NEC

User's Manual

V850E/MA1™

32-Bit Single-Chip Microcontroller

Hardware

μPD703103
μPD703105
μPD703106
μPD703107
μPD70F3107

Document No. U14359EJ3V0UM00 (3rd edition)
Date Published February 2001 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed:	μPD703103, 70F3107
The customer must judge the need for license:	μPD703105, 703106, 703107

- **The information in this document is current as of December, 2000. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.
  The quality grade of NEC semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.
  (Note)
  (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.
  (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00.4

Regional Information

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

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

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

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

NEC 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

Major Revisions in This Edition (1/2)

Page	Description
Throughout	• The following products have been developed μPD703016GJ-xxx-UEN, 703107GJ-xxx-UEN, and 70F3107GJ-UEN • Addition of product under development 161-pin plastic FBGA package
p.33	Change of pin names in **Pin Identification**
p.37	Modification of description in **1.6.2 (11) Ports**
p.41	Change of pin names in **2.1 (2) Non-port pins**
p.57	Addition of description to **2.3 (12) (b) (ii) SDCLK (SDRAM clock output)**
p.62	Change of description in **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**
p.80	Addition of Note and modification of Caution 1 in **3.4.5 (3) Internal peripheral I/O area**
p.83	Modification of Note in **Figure 3-9 Recommended Memory Map**
p.93	Change of description in **3.4.10 System wait control register (VSWC)**
p.96	Addition of Note to **4.3 Memory Block Function**
p.102	Modification of description in **4.5.1 Number of access clocks**
p.124	Modification of description in **Table 4-1 Bus Cycles in Which Wait Function Is Valid**
p.138	Modification of description in **4.9 Bus Priority Order**
p.154	Addition of Note to **Figure 5-5 Page ROM Access Timing**
p.199	Modification of **Figure 5-18 CBR Refresh Timing (SDRAM)**
p.201	Modification of **Figure 5-19 Self-Refresh Timing (SDRAM)**
p.253	Modification of Remark 1 in **Table 6-1 Relationship Between Transfer Type and Transfer Object**
p.294	Modification of **Figure 7-14 Pipeline Operation at Interrupt Request Acknowledgement (Outline)**
p.295	Addition of description to **7.8 Periods in Which Interrupts Are Not Acknowledged**
p.298	Modification of description in **9.3.1 Direct mode**
p.299	Modification of Caution in **9.3.2 PLL mode**
p.301	Modification of Caution 3 in **9.3.4 Clock control register (CKC)**
p.308	Modification of Caution 4 in **9.5.2 (3) Power-save control register (PSC)**
p.317	Modification of Figure in **9.6.1 (1) Securing the time using an on-chip time base counter**
p.318	Modification of Figure in **9.6.1 (2) Securing the time according to the signal level width ($\overline{\text{RESET}}$ pin input)**
p.326	Addition of Caution to **10.1.5 (1) Timer mode control registers C00 to C30 (TMCC00 to TMCC30)**
pp.335, 336	Addition of description to **10.1.6 (4) Compare operation**
p.335	Modification and addition in **Figure 10-5 Compare Operation Example**
p.339	Modification of **Figure 10-8 Interval Timer Operation Timing Example**
p.341	Modification of **Figure 10-10 PWM Output Timing Example**
p.344	Modification of **Figure 10-12 Cycle Measurement Operation Timing Example**
p.352	Modification of **Figure 10-14 TMD0 Compare Operation Example**
p.359	Modification of Caution in **11.2.3 (1) Asynchronous serial interface mode registers 0 to 2 (ASIM0 to ASIM2)**
p.394	Modification of description in **11.3.4 (1) Transfer mode**

Major Revisions in This Edition (2/2)

Page	Description
p.404	Addition of description to **12.3 (2) A/D converter mode register 1 (ADM1)**
p.405	Modification of Caution in **12.3 (3) A/D converter mode register 2 (ADM2)**
p.451	Modification of **Figure 14-10 Block Diagram of Type K**
p.500	Modification of Remark in **Figure 16-1 Connection Example of Adapter (FA-144GJ-UEN) for V850E/MA1 Flash Memory Programming**
p.504	Modification of description in **16.5.6 Port pins**
p.508	Addition of description to **16.7.1 Outline of self-programming**
p.516	Modification of **Table 16-8 Flash Information**
p.519	Modification of Caution 1 in **16.7.12 Flash programming mode control register (FLPMC)**

The mark ⋆ shows major revised points.

