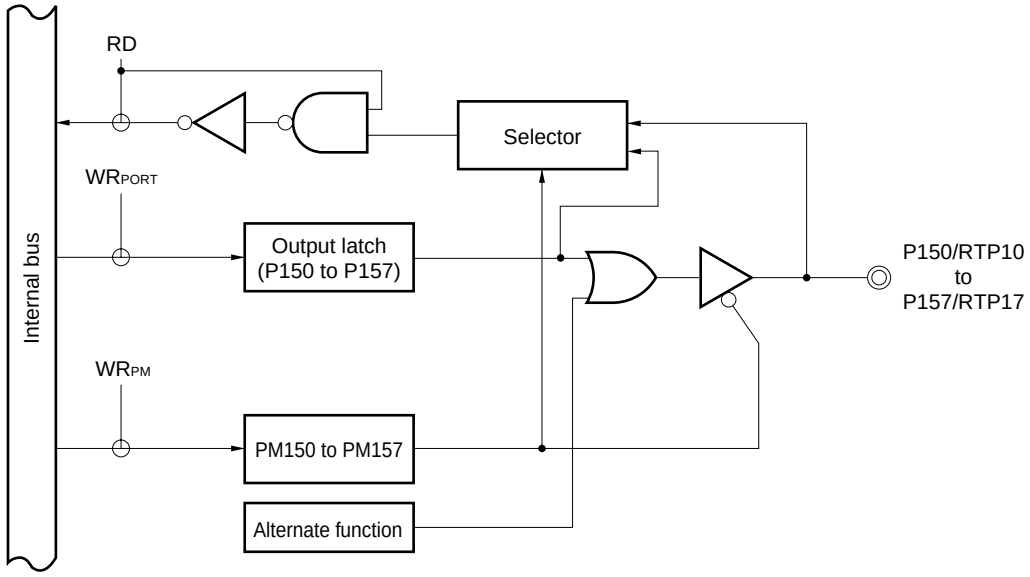
After reset: FFH R/W Address: FFFFF03EH

	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 of port 15

Figure 16-70. Block Diagram of P150 to P157



PM: Port mode register

RD: Read signal of port 15

WR: Write signal of port 15

16.2.15 Port 16

Port 16 is an 8-bit I/O port.
P16 can be read/written in 1-bit or 8-bit units.

Figure 16-71. Port 16 (P16)

After reset: 00H R/W Address: FFFFF080H

	7	6	5	4	3	2	1	0
P16	P167	P166	P165	P164	P163	P162	P161	P160

P16n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P16 register is read, the pin levels at that time are read. Writing to P16 writes the values to that register. This does not affect the input pins.
In output mode: When the P16 register is read, the P16 register's values are read. Writing to P16 writes the values to that register, and those values are immediately output.

Port 16 includes the following alternate functions.

Table 16-15. Alternate Function of Port 16

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 16	P160	PWM0	I/O	No	–
	P161	PWM1			
	P162	PWM2			
	P163	PWM3			
	P164	CSYNCIN			
	P165	VSOUT			
	P166	HSOUT0			
	P167	HSOUT1			

Note Software pull-up function

(1) Function of P16 pins

Port 16 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. The I/O settings are controlled via the port 16 mode register (PM16).
In output mode, the value set for each bit is output to the port 16 register (P16).
When using this port in input mode, the pin statuses can be read by reading the P16 register. Also, the P16 register (output latch) values can be read by reading the P16 register while in output mode.
Clear the P16 and PM16 registers to 0 when using the alternate-function pins as outputs. The ORed result of the port output and the alternate-function pin is output from the pins.
When a reset is input, the settings are initialized to input mode.

(2) Control register

(a) Port 16 mode register (PM16)

PM16 can be read/written in 1-bit or 8-bit units.

Figure 16-72. Port 16 Mode Register (PM16)

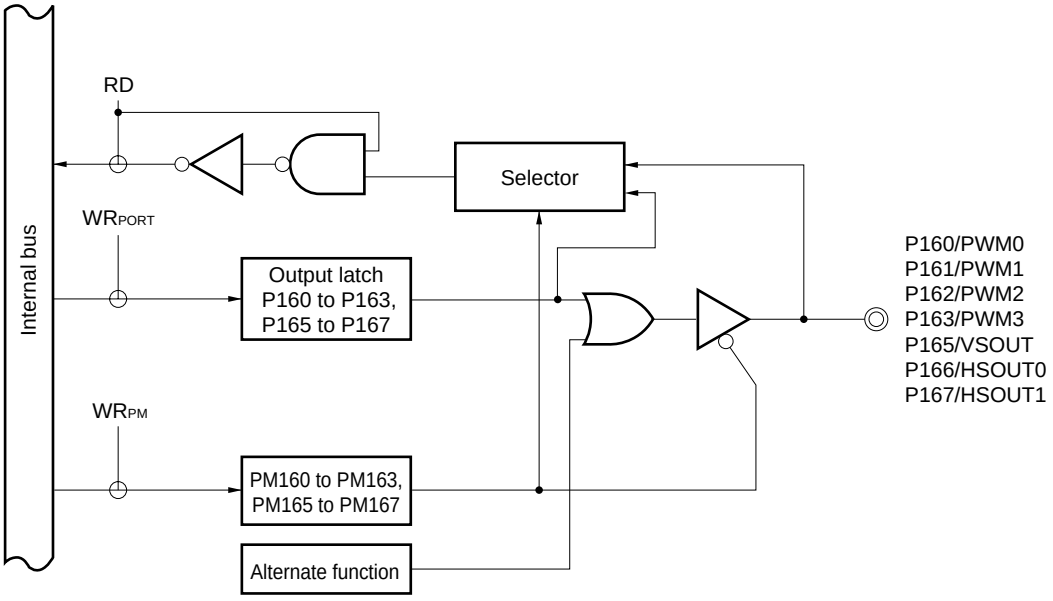
After reset: FFH R/W Address: FFFFF090H

	7	6	5	4	3	2	1	0
PM16	PM167	PM166	PM165	PM164	PM163	PM162	PM161	PM160

PM16n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagrams of port 16

Figure 16-73. Block Diagram of P160 to P163 and P165 to P167

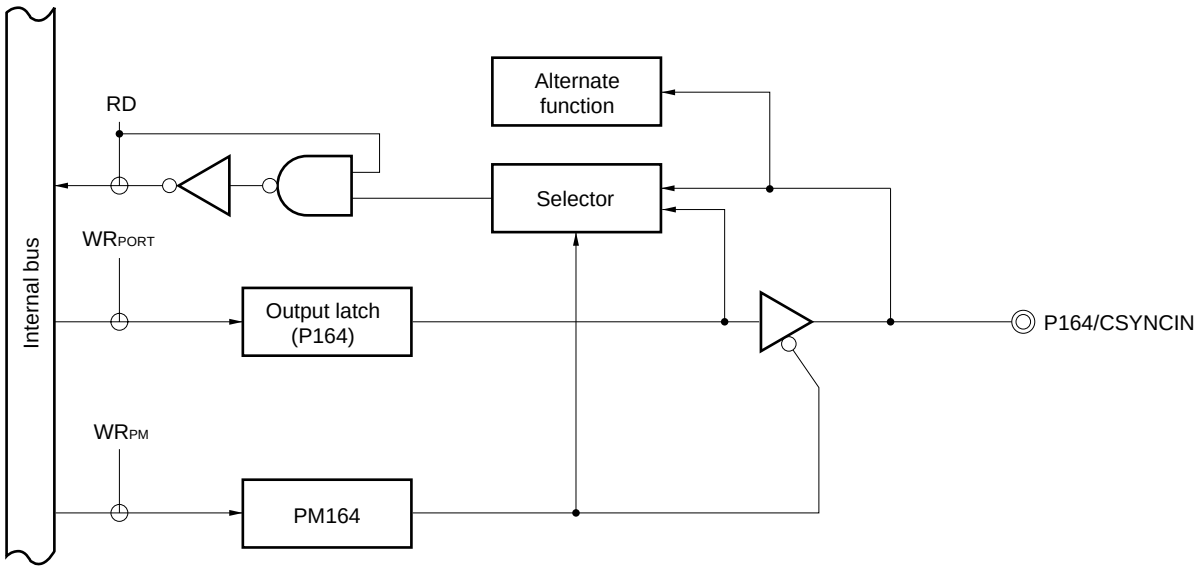


PM: Port mode register

RD: Read signal of port 16

WR: Write signal of port 16

Figure 16-74. Block Diagram of P164



PM: Port mode register

RD: Read signal of port 16

WR: Write signal of port 16

16.2.16 Port 17

Port 17 is an 8-bit I/O port. A pull-up resistor can be connected in 1-bit units (software pull-up function).

P17 can be read/written in 1-bit or 8-bit units.

When P170 to P177 are used as the KR0 to KR7 pins, noise is eliminated from these pins by an analog noise elimination circuit.

Figure 16-75. Port 17 (P17)

After reset: 00H R/W Address: FFFFF082H

	7	6	5	4	3	2	1	0
P17	P177	P176	P175	P174	P173	P172	P171	P170

P17n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

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

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

Port 17 includes the following alternate functions.

Table 16-16. Alternate Function of Port 17

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 17	P170	KR0	I/O	Yes	Analog noise elimination
	P171	KR1			
	P172	KR2			
	P173	KR3			
	P174	KR4			
	P175	KR5			
	P176	KR6			
	P177	KR7			

Note Software pull-up function

(1) Function of P17 pins

Port 17 is an 8-bit I/O port for which I/O settings can be controlled in 1-bit units. The I/O settings are controlled via the port 17 mode register (PM17).

In output mode, the value set for each bit is output to the port 17 register (P17).

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

A pull-up resistor can be connected in 1-bit units when specified via pull-up resistor option register 17 (PU17).

When using the alternate function as the KR0 to KR7 pins, noise elimination is provided by an analog noise elimination circuit (same as the analog noise elimination circuit for port 0).