INTRODUCTION

Readers

This manual is intended for users who wish to understand the functions of the V850E/MA1 (μPD703103, 703105, 703106, 703107, 70F3107) to design application systems using the V850E/MA1.

Purpose

The purpose of this manual is for users to gain an understanding of the hardware functions of the VE850E/MA1.

Organization

The **V850E/MA1 User's Manual** is divided into two parts: Hardware (this manual) and Architecture **(V850E1 User's Manual Architecture)**. The organization of each manual is as follows:

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

How to Read This Manual

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

- To find the details of a register where the name is known
  →Refer to **APPENDIX A REGISTER INDEX**.

- To find the details of a function, etc. where the name is known
  →Refer to **APPENDIX C INDEX**.

- To understand the details of an instruction function
  →Refer to the **V850E1 User's Manual Architecture**.

- To understand the overall functions of the V850E/MA1
  →Read this manual according to the **CONTENTS**.

- How to interpret the register format
  →For a bit whose bit number is enclosed in brackets, its bit name is defined as a reserved word in the device file.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
Memory map address:	Higher addresses on the top and lower addresses on the bottom
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH
Prefix indicating power of 2 (address space, memory capacity):	K (kilo): $2^{10} = 1,024$
	M (mega): $2^{20} = 1,024^2$
	G (giga): $2^{30} = 1,024^3$
Data type:	Word ... 32 bits
	Halfword ... 16 bits
	Byte ... 8 bits

Related documents

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

Document related to V850E/MA1

Document Name	Document No.
V850E1 User's Manual Architecture	U14559E
V850E/MA1 User's Manual Hardware	This manual
μPD70F3107 Data Sheet	U14618E
μPD703106, 703107 Data Sheet	U14792E

Document related to development tools (User's Manuals)

Document Name		Document No.
IE-V850E-MC, IE-V850E-MC-A (In-circuit emulator)		U14487E
IE-703107-MC-EM1 (Peripheral I/O board)		U14481E
CA850 (Ver. 2.30 or later) (C compiler package)	Operation	U14568E
	C Language	U14566E
	Project Manager	U14569E
	Assembly Language	U14567E
ID850 (Ver. 2.20 or later) (Integrated debugger)	Operation Windows™ Based	U14580E
RX850 (Ver. 3.13 or later) (Real-time OS)	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro (Ver. 3.13) (Real-time OS)	Basics	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 (Ver. 3.01) (Task debugger)		U13737E
RD850 Pro (Ver. 3.01) (Task debugger)		U13916E
AZ850 (System performance analyzer)		U14410E
PG-FP3 (Flash memory programmer)		U13502E

CONTENTS

LIST OF FIGURES (1/5)

LIST OF FIGURES (2/5)

LIST OF FIGURES (3/5)

LIST OF FIGURES (4/5)

LIST OF FIGURES (5/5)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 INTRODUCTION

The V850E/MA1 is a product of NEC's single-chip microcontroller "V850 Family™". This chapter gives a simple outline of the V850E/MA1.

1.1 Outline

The V850E/MA1 is a 32-bit single-chip microcontroller that integrates the V850E1 CPU, which is a 32-bit RISC-type CPU core for ASIC, newly developed as the CPU core central to system LSI for the current age of system-on-chip. This device incorporates ROM, RAM, and various peripheral functions such as memory controllers, a DMA controller, real-time pulse unit, serial interfaces, and an A/D converter for realizing high-capacity data processing and sophisticated real-time control.

(1) V850E1 CPU

The V850E1 CPU is a CPU core that enhances the external bus interface performance of the V850 CPU, which is the CPU core integrated in the V850 Family, and has added instructions supporting high-level languages, such as C-language switch statement processing, table lookup branching, stack frame creation/deletion, and data conversion. This enhances the performance of both data processing and control.

It is possible to use the software resources of the V850 CPU integrated system since the instruction codes of the V850E1 are upwardly compatible at the object code level with those of the V850 CPU.

(2) External memory interface function

The V850E/MA1 features various on-chip external memory interfaces including separately configured address (26 bits) and data (16 bits) buses, and SDRAM and ROM interfaces, as well as on-chip memory controllers that can be directly linked to EDO DRAM, page ROM, etc., thereby raising system performance and reducing the number of parts needed for application systems.

Also, through the DMA controller, CPU internal calculations and data transfers can be performed simultaneously with transfers to and from the external memory, so it is possible to process large volumes of image data or voice data, etc., and through high-speed execution of instructions using internal ROM and RAM, motor control, communications control and other real-time control tasks can be realized simultaneously.