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

(2) Control registers

(a) Port 17 mode register (PM17)

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

Figure 16-76. Port 17 Mode Register (PM17)

After reset: FFH R/W Address: FFFFF092H

	7	6	5	4	3	2	1	0
PM17	PM177	PM176	PM175	PM174	PM173	PM172	PM171	PM170

PM17n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

Figure 16-77. Pull-Up Resistor Option Register 17 (PU17)

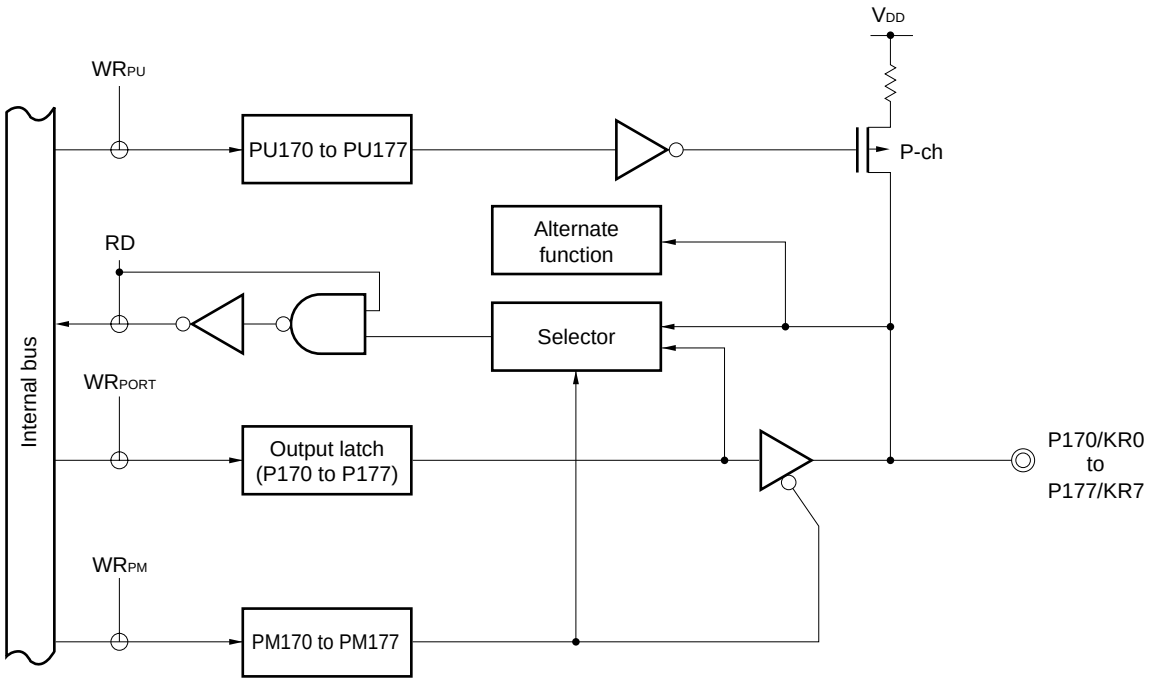
After reset: 00H R/W Address: FFFFF0AAH

	7	6	5	4	3	2	1	0
PU17	PU177	PU176	PU175	PU174	PU173	PU172	PU171	PU170

PU17n	Control of On-Chip Pull-Up Resistor Connection (n = 0 to 7)
0	Do not connect
1	Connect

(3) Block diagram of port 17

Figure 16-78. Block Diagram of P170 to P177



PU: Pull-up resistor option register

PM: Port mode register

RD: Read signal of port 17

WR: Write signal of port 17

16.2.17 Port 18

Port 18 is an 8-bit I/O port.
P18 can be read/written in 1-bit or 8-bit units.

Figure 16-79. Port 18 (P18)

After reset: 00H R/W Address: FFFFF084H

	7	6	5	4	3	2	1	0
P18	P187	P186	P185	P184	P183	P182	P181	P180

P18n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P18 register is read, the pin levels at that time are read. Writing to P18 writes the values to that register. This does not affect the input pins.
In output mode: When the P18 register is read, the P18 register's values are read. Writing to P18 writes the values to that register, and those values are immediately output.

Port 18 has no alternate functions.

Table 16-17. Port 18 (No Alternate Functions)

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 18	P180	–	I/O	No	–
	P181	–			
	P182	–			
	P183	–			
	P184	–			
	P185	–			
	P186	–			
	P187	–			

Note Software pull-up function

(1) Function of P18 pins

Port 18 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 18 mode register (PM18).
In output mode, the value set for each bit is output to the port 18 register (P18).
When using this port in input mode, the pin statuses can be read by reading the P18 register. Also, the P18 register (output latch) values can be read by reading the P18 register while in output mode.
When a reset is input, the settings are initialized to input mode.

(2) Control register

(a) Port 18 mode register (PM18)

PM18 can be read/written in 1-bit or 8-bit units.

Figure 16-80. Port 18 Mode Register (PM18)

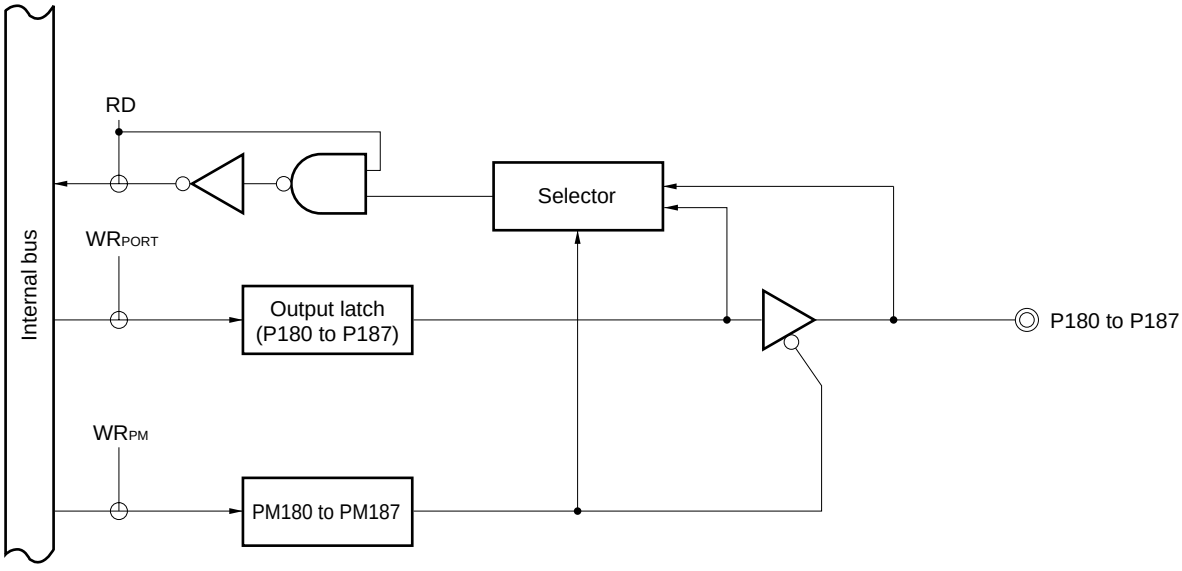
After reset: FFH R/W Address: FFFFF094H

	7	6	5	4	3	2	1	0
PM18	PM187	PM186	PM185	PM184	PM183	PM182	PM181	PM180

PM18n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagram of port 18

Figure 16-81. Block Diagram of P180 to P187



PM: Port mode register

RD: Read signal of port 18

WR: Write signal of port 18

16.2.18 Port 19

Port 19 is an 8-bit I/O port. P19 can be read/written in 1-bit or 8-bit units.

Figure 16-82. Port 19 (P19)

After reset: 00H R/W Address: FFFFF084H

	7	6	5	4	3	2	1	0
P19	P197	P196	P195	P194	P193	P192	P191	P190

P19n	Control of Output Data (in Output Mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

Remark In input mode: When the P19 register is read, the pin levels at that time are read. Writing to P19 writes the values to that register. This does not affect the input pins.

In output mode: When the P19 register is read, the P19 register's values are read. Writing to P19 writes the values to that register, and those values are immediately output.

Port 19 has no alternate functions.

Table 16-18. Port 19 (No Alternate Functions)

Pin Name		Alternate Function	I/O	PULL[Note]	Remark
Port 19	P190	–	I/O	No	–
	P191	–			
	P192	–			
	P193	–			
	P194	–			
	P195	–			
	P196	–			
	P197	–			

Note Software pull-up function

(1) Function of P19 pins

Port 19 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 19 mode register (PM19).

In output mode, the value set for each bit is output to the port 19 register (P19).

When using this port in input mode, the pin statuses can be read by reading the P19 register. Also, the P19 register (output latch) values can be read by reading the P19 register while in output mode.

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

(2) Control register

(a) Port 19 mode register (PM19)

PM19 can be read/written in 1-bit or 8-bit units.