(3) On-chip flash memory (μPD70F3107)

The on-chip flash memory version (μPD70F3107) has on-chip flash memory, which is capable of high-speed access, and since it is possible to rewrite a program with the V850E/MA1 mounted as is in the application system, system development time can be reduced and system maintainability after shipping can be markedly improved.

(4) A full range of middleware and development environment products

The V850E/MA1 can execute middleware such as JPEG, JBIG, and MH/MR/MMR at high speed. Also, middleware that enables speech recognition, voice synthesis, and other such processing is available, and by including these middleware programs, a multimedia system can be easily realized.

A development environment system that includes an optimized C compiler, debugger, in-circuit emulator, simulator, system performance analyzer, and other elements is also available.

1.2 Features

- Number of instructions: 83
- Minimum instruction execution time: 20 ns (at internal 50 MHz operation)
- General-purpose registers: 32 bits × 32
- Instruction set:
  V850E1 CPU
  Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits): 1 to 2 clocks
  Saturated operation instructions (with overflow/underflow detection function)
  32-bit shift instructions: 1 clock
  Bit manipulation instructions
  Load/store instructions with long/short format
  Signed load instructions
- Memory space:
  256 MB linear address space (common program/data use)
  Chip select output function: 8 spaces
  Memory block division function: 2, 4, 8 MB/block
  Programmable wait function
  Idle state insertion function
- External bus interface:
  16-bit data bus (address/data separated)
  16-/8-bit bus sizing function
  Bus hold function
  External wait function
  Address setup wait function
  Endian control function
- Internal memory

Part Number	Internal ROM	Internal RAM
μPD703103	None	4 KB
μPD703105	128 KB (Mask ROM)	4 KB
μPD703106	128 KB (Mask ROM)	10 KB
μPD703107	256 KB (Mask ROM)	10 KB
μPD70F3107	256 KB (Flash memory)	10 KB

- Interrupts/exceptions:
  External interrupts: 25 (including NMI)
  Internal interrupts: 33 sources
  Exceptions: 1 source
  Eight levels of priorities can be set.
- Memory access controller
  DRAM controller (compatible with EDO DRAM and SDRAM)
  Page ROM controller

- DMA controller:
  - 4 channels
  - Transfer unit: 8 bits/16 bits
  - Maximum transfer count: 65,536 (2^{16})
  - Transfer type: Flyby (1-cycle)/2-cycle
  - Transfer mode: Single/Single step/Block
  - Transfer target: Memory ↔ memory, memory ↔ I/O
  - Transfer request: External request/On-chip peripheral I/O/ Software
  - DMA transfer terminate (terminal count) output signal
  - Next address setting function
- I/O lines:
  - Input ports: 9
  - I/O ports: 106
- Real-time pulse unit:
  - 16-bit timer/event counter: 4 channels
    - 16-bit timers: 4
    - 16-bit capture/compare registers: 8
  - 16-bit interval timer: 4 channels
- Serial interfaces (SIO):
  - Asynchronous serial interface (UART)
  - Clocked serial interface (CSI)
    - CSI/UART: 2 channels
    - UART: 1 channel
    - CSI: 1 channel
- A/D converter:
  - 10-bit resolution A/D converter: 8 channels
- PWM (Pulse Width Modulation):
  - 8-/9-/10-/12-bit resolution PWM: 2 channels
- Clock generator:
  - A ×10 function through a PLL clock synthesizer.
  - Divide-by-two function through an external clock input.
- Power-save function:
  - HALT/IDLE/software STOP mode
- Package:
  - 144-pin plastic LQFP (fine pitch) (20 × 20)
- ★ 161-pin plastic FBGA (13 × 13)
- CMOS technology:
  - All static circuits

1.3 Applications

Ink-jet printers, facsimiles, digital still cameras, DVD players, video printers, PPC, information equipment, etc.

★ 1.4 Ordering Information

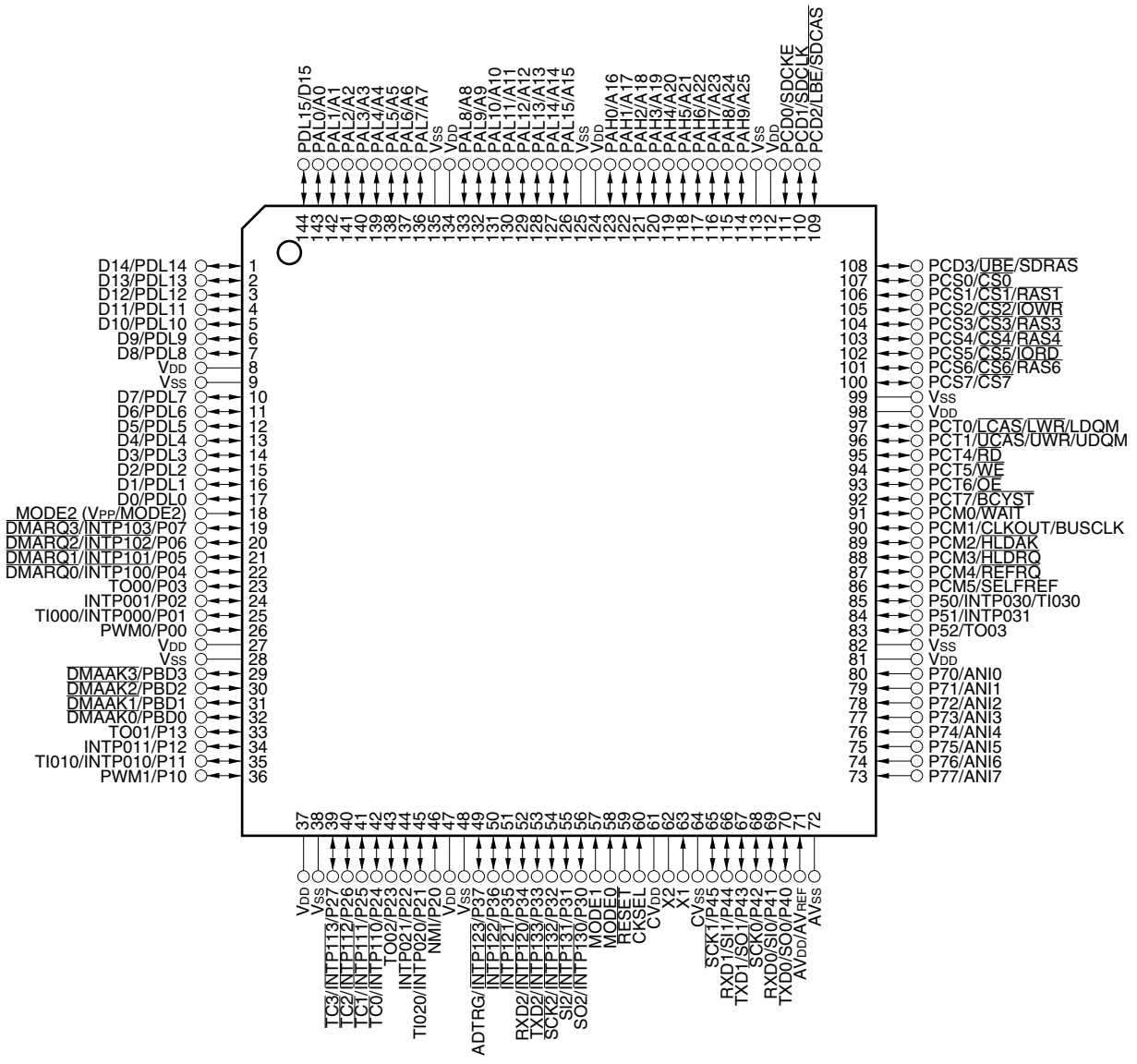
Part Number	Package	Internal ROM	Internal RAM
μPD703103GJ-UEN[Note]	144-pin plastic LQFP (fine pitch) (20 × 20)	None	4 KB
μPD703105GJ-×××-UEN[Note]	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (128 KB)	4 KB
μPD703106GJ-×××-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (128 KB)	10 KB
μPD703107GJ-×××-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)	10 KB
μPD70F3107GJ-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Flash memory (256 KB)	10 KB
μPD703106F1-×××-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (128 KB)	10 KB
μPD703107F1-×××-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)	10 KB
μPD70F3107F1-EN4[Note]	161-pin plastic FBGA (13 × 13)	Flash memory (256 KB)	10 KB

Note Under development

Remark ××× indicates ROM code suffix.