Figure 16-83. Port 19 Mode Register (PM19)

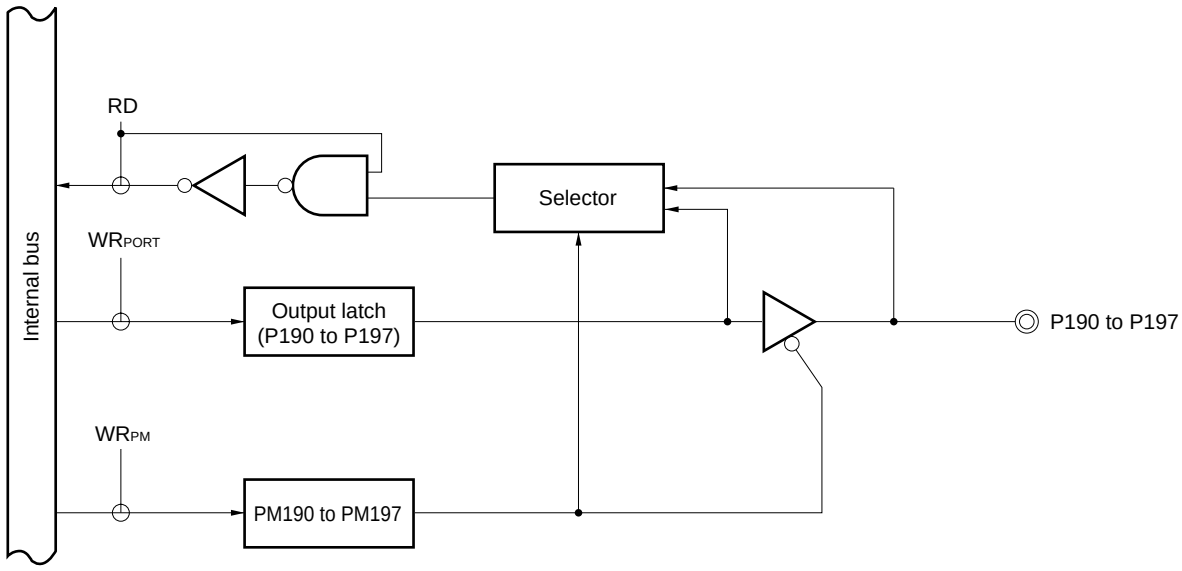
After reset: FFH R/W Address: FFFFF094H

	7	6	5	4	3	2	1	0
PM19	PM197	PM196	PM195	PM194	PM193	PM192	PM191	PM190

PM19n	Control of I/O Mode (n = 0 to 7)
0	Output mode
1	Input mode

(3) Block diagram of port 19

Figure 16-84. Block Diagram of P190 to P197



PM: Port mode register

RD: Read signal of port 19

WR: Write signal of port 19

16.3 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 and PM9 to PM11) and output latch as shown in Table 16-19.

Table 16-19. Setting When Port Pin Is Used for Alternate Function (1/3)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn	Other Bits (Register)
	Name	I/O			
P00	NMI	Input	PM00 = 1	P00 = Setting not required	–
P01	INTP0	Input	PM01 = 1	P01 = Setting not required	–
P02	INTP1	Input	PM02 = 1	P02 = Setting not required	–
P03	INTP2	Input	PM03 = 1	P03 = Setting not required	–
P04	INTP3	Input	PM04 = 1	P04 = Setting not required	–
P05	INTP4	Input	PM05 = 1	P05 = Setting not required	–
	ADTRG	Input			
P06	INTP5	Input	PM06 = 1	P06 = Setting not required	–
	RTPTRG0	Input			
P07	INTP6	Input	PM07 = 1	P07 = Setting not required	–
P10	SI0	Input	PM10 = 1	P10 = Setting not required	–
	SDA0[Note]	I/O	PM10 = 0	P10 = 0	
P11	SO0	Output	PM11 = 0	P11 = 0	–
P12	$\overline{\text{SCK0}}$	Input	PM12 = 1	P12 = Setting not required	–
		Output	PM12 = 0	P12 = 0	
	SCL0[Note]	I/O			
P13	SI1	Input	PM13 = 1	P13 = Setting not required	–
	RXD0	Input			
P14	SO1	Output	PM14 = 0	P14 = 0	–
	TXD0	Output			
P15	$\overline{\text{SCK1}}$	Input	PM15 = 1	P15 = Setting not required	–
		Output	PM15 = 0	P15 = 0	
	ASCK0	Input	PM15 = 1	P15 = Setting not required	
P20	SI2	Input	PM20 = 1	P20 = Setting not required	–
	SDA1[Note]	I/O	PM20 = 0	P20 = 0	
P21	SO2	Output	PM21 = 0	P21 = 0	–
P22	$\overline{\text{SCK2}}$	Input	PM22 = 1	P22 = Setting not required	–
		Output	PM22 = 0	P22 = 0	
	SCL1[Note]	I/O			
P23	SI3	Input	PM23 = 1	P23 = Setting not required	–
	RXD1	Input			
P24	SO3	Output	PM24 = 0	P24 = 0	–
	TXD1	Output			
P25	$\overline{\text{SCK3}}$	Input	PM25 = 1	P25 = Setting not required	–
		Output	PM25 = 0	P25 = 0	
	ASCK1	Input	PM25 = 1	P25 = Setting not required	

Note Provided for the μPD703039Y, 703040Y, 703041Y, and 70F3040Y only.

Table 16-19. Setting When Port Pin Is Used for Alternate Function (2/3)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn	Other Bits (Register)
	Name	I/O			
P26	TI2	Input	PM26 = 1	P26 = Setting not required	–
	TO2	Output	PM26 = 0	P26 = 0	
P27	TI3	Input	PM27 = 1	P27 = Setting not required	–
	TO3	Output	PM27 = 0	P27 = 0	
P30	TI000	Input	PM30 = 1	P30 = Setting not required	–
P31	TI001	Input	PM31 = 1	P31 = Setting not required	–
P32	TI010	Input	PM32 = 1	P32 = Setting not required	–
P33	TI011	Input	PM33 = 1	P33 = Setting not required	–
P34	TO0	Output	PM34 = 0	P34 = 0	–
P35	TO1	Output	PM35 = 0	P35 = 0	–
P36	TI4	Input	PM36 = 1	P36 = Setting not required	–
	TO4	Output	PM36 = 0	P36 = 0	
P37	TI5	Input	PM37 = 1	P37 = Setting not required	–
	TO5	Output	PM37 = 0	P37 = 0	
P40 to P47	AD0 to AD7	I/O	PM40 to PM47 = Setting not required	P40 to P47 = Setting not required	Refer to Figure 3-7 (MM)
P50 to P57	AD8 to AD15	I/O	PM50 to PM57 = Setting not required	P50 to P57 = Setting not required	Refer to Figure 3-7 (MM)
P60 to P65	A16 to A21	Output	PM60 to PM65 = Setting not required	P60 to P65 = Setting not required	Refer to Figure 3-7 (MM)
P70 to P77	ANI0 to ANI7	Input	None	P70 to P77 = Setting not required	–
P80 to P83	ANI8 to ANI11	Input	None	P80 to P83 = Setting not required	–
P90	$\overline{\text{LBEN}}$	Output	PM90 = Setting not required	P90 = Setting not required	Refer to Figure 3-7 (MM)
	$\overline{\text{WRL}}$	Output			
P91	$\overline{\text{UBEN}}$	Output	PM91 = Setting not required	P91 = Setting not required	Refer to Figure 3-7 (MM)
P92	R/$\overline{\text{W}}$	Output	PM92 = Setting not required	P92 = Setting not required	Refer to Figure 3-7 (MM)
	$\overline{\text{WRH}}$	Output			
P93	$\overline{\text{DSTB}}$	Output	PM93 = Setting not required	P93 = Setting not required	Refer to Figure 3-7 (MM)
	$\overline{\text{RD}}$	Output			
P94	ASTB	Output	PM94 = Setting not required	P94 = Setting not required	Refer to Figure 3-7 (MM)
P95	$\overline{\text{HLDAK}}$	Output	PM95 = Setting not required	P95 = Setting not required	Refer to Figure 3-7 (MM)
P96	$\overline{\text{HLDRQ}}$	Input	PM96 = Setting not required	P96 = Setting not required	Refer to Figure 3-7 (MM)
P100 to P107	RTP00 to RTP07	Output	PM100 to PM107 = 0	P100 to P107 = 0	–
P120	SI4	Input	PM120 = 1	PM120 = Setting not required	–
P121	SO4	Output	PM121 = 0	PM121 = 0	–
P122	$\overline{\text{SCK4}}$	Input	PM122 = 1	PM122 = Setting not required	–
		Output	PM122 = 0	PM122 = 0	–
P123	CLO	Output	PM123 = 0	PM123 = 0	–
P124	TI6	Input	PM124 = 1	PM124 = Setting not required	–
	TO6	Output	PM124 = 0	PM124 = 0	

Table 16-19. Setting When Port Pin Is Used for Alternate Function (3/3)

Pin Name	Alternate Function		PMnx Bit of PMn Register	Pnx Bit of Pn	Other Bits (Register)
	Name	I/O			
P125	TI7	Input	PM125 = 1	PM125 = Setting not required	–
	TO7	Output	PM125 = 0	PM125 = 0	
P126	TI10	Input	PM126 = 1	PM126 = Setting not required	–
	TO10	Output	PM126 = 0	PM126 = 0	
P127	TI11	Input	PM127 = 1	PM127 = Setting not required	–
	TO11	Output	PM127 = 0	PM127 = 0	
P130	INTCP80	Input	PM130 = 1	PM130 = Setting not required	–
P131	INTCP81	Input	PM131 = 1	PM131 = Setting not required	–
P132	INTCP82	Input	PM132 = 1	PM132 = Setting not required	–
P133	INTCP83	Input	PM133 = 1	PM133 = Setting not required	–
P134	TI8	Input	PM134 = 1	PM134 = Setting not required	–
	INTTI8	Input	PM134 = 1	PM134 = Setting not required	–
P135	TCLR8	Input	PM135 = 1	PM135 = Setting not required	–
	INTTCLR8	Input	PM135 = 1	PM135 = Setting not required	–
P136	TO80	Output	PM136 = 0	PM136 = 0	–
P137	TO81	Output	PM137 = 0	PM137 = 0	–
P140	INTCP90	Input	PM140 = 1	PM140 = Setting not required	–
P141	INTCP91	Input	PM141 = 1	PM141 = Setting not required	–
P142	INTCP92	Input	PM142 = 1	PM142 = Setting not required	–
P143	INTCP93	Input	PM143 = 1	PM143 = Setting not required	–
P144	TI9	Input	PM144 = 1	PM144 = Setting not required	–
P145	RTPTRG1	Input	PM145 = 1	PM145 = Setting not required	–
P150 to P157	RTP10 to RTP17	Output	PM150 to PM157 = 0	PM150 to PM157 = 0	–
P160 to P163	PWM0 to PWM3	Output	PM160 to PM163 = 0	PM160 to PM163 = 0	–
P164	CSYNCIN	Input	PM164 = 1	PM164 = Setting not required	–
P165	VSOUT	Output	PM165 = 0	PM165 = 0	–
P166	HSOUT0	Output	PM166 = 0	PM166 = 0	–
P167	HSOUT1	Output	PM167 = 0	PM167 = 0	–
P170 to P177	KR0 to KR7	Input	PM170 to PM177 = 1	PM170 to PM177 = Setting not required	–

Cautions 1. When changing the output level of port 0 by setting port 0's port function output mode, the interrupt request flag will be set because port 0 also has an alternate function as external interrupt request input. Therefore, set the corresponding interrupt mask flag to 1 before using the output mode.

2. When using I^2C bus mode, specify N-ch open-drain output for the SDA0/P10, SCL0/P12, SDA1/P20, and SCL1/P22 pins by setting the port n function register (PFn) (n = 1, 2).

Remark PMnx bit of PMn register and Pnx bit of Pn

n: 0 (x = 0 to 7) n: 1 (x = 0 to 5) n: 2 (x = 0 to 7) n: 3 (x = 0 to 7) n: 4 (x = 0 to 7)
n: 5 (x = 0 to 7) n: 6 (x = 0 to 5) n: 7 (x = 0 to 7) n: 8 (x = 0 to 7) n: 9 (x = 0 to 6)
n: 10 (x = 0 to 7) n: 12 (x = 0 to 7) n: 13 (x = 0 to 7) n: 14 (x = 0 to 5) n: 15 (x = 0 to 7)
n: 16 (x = 0 to 7) n: 17 (x = 0 to 7)

CHAPTER 17 RESET FUNCTION

17.1 Outline

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 sub clock 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. The contents of the various registers should be initialized within the program as necessary.

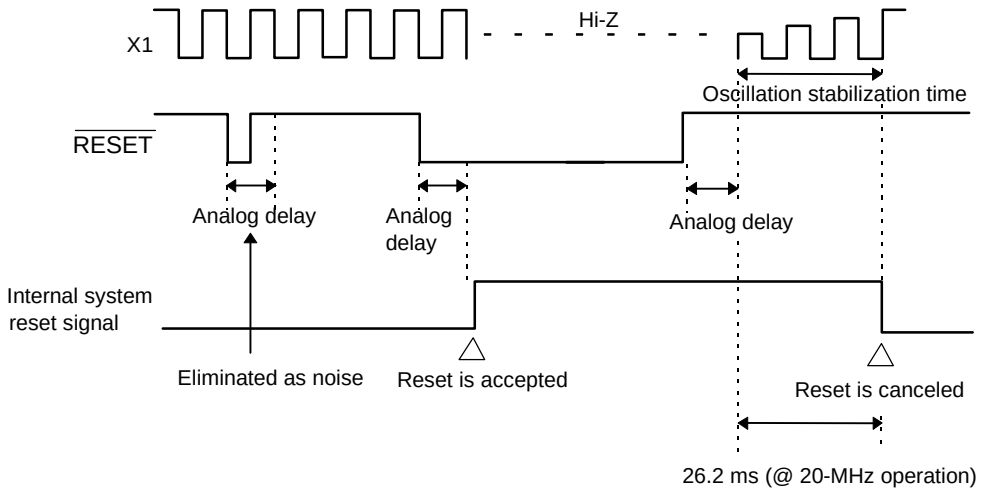
An on-chip noise elimination circuit uses analog delay to prevent noise-related malfunction of the $\overline{\text{RESET}}$ pin.

17.2 Pin Operations

During the system reset period, high impedance is set at almost all pins (all pins except for $\overline{\text{RESET}}$, X2, XT2, AV_{REF}, V_{DD}, V_{SS}, AV_{DD}, AV_{SS}, BV_{DD}, BV_{SS}, and V_{PP}/IC).

Accordingly, if connected to an external memory device, be sure to attach a pull-up (or pull-down) resistor for each pin. If such a resistor is not attached, high impedance will be set for these pins, 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 17-1. System Reset Timing



[MEMO]

CHAPTER 18 ROM CORRECTION FUNCTION

18.1 Outline

The ROM correction function provided in the V850/SV1 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 RAM area by an instruction such as a jump instruction. (n = 0 to 3)

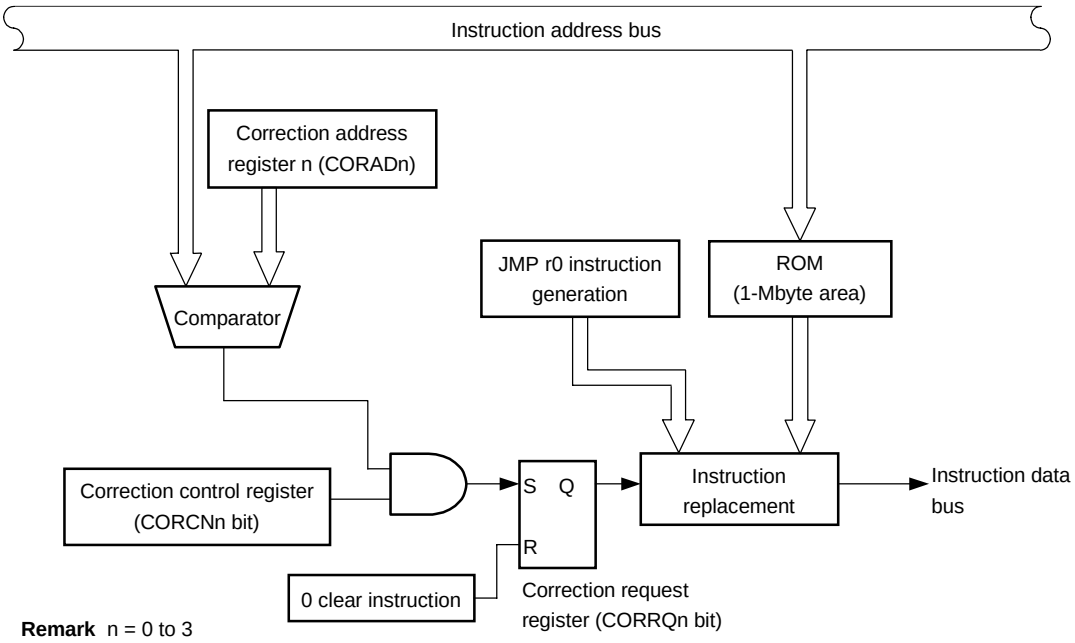
Instruction bugs found in the mask ROM can be avoided, and program flow can be changed by using the ROM correction function.

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 is prohibited for any instruction which accesses the correction request register (CORRQ), correction control register (CORCN), or correction address registers 0 to 3 (CORAD0 to CORAD3).

Figure 18-1. Block Diagram of ROM Correction



Remark n = 0 to 3

18.2 ROM Correction Peripheral I/O Registers

18.2.1 Correction control register (CORCN)

CORCN controls whether or not the instruction of the correction address set in correction address register n (CORADn) is replaced with the JMP r0 instruction when the correction address matches the fetch address (n = 0 to 3).

Whether match detection by a comparator is enabled or disabled can be set for each channel.

CORCN can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 18-2. Correction Control Register (CORCN)

After reset: 00H R/W Address: FFFFF36CH

	7	6	5	4	3	2	1	0
CORCN	0	0	0	0	COREN3	COREN2	COREN1	COREN0

CORENn	CORADn Register and Fetch Address Match Detection Control
0	Match detection disabled (not detected)
1	Match detection enabled (detected)

Remark n = 0 to 3

18.2.2 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

Figure 18-3. Correction Request Register (CORRQ)

After reset: 00H R/W Address: FFFFF36EH

	7	6	5	4	3	2	1	0
CORRQ	0	0	0	0	CORRQ3	CORRQ2	CORRQ1	CORRQ0

CORRQn	Channel n ROM Correction Request Flag
0	No ROM correction request occurred.
1	ROM correction request occurred.

Remark n = 0 to 3

18.2.3 Correction address registers 0 to 3 (CORAD0 to CORAD3)

The correction address registers set the start address of an instruction to be corrected (correction address) in the ROM.

Up to four points of the program can be corrected at once since the V850/SV1 has four correction address registers (CORADn) (n = 0 to 3).

For the V850/SV1, the ROM size differs depends on the version. Set the correction address within the following range:

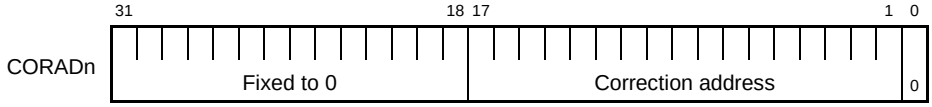
- μPD703041 or 703041Y (192 Kbytes): 00000000H to 0002FFFEH
- μPD703039, 703039Y, 703040, or 703040Y (256 Kbytes): 00000000H to 0003FFFEH

Bits 0 and 18 to 31 should be fixed to 0.

Figure 18-4. Correction Address Registers 0 to 3 (CORAD0 to CORAD3)