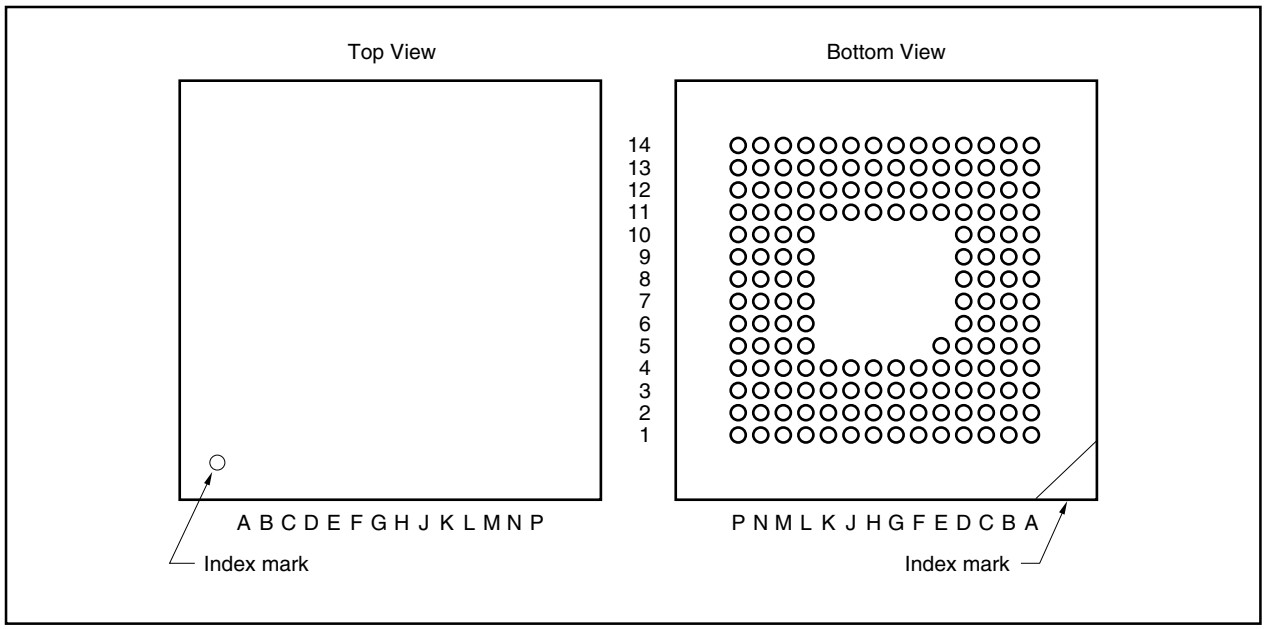
1.5 Pin Configuration (Top View)

- 144-pin plastic LQFP (fine pitch) (20 × 20)
  μPD703103GJ-UEN
  μPD703105GJ-×××-UEN
  μPD703106GJ-×××-UEN
  μPD703107GJ-×××-UEN
  μPD70F3107GJ-UEN



Remark Items in parentheses are pin names in the μPD70F3107.

★ • 161-pin plastic FBGA (13 × 13)
μPD703106F1-×××-EN4
μPD703107F1-×××-EN4
μPD70F3107F1-EN4



(1/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	–	B7	V_{SS}	C13	D9/PDL9
A2	D15/PDL15	B8	A13/PAL13	C14	–
A3	A2/PAL2	B9	A8/PAL8	D1	V_{SS}
A4	A5/PAL5	B10	V_{SS}	D2	D10/PDL10
A5	–	B11	A4/PAL4	D3	D14/PDL14
A6	A9/PAL9	B12	A0/PAL0	D4	A3/PAL3
A7	A12/PAL12	B13	D12/PDL12	D5	A6/PAL6
A8	A15/PAL15	B14	–	D6	A10/PAL10
A9	A17/PAH1	C1	–	D7	A14/PAL14
A10	–	C2	$\overline{CS2}$/$\overline{IOWR}$/PCS2	D8	A16/PAH0
A11	A24/PAH8	C3	$\overline{CS3}$/$\overline{RAS3}$/PCS3	D9	A20/PAH4
A12	V_{DD}	C4	V_{SS}	D10	A23/PAH7
A13	$\overline{LBE}$/$\overline{SDCAS}$/PCD2	C5	A22/PAH6	D11	SDCKE/PCD0
A14	$\overline{UBE}$/$\overline{SDRAS}$/PCD3	C6	A19/PAH3	D12	$\overline{CS0}$/PCS0
B1	–	C7	V_{DD}	D13	$\overline{CS5}$/$\overline{IORD}$/PCS5
B2	$\overline{CS1}$/$\overline{RAS1}$/PCS1	C8	A11/PAL11	D14	–
B3	SDCLK/PCD1	C9	V_{DD}	E1	D5/PDL5
B4	A25/PAH9	C10	A7/PAL7	E2	D7/PDL7
B5	A21/PAH5	C11	A1/PAL1	E3	D8/PDL8
B6	A18/PAH2	C12	D13/PDL13	E4	D11/PDL11