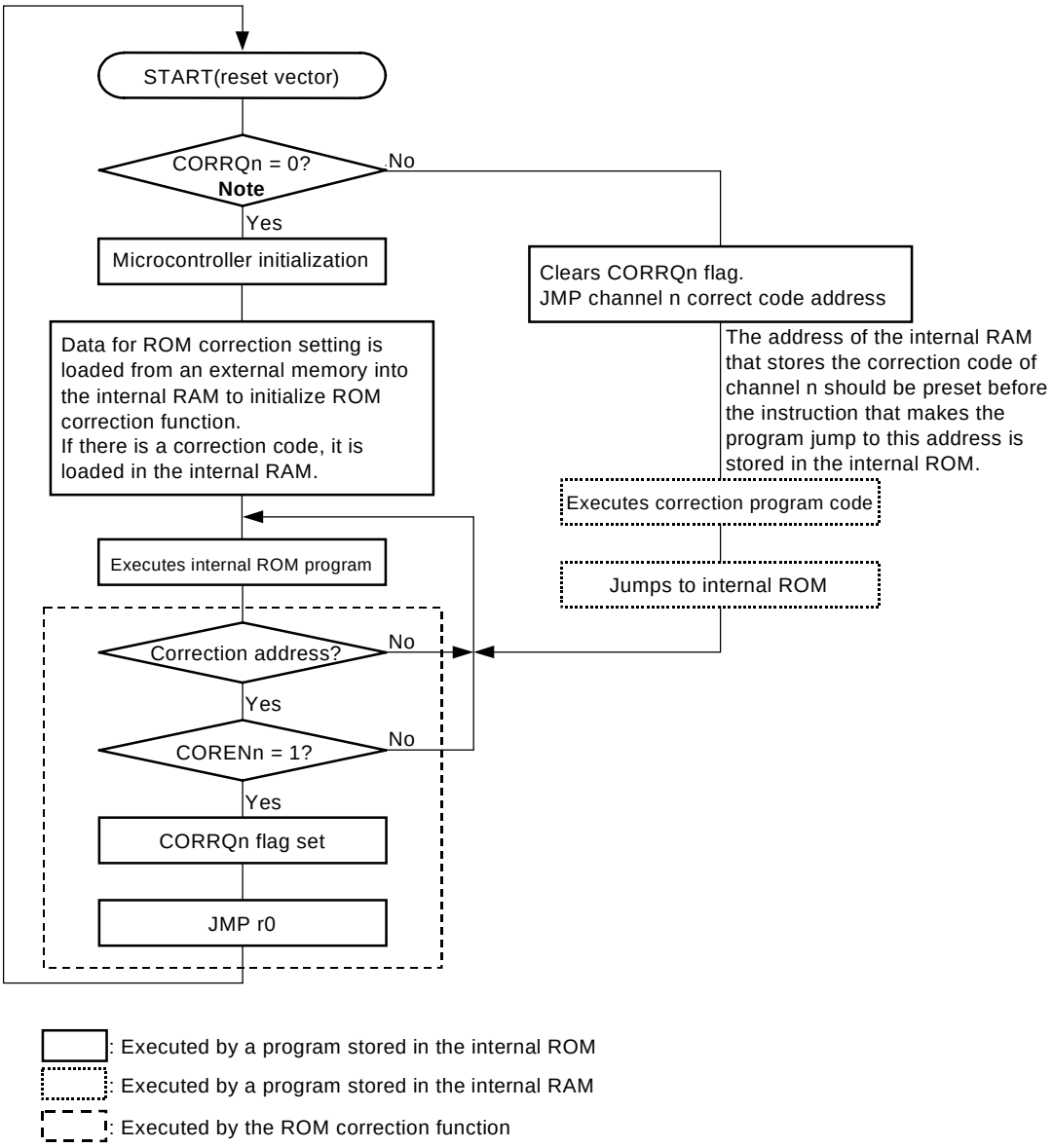
After reset: 00000000H R/W Address: CORAD0: FFFFF370H CORAD2: FFFFF378H

CORAD1: FFFFF374H CORAD3: FFFFF37CH



(n = 0 to 3)

Figure 18-5. ROM Correction Operation and Program Flow



Note To execute ROM correction in a vector interrupt routine, check the vector table with the highest interrupt request level first to see whether ROM correction occurred.
When an interrupt request and ROM correction are in contention, a branch occurs to the interrupt request vector. Executing the ROM correction at the branch destination again sets a CORRQ flag. In this case, multiple CORRQ flags can be set. The channel in which the ROM correction is executed can be determined using the interrupt request level.

Remark n = 0 to 3

CHAPTER 19 FLASH MEMORY (μPD70F3040, 70F3040Y)

The μPD70F3040 and 70F3040Y are flash memory versions of the V850/SV1 Series and are provided with a 256-Kbyte flash memory. 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 a flash memory can be performed with memory mounted on the target system (on board). The dedicated flash programmer is connected to the target system to perform writing.

The followings can be considered as the development environment and the applications using a flash memory.

- Software can be altered after the V850/SV1 is solder mounted on the target system.
- Small scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.

19.1 Features

- 4-byte/1-clock access (in instruction fetch access)
- All area one-shot erase
- Communication via serial interface with the dedicated flash programmer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming
- Number of rewrite: 100 times (target)

19.2 Writing by Flash Programmer

Writing can be performed either on-board or off-board by the dedicated flash programmer.

(1) On-board programming

The contents of the flash memory is rewritten after the V850/SV1 is mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

(2) Off-board programming

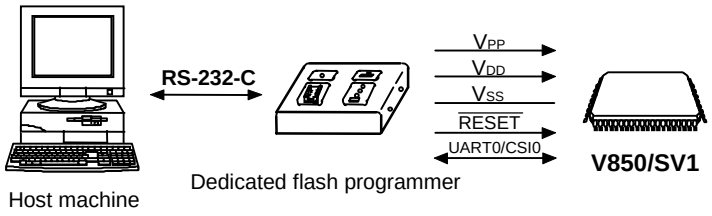
Writing to a flash memory is performed by the dedicated program adapter (FA Series), etc., before mounting the V850/SV1 on the target system.

Remark FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

19.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850/SV1.

Figure 19-1. Programming Environment



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/SV1 to perform writing, erasing, etc. A dedicated program adaptor (FA Series) required for off-board writing.

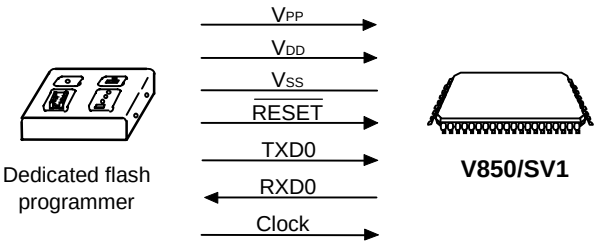
19.4 Communication System

The communication between the dedicated flash programmer and the V850/SV1 is performed by serial communication using UART0 or CSI0.

(1) UART0

Transfer rate: 4,800 to 76,800 bps

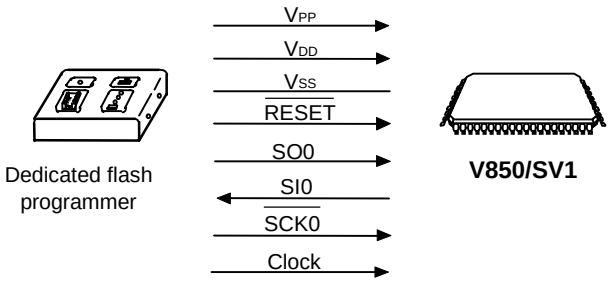
Figure 19-2. Communication System (UART0)



(2) CSI0

Serial clock: up to 1 MHz (MSB first)

Figure 19-3. Communication System (CSI0)



The dedicated flash programmer outputs the transfer clock, and the V850/SV1 operates as a slave.

When PG-FP3 is used as the dedicated flash programmer, it generates the following signals to the V850/SV1. For the details, refer to the PG-FP3 manual.

Table 19-1. Signal Generation of Dedicated Flash Programmer (PG-FP3)

PG-FP3			V850/SV1	Measures When Connected	
Signal Name	I/O	Pin Function	Pin Name	CSI0	UART0
V_{PP}	Output	Writing voltage	V_{PP}	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/ voltage monitoring	V_{DD}	◎	◎
GND	–	Ground	V_{SS}	◎	◎
CLK[Note]	Output	Clock output to V850/SV1	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	$\overline{SCK0}$	◎	×

Note Supply clocks on the target board.

Remark ◎: Always connected
×: Does not need to connect

19.5 Pin Connection

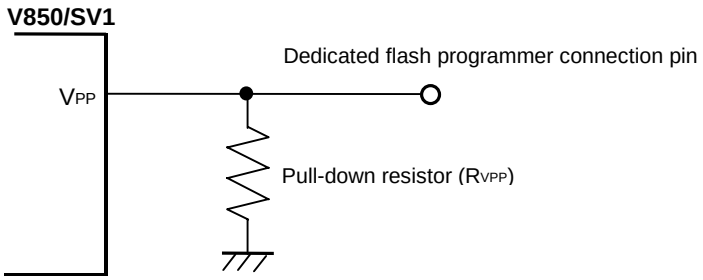
When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, install the function on-board to switch from the normal operation mode to the flash memory programming mode.

When switched to the flash memory programming mode, all the pins not used for the flash memory programming become the same status as that immediately after reset. Therefore, all the ports become output high-impedance status, so that pin handling is required when the external device does not acknowledge the output high-impedance status.

19.5.1 V_{PP} pin

In the normal operation mode, 0 V is input to V_{PP} pin. In the flash memory programming mode, 7.8-V writing voltage is supplied to V_{PP} pin. The following shows an example of the connection of V_{PP} pin.

Figure 19-4. Example of Connection of V_{PP} Pin



19.5.2 Serial interface pin

The following shows the pins used by each serial interface.

Table 19-2. Pins Used by Each Serial Interface

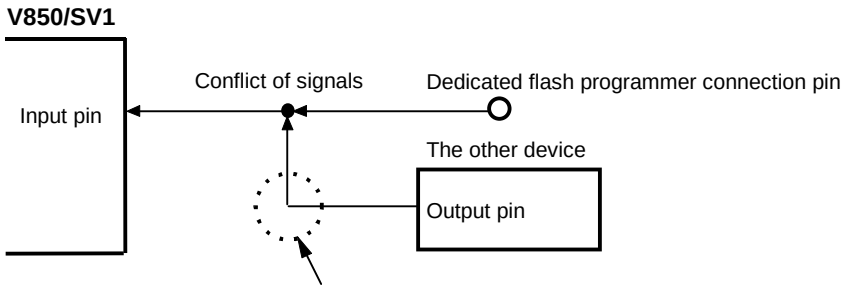
Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$
UART0	TXD0, RXD0

When connecting a dedicated flash programmer to a serial interface pin that is connected to other devices on-board, care should be taken to 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 19-5. Conflict of Signals (Serial Interface Input Pin)

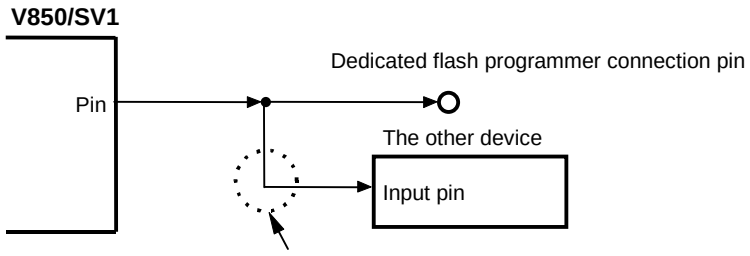


In the flash memory programming mode, the signal that the dedicated flash programmer sends out conflicts with signals the other device outputs. Therefore, isolate the signals on the other device side.

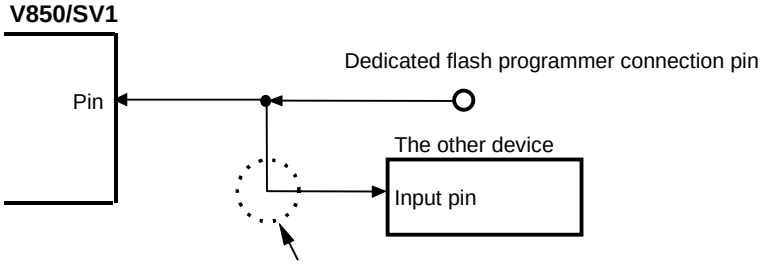
(2) Malfunction of the other device

When connecting a dedicated flash programmer (output or input) to a serial interface pin (input or output) that is connected to another device (input), the signal output to the other device may cause the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.

Figure 19-6. Malfunction of Other Device



In the flash memory programming mode, if the signal the V850/SV1 outputs affects the other device, isolate the signal on the other device side.



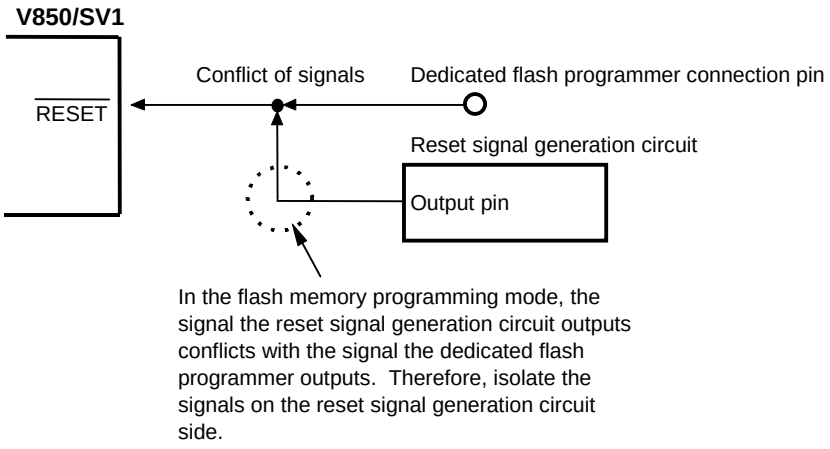
In the flash memory programming mode, if the signal the dedicated flash programmer outputs affects the other device, isolate the signal on the other device side.

19.5.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generation circuit on-board, conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generation circuit.

When a reset signal is input from the user system in the flash memory programming mode, programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

Figure 19-7. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



19.5.4 Port pin (including NMI)

When the flash memory programming mode is set, all the port pins except the pins that communicate with the dedicated flash programmer become output high-impedance status. If problems such as disabling output high-impedance status should occur to the external devices connected to the port, connect them to V_{DD} or V_{SS} through resistors.

19.5.5 Other signal pins

Connect X1, X2, XT2, and AV_{REF} to the same status as that in the normal operation mode.

19.5.6 Power supply

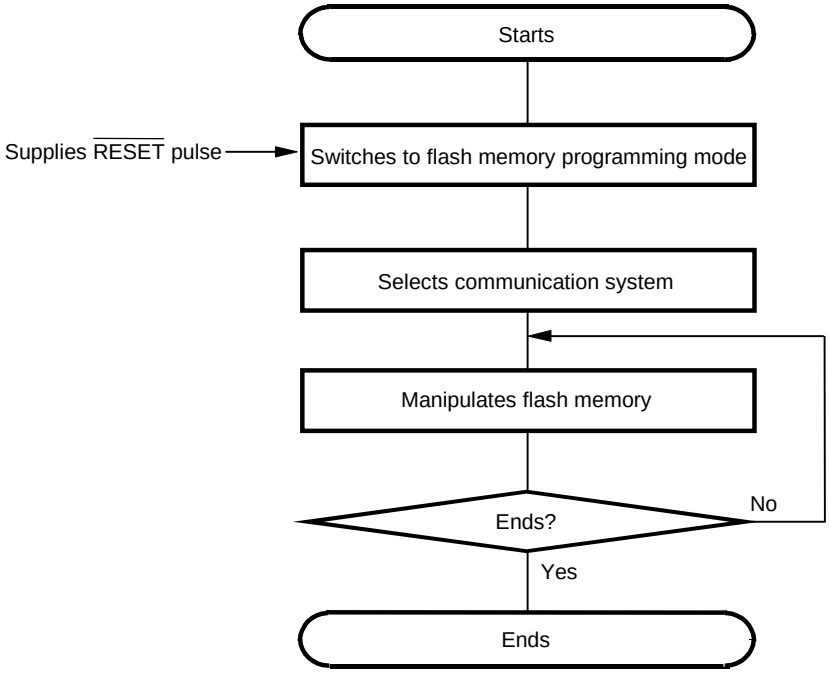
Supply AV_{DD}, AV_{SS}, BV_{DD}, and BV_{SS} the same as when in normal operation mode.

19.6 Programming Method

19.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 19-8. Procedure for Manipulating Flash Memory

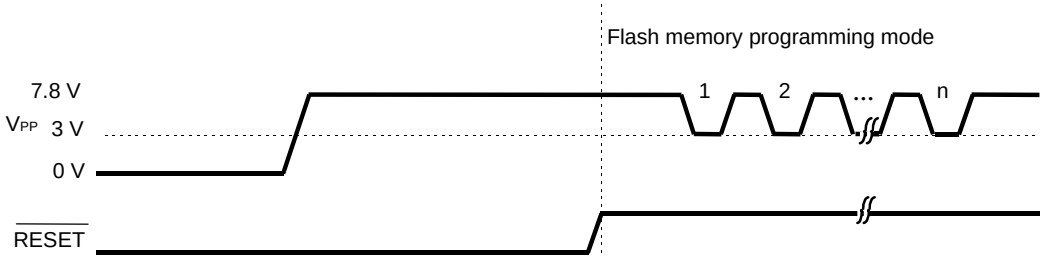


19.6.2 Flash memory programming mode

When rewriting the contents of a flash memory using the dedicated flash programmer, set the V850/SV1 in the flash memory programming mode. When switching to modes, set V_{PP} pin before releasing reset.

When performing on-board writing, change modes using a jumper, etc.

Figure 19-9. Flash Memory Programming Mode



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

19.6.3 Selection of communication system

In the V850/SV1, a communication system is selected by inputting pulse (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 systems.

Table 19-3. List of Communication Systems

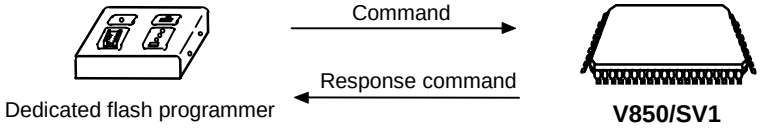
V_{PP} Pulse	Communication System	Remarks
0	CSI0	V850/SV1 performs slave operation, MSB first
8	UART0	Communication rate: 9,600 bps (at reset), LSB first
Others	RFU	Setting prohibited

Caution When UART is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving V_{PP} pulse.

19.6.4 Communication command

The V850/SV1 communicates with the dedicated flash programmer by means of commands. The command sent from the dedicated flash programmer to the V850/SV1 is called a “command”. The response signal sent from the V850/SV1 to the dedicated flash programmer is called a “response command”.

Figure 19-10. Communication Command



The following shows the commands for flash memory control of the V850/SV1. All of these commands are issued from the dedicated flash programmer, and the V850/SV1 performs the various processings corresponding to the commands.

Table 19-4. Commands for Flash Memory Control

Category	Command Name	Function
Verify	One-shot verify command	Compares the contents of the entire memory and the input data.
Erase	One-shot erase command	Erases the contents of the entire memory.
	Write back command	Writes back the contents which is overerased.
Blank check	One-shot blank check command	Checks the erase state of the entire memory.
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.
	Erasing time setting command	Sets the erasing time of one-shot erase.
	Writing time setting command	Sets the writing time of data write.
	Write back time setting command	Sets the write back time.
	Baud rate setting command	Sets the baud rate when using UART.
	Silicon signature command	Reads outs the silicon signature information.
	Reset command	Escapes from each state.

The V850/SV1 sends back response commands to the commands issued from the dedicated flash programmer. The following shows the response commands the V850/SV1 sends out.

Table 19-5. Response Commands

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

19.6.5 Resources used

The resources used in the flash memory programming mode are all the FFE000H to FFE7FFH area of the internal RAM and all the registers. The FFE800H to FFEFFFH area of the internal RAM retains data as long as the power is on. The registers that are initialized by reset are changed to the default value.

[MEMO]

APPENDIX A REGISTER INDEX

(1/9)

Symbol	Name	Unit	Page
ADCR	A/D conversion result register	ADC	384, 390
ADIC	Interrupt control register	INTC	141
ADM0	A/D converter mode register 0	ADC	386
ADM1	A/D converter mode register 1	ADC	388
ASIM00	Asynchronous serial interface mode register 00	UART	357
ASIM01	Asynchronous serial interface mode register 01	UART	359
ASIM10	Asynchronous serial interface mode register 10	UART	357
ASIM11	Asynchronous serial interface mode register 11	UART	359
ASIS0	Asynchronous serial interface status register 0	UART	360
ASIS1	Asynchronous serial interface status register 1	UART	360
BCC	Bus cycle control register	BCU	111
BRGC0	Baud rate generator control register 0	BRG	361
BRGC1	Baud rate generator control register 1	BRG	361
BRGCK4	Baud rate output clock select register 4	BRG	378
BRGCN4	Baud rate generator source clock select register 4	BRG	377
BRGMC00	Baud rate generator mode control register 00	BRG	362
BRGMC01	Baud rate generator mode control register 01	BRG	362
BRGMC10	Baud rate generator mode control register 10	BRG	362
BRGMC11	Baud rate generator mode control register 11	BRG	362
CC80	Capture/compare register 80	RPU	178
CC81	Capture/compare register 81	RPU	178
CC82	Capture/compare register 82	RPU	178
CC83	Capture/compare register 83	RPU	178
CC8IC0	Interrupt control register	INTC	140
CC8IC1	Interrupt control register	INTC	140
CC8IC2	Interrupt control register	INTC	140
CC8IC3	Interrupt control register	INTC	140
CLOM	Clock output mode register	CG	161
CM90	Compare register 90	RPU	180
CM91	Compare register 91	RPU	180
CM9IC0	Interrupt control register	INTC	140
CM9IC1	Interrupt control register	INTC	140
CORAD0	Correction address register 0	CPU	537
CORAD1	Correction address register 1	CPU	537
CORAD2	Correction address register 2	CPU	537
CORAD3	Correction address register 3	CPU	537
CORCN	Correction control register	CPU	536

(2/9)

Symbol	Name	Unit	Page
CORRQ	Correction request register	CPU	537
CP90	Capture register 90	RPU	179
CP91	Capture register 91	RPU	179
CP92	Capture register 92	RPU	179
CP93	Capture register 93	RPU	179
CP9IC0	Interrupt control register	INTC	140
CP9IC1	Interrupt control register	INTC	140
CP9IC2	Interrupt control register	INTC	140
CP9IC3	Interrupt control register	INTC	140
CR00	16-bit capture/compare register 00	RPU	219
CR01	16-bit capture/compare register 01	RPU	220
CR10	16-bit capture/compare register 10	RPU	219
CR100	8-bit compare register 100	RPU	253
CR1011	16-bit compare register 1011 (when connected to TM10, TM11 cascade)	RPU	268
CR11	16-bit capture/compare register 11	RPU	220
CR110	8-bit compare register 110	RPU	253
CR20	8-bit compare register 2	RPU	253
CR23	16-bit compare register 23 (when connected to TM2, TM3 cascade)	RPU	268
CR30	8-bit compare register 3	RPU	253
CR40	8-bit compare register 4	RPU	253
CR45	16-bit compare register 45 (when connected to TM4, TM5 cascade)	RPU	268
CR50	8-bit compare register 5	RPU	253
CR60	8-bit compare register 6	RPU	253
CR67	16-bit compare register 67 (when connected to TM6, TM7 cascade)	RPU	268
CR70	8-bit compare register 7	RPU	253
CRC0	Capture/compare control register 0	RPU	224
CRC1	Capture/compare control register 1	RPU	224
CSIB4	Variable length serial setting register 4	CSI	376
CSIC0	Interrupt control register	INTC	140
CSIC1	Interrupt control register	INTC	140
CSIC2	Interrupt control register	INTC	141
CSIC3	Interrupt control register	INTC	141
CSIC4	Interrupt control register	INTC	141
CSIM0	Serial operation mode register 0	CSI	287
CSIM1	Serial operation mode register 1	CSI	287
CSIM2	Serial operation mode register 2	CSI	287
CSIM3	Serial operation mode register 3	CSI	287
CSIM4	Variable length serial control register 4	CSI	375

(3/9)

Symbol	Name	Unit	Page
CSIS0	Serial clock select register 0	CSI	287
CSIS1	Serial clock select register 1	CSI	287
CSIS2	Serial clock select register 2	CSI	287
CSIS3	Serial clock select register 3	CSI	287
DBC0	DMA byte count register 0	DMAC	416
DBC1	DMA byte count register 1	DMAC	416
DBC2	DMA byte count register 2	DMAC	416
DBC3	DMA byte count register 3	DMAC	416
DBC4	DMA byte count register 4	DMAC	416
DBC5	DMA byte count register 5	DMAC	416
DCHC0	DMA channel control register 0	DMAC	417
DCHC1	DMA channel control register 1	DMAC	417
DCHC2	DMA channel control register 2	DMAC	417
DCHC3	DMA channel control register 3	DMAC	417
DCHC4	DMA channel control register 4	DMAC	417
DCHC5	DMA channel control register 5	DMAC	417
DIOA0	DMA peripheral I/O address register 0	DMAC	413
DIOA1	DMA peripheral I/O address register 1	DMAC	413
DIOA2	DMA peripheral I/O address register 2	DMAC	413
DIOA3	DMA peripheral I/O address register 3	DMAC	413
DIOA4	DMA peripheral I/O address register 4	DMAC	413
DIOA5	DMA peripheral I/O address register 5	DMAC	413
DMAIC0	Interrupt control register	INTC	141
DMAIC1	Interrupt control register	INTC	141
DMAIC2	Interrupt control register	INTC	141
DMAIC3	Interrupt control register	INTC	141
DMAIC4	Interrupt control register	INTC	141
DMAIC5	Interrupt control register	INTC	141
DRA0	DMA internal RAM address register 0	DMAC	414
DRA1	DMA internal RAM address register 1	DMAC	414
DRA2	DMA internal RAM address register 2	DMAC	414
DRA3	DMA internal RAM address register 3	DMAC	414
DRA4	DMA internal RAM address register 4	DMAC	414
DRA5	DMA internal RAM address register 5	DMAC	414
DWC	Data wait control register	BCU	109
ECR	Interrupt source register	CPU	74
EDV0	Event divide counter 0	RPU	190
EDV1	Event divide counter 1	RPU	190

(4/9)

Symbol	Name	Unit	Page
EDV2	Event divide counter 2	RPU	190
EDVC0	Event divide control register 0	RPU	190
EDVC1	Event divide control register 1	RPU	190
EDVC2	Event divide control register 2	RPU	190
EGN0	Falling edge specification register 0	INTC	131, 463
EGN2	Falling edge specification register 2	RPU	187, 510
EGN3	Falling edge specification register 3	RPU	188, 515
EGP0	Rising edge specification register 0	INTC	131, 463
EGP2	Rising edge specification register 2	RPU	186, 510
EGP3	Rising edge specification register 3	RPU	187, 515
EIPC	Interrupt status saving register	CPU	74
EIPSW	Interrupt status saving register	CPU	74
EVS	Event select retgister	RPU	191
FEPC	NMI status saving register	CPU	74
FEPSW	NMI status saving register	CPU	74
HCCMP	Hsync compensation register	VHD	443
HMCMP	Hsync mask width register	VHD	443
HSCMP	Hsync compare register	VHD	442
IIC0	IIC shift register 0	I^2C	296, 308
IIC1	IIC shift register 1	I^2C	296, 308
IICC0	IIC control register 0	I^2C	298
IICC1	IIC control register 1	I^2C	298
IICCL0	IIC clock select register 0	I^2C	306
IICCL1	IIC clock select register 1	I^2C	306
IICIC1	Interrupt control register	I^2C	141
IICS0	IIC status register 0	I^2C	303
IICS1	IIC status register 1	I^2C	303
IICX0	IIC function expansion register 0	I^2C	307
IICX1	IIC function expansion register 1	I^2C	307
ISPR	In-service priority register	INTC	142
KRIC	Interrupt control register	KR	141
KRM	Key return mode register	KR	155
MM	Memory expansion mode register	Port	87
NCC	Noise elimination control register	INTC	144
OSTS	Oscillation stabilization time select register	WDT	163, 279
OVIC8	Interrupt control register	INTC	140
OVIC9	Interrupt control register	INTC	140
P0	Port 0	Port	459

(5/9)

Symbol	Name	Unit	Page
P1	Port 1	Port	465
P10	Port 10	Port	495
P11	Port 11	Port	499
P12	Port 12	Port	502
P13	Port 13	Port	508
P14	Port 14	Port	513
P15	Port 15	Port	518
P16	Port 16	Port	520
P17	Port 17	Port	523
P18	Port 18	Port	526
P19	Port 19	Port	528
P2	Port 2	Port	471
P3	Port 3	Port	478
P4	Port 4	Port	483
P5	Port 5	Port	483
P6	Port 6	Port	486
P7	Port 7	Port	489
P8	Port 8	Port	489
P9	Port 9	Port	491
PCC	Processor clock control register	CG	160
PF1	Port 1 function register	Port	467
PF10	Port 10 function register	Port	497
PF12	Port 12 function register	Port	503
PF2	Port 2 function register	Port	473
PIC0	Interrupt control register	INTC	140
PIC1	Interrupt control register	INTC	140
PIC2	Interrupt control register	INTC	140
PIC3	Interrupt control register	INTC	140
PIC4	Interrupt control register	INTC	140
PIC5	Interrupt control register	INTC	140
PIC6	Interrupt control register	INTC	140
PM0	Port 0 mode register	Port	462
PM1	Port 1 mode register	Port	466
PM10	Port 10 mode register	Port	496
PM11	Port 11 mode register	Port	500
PM12	Port 12 mode register	Port	503
PM13	Port 13 mode register	Port	509
PM14	Port 14 mode register	Port	514

(6/9)

Symbol	Name	Unit	Page
PM15	Port 15 mode register	Port	519
PM16	Port 16 mode register	Port	521
PM17	Port 17 mode register	Port	524
PM18	Port 18 mode register	Port	527
PM19	Port 19 mode register	Port	529
PM2	Port 2 mode register	Port	472
PM3	Port 3 mode register	Port	479
PM4	Port 4 mode register	Port	485
PM5	Port 5 mode register	Port	485
PM6	Port 6 mode register	Port	487
PM9	Port 9 mode register	Port	492
PRCMD	Command register	CG	103
PRM00	Prescaler mode register 00	RPU	226
PRM01	Prescaler mode register 01	RPU	226
PRM10	Prescaler mode register 10	RPU	227
PRM11	Prescaler mode register 11	RPU	227
PSC	Power save control register	CG	162
PSW	Program status word	CPU	74
PU0	Pull-up resistor option register 0	Port	462
PU1	Pull-up resistor option register 1	Port	467
PU10	Pull-up resistor option register 10	Port	497
PU17	Pull-up resistor option register 17	Port	524
PU2	Pull-up resistor option register 2	Port	473
PU3	Pull-up resistor option register 3	Port	479
PWM0	PWM modulo register 0	PWM	432
PWM1	PWM modulo register 1	PWM	432
PWM2	PWM modulo register 2	PWM	432
PWM3	PWM modulo register 3	PWM	432
PWMC0	PWM control register 0	PWM	431
PWMC1	PWM control register 1	PWM	431
PWMC2	PWM control register 2	PWM	431
PWMC3	PWM control register 3	PWM	431
PWPR0	PWM prescaler register 0	PWM	432
PWPR1	PWM prescaler register 1	PWM	432
PWPR2	PWM prescaler register 2	PWM	432
PWPR3	PWM prescaler register 3	PWM	432
RTBH0	Real-time output buffer register H0	RPU	421
RTBH1	Real-time output buffer register H1	RPU	421

(7/9)

Symbol	Name	Unit	Page
RTBL0	Real-time output buffer register L0	RPU	421
RTBL1	Real-time output buffer register L1	RPU	421
RTPC0	Real-time output port control register 0	RPU	424
RTPC1	Real-time output port control register 1	RPU	424
RTPM0	Real-time output port mode register 0	RPU	423
RTPM1	Real-time output port mode register 1	RPU	423
RX0	Receive shift register 0	UART	356
RX1	Receive shift register 1	UART	356
RXB0	Receive buffer register 0	UART	357
RXB1	Receive buffer register 1	UART	357
SERIC0	Interrupt control register	INTC	140
SERIC1	Interrupt control register	INTC	141
SIO0	Serial I/O shift register 0	CSI	286
SIO1	Serial I/O shift register 1	CSI	286
SIO2	Serial I/O shift register 2	CSI	286
SIO3	Serial I/O shift register 3	CSI	286
SIO4	Variable length serial I/O shift register 4	CSI	373
STIC0	Interrupt control register	INTC	141
STIC1	Interrupt control register	INTC	141
SVA0	Slave address register 0	I^2C	296, 308
SVA1	Slave address register 1	I^2C	296, 308
SYC	System control register	CG	106
SYS	System status register	CG	103
TCL100	Timer clock select register 100	RPU	254
TCL101	Timer clock select register 101	RPU	254
TCL110	Timer clock select register 110	RPU	254
TCL111	Timer clock select register 111	RPU	254
TCL20	Timer clock select register 20	RPU	254
TCL21	Timer clock select register 21	RPU	254
TCL30	Timer clock select register 30	RPU	254
TCL31	Timer clock select register 31	RPU	254
TCL40	Timer clock select register 40	RPU	254
TCL41	Timer clock select register 41	RPU	254
TCL50	Timer clock select register 50	RPU	254
TCL51	Timer clock select register 51	RPU	254
TCL60	Timer clock select register 60	RPU	254
TCL61	Timer clock select register 61	RPU	254
TCL70	Timer clock select register 70	RPU	254

(8/9)

Symbol	Name	Unit	Page
TCL71	Timer clock select register 71	RPU	254
TM0	16-bit timer register 0	RPU	218
TM1	16-bit timer register 1	RPU	218
TM10	8-bit counter 10	RPU	253
TM1011	16-bit counter 1011 (When connected to TM10, TM11 cascade)	RPU	268
TM11	8-bit counter 11	RPU	253
TM2	8-bit counter 2	RPU	253
TM23	16-bit counter 23 (When connected to TM2, TM3 cascade)	RPU	268
TM3	8-bit counter 3	RPU	253
TM4	8-bit counter 4	RPU	253
TM45	16-bit counter 45 (When connected to TM4, TM5 cascade)	RPU	268
TM5	8-bit counter 5	RPU	253
TM6	8-bit counter 6	RPU	253
TM67	16-bit counter 67 (When connected to TM6, TM7 cascade)	RPU	268
TM7	8-bit counter 7	RPU	253
TM8	Timer 8	RPU	178
TM9	Timer 9	RPU	179
TMC0	16-bit timer mode control register 0	RPU	221
TMC1	16-bit timer mode control register 1	RPU	221
TMC10	8-bit timer mode control register 10	RPU	258
TMC11	8-bit timer mode control register 11	RPU	258
TMC2	8-bit timer mode control register 2	RPU	258
TMC3	8-bit timer mode control register 3	RPU	258
TMC4	8-bit timer mode control register 4	RPU	258
TMC5	8-bit timer mode control register 5	RPU	258
TMC6	8-bit timer mode control register 6	RPU	258
TMC7	8-bit timer mode control register 7	RPU	258
TMC80	24-bit timer mode control register 80	RPU	181
TMC81	24-bit timer mode control register 81	RPU	182
TMC82	24-bit timer mode control register 82	RPU	183
TMC90	24-bit timer mode control register 90	RPU	184
TMC91	24-bit timer mode control register 91	RPU	185
TMIC000	Interrupt control register	INTC	140
TMIC001	Interrupt control register	INTC	140
TMIC010	Interrupt control register	INTC	140
TMIC011	Interrupt control register	INTC	140
TMIC10	Interrupt control register	INTC	140
TMIC11	Interrupt control register	INTC	140

(9/9)

Symbol	Name	Unit	Page
TMIC2	Interrupt control register	INTC	140
TMIC3	Interrupt control register	INTC	140
TMIC4	Interrupt control register	INTC	140
TMIC5	Interrupt control register	INTC	140
TMIC6	Interrupt control register	INTC	140
TMIC7	Interrupt control register	INTC	140
TOC0	Timer output control register 0	RPU	224
TOC1	Timer output control register 1	RPU	224
TOC8	Timer output control register 8	RPU	185
TOVS	Timer overflow status register	RPU	186
TXS0	Transmit shift register 0	UART	356
TXS1	Transmit shift register 1	UART	356
VSC	Vsync control register	VHD	444
VSCMP	Vsync compare register	VHD	442
VSUDC	Vsync up/down counter	VHD	442
WDCS	Watchdog timer clock select register	WDT	280
WDTIC	Interrupt control register	INTC	140
WDTM	Watchdog timer mode register	WDT	143, 281
WTNCS	Watch timer clock select register	WT	274
WTNIC	Interrupt control register	INTC	141
WTNIIC	Interrupt control register	INTC	140
WTNM	Watch timer mode control register	WT	273

[MEMO]

APPENDIX B LIST OF INSTRUCTION SETS

- **How to read instruction set list**

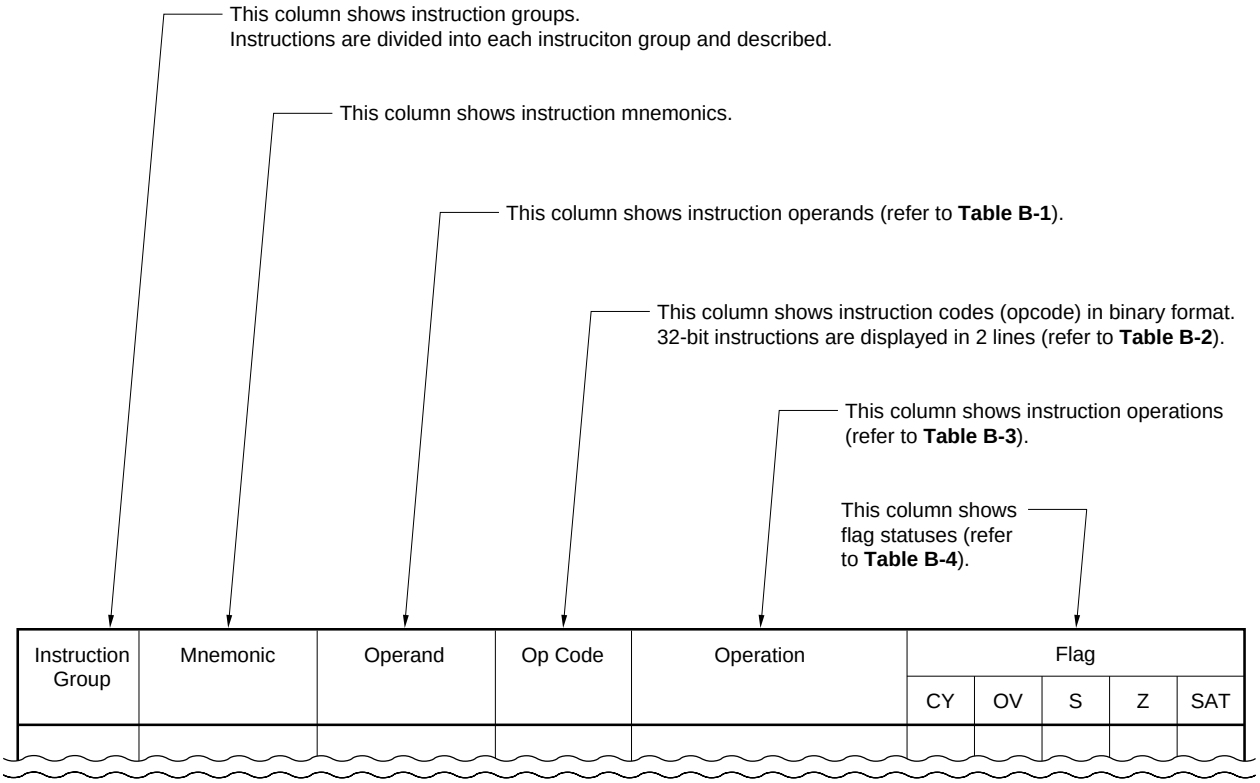


Table B-1. Symbols in Operand Description

Symbol	Description
reg1	General register (r0 to r31): Used as source register
reg2	General 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 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	Half-word (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. dddddddd 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 half-word 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. ddddddddddddddddddddd is the higher 21 bits of dip22.

2. dddddddd is the higher 8 bits of disp9.

Instruction Set List (4/4)

Instruction Group	Mnemonic	Operand	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 this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

[MEMO]

[Number]

[A]

[B]

[C]

[D]

[E]

[F]

[G]

[H]

[I]

[K]

[R]

[S]

[T]

[U]

[V]

[W]

[X]

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
Technical Documentation 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: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-435-9608

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	❑	❑	❑	❑