(2/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
E5	–	J12	TI030/INTP030/P50	M10	$\overline{SCK1}$/P45
E11	$\overline{CS6}$/$\overline{RAS6}$/PCS6	J13	SELFREF/PCM5	M11	TXD0/SO0/P40
E12	$\overline{CS4}$/$\overline{RAS4}$/PCS4	J14	INTP031/P51	M12	ANI6/P76
E13	$\overline{CS7}$/PCS7	K1	PWM0/P00	M13	ANI5/P75
E14	V_{SS}	K2	V_{SS}	M14	–
F1	D2/PDL2	K3	$\overline{DMAAK1}$/PBD1	N1	–
F2	D3/PDL3	K4	$\overline{DMAAK3}$/PBD3	N2	PWM1/P10
F3	D4/PDL4	K11	ANI1/P71	N3	$\overline{TC3}$/$\overline{INTP113}$/P27
F4	V_{DD}	K12	ANI0/P70	N4	$\overline{TC0}$/$\overline{INTP110}$/P24
F11	$\overline{RD}$/PCT4	K13	V_{SS}	N5	NMI/P20
F12	V_{DD}	K14	V_{DD}	N6	ADTRG/$\overline{INTP123}$/P37
F13	$\overline{LCAS}$/$\overline{LWR}$/LDQM/PCT0	L1	–	N7	TXD2/$\overline{INTP133}$/P33
F14	$\overline{UCAS}$/$\overline{UWR}$/UDQM/PCT1	L2	$\overline{DMAAK2}$/PBD2	N8	SO2/$\overline{INTP130}$/P30
G1	MODE2 (MODE2/V_{PP})	L3	TI010/INTP010/P11	N9	X2
G2	$\overline{DMARQ3}$/$\overline{INTP103}$/P07	L4	$\overline{DMAAK0}$/PBD0	N10	CV_{SS}
G3	D0/PDL0	L5	TO02/P23	N11	$\overline{SCK0}$/P42
G4	D6/PDL6	L6	V_{DD}	N12	AV_{DD}/AV_{REF}
G11	$\overline{WAIT}$/PCM0	L7	$\overline{INTP122}$/P36	N13	AV_{SS}
G12	$\overline{WE}$/PCT5	L8	SI2/$\overline{INTP131}$/P31	N14	–
G13	$\overline{BCYST}$/PCT7	L9	$\overline{RESET}$	P1	V_{DD}
G14	$\overline{OE}$/PCT6	L10	TXD1/SO1/P43	P2	V_{SS}
H1	$\overline{DMARQ2}$/$\overline{INTP102}$/P06	L11	ANI7/P77	P3	$\overline{TC1}$/$\overline{INTP111}$/P25
H2	$\overline{DMARQ1}$/$\overline{INTP101}$/P05	L12	ANI4/P74	P4	INTP021/P22
H3	$\overline{DMARQ0}$/$\overline{INTP100}$/P04	L13	ANI3/P73	P5	–
H4	D1/PDL1	L14	ANI2/P72	P6	$\overline{INTP121}$/P35
H11	$\overline{REFRQ}$/PCM4	M1	–	P7	$\overline{SCK2}$/$\overline{INTP132}$/P32
H12	$\overline{HLDRQ}$/PCM3	M2	INTP011/P12	P8	MODE1
H13	$\overline{HLDAK}$/PCM2	M3	TO01/P13	P9	CV_{DD}
H14	CLKOUT/BUSCLK/PCM1	M4	$\overline{TC2}$/$\overline{INTP112}$/P26	P10	X1
J1	TO00/P03	M5	TI020/INTP020/P21	P11	–
J2	TI000/INTP000/P01	M6	V_{SS}	P12	RXD1/SI1/P44
J3	V_{DD}	M7	RXD2/$\overline{INTP120}$/P34	P13	RXD0/SI0/P41
J4	INTP001/P02	M8	MODE0	P14	–
J11	TO03/P52	M9	CKSEL		

Remarks 1. Leave the A1, A5, A10, B1, B14, C1, C14, D14, E5, L1, M1, M14, N1, N14, P5, P11, and P14 pins open.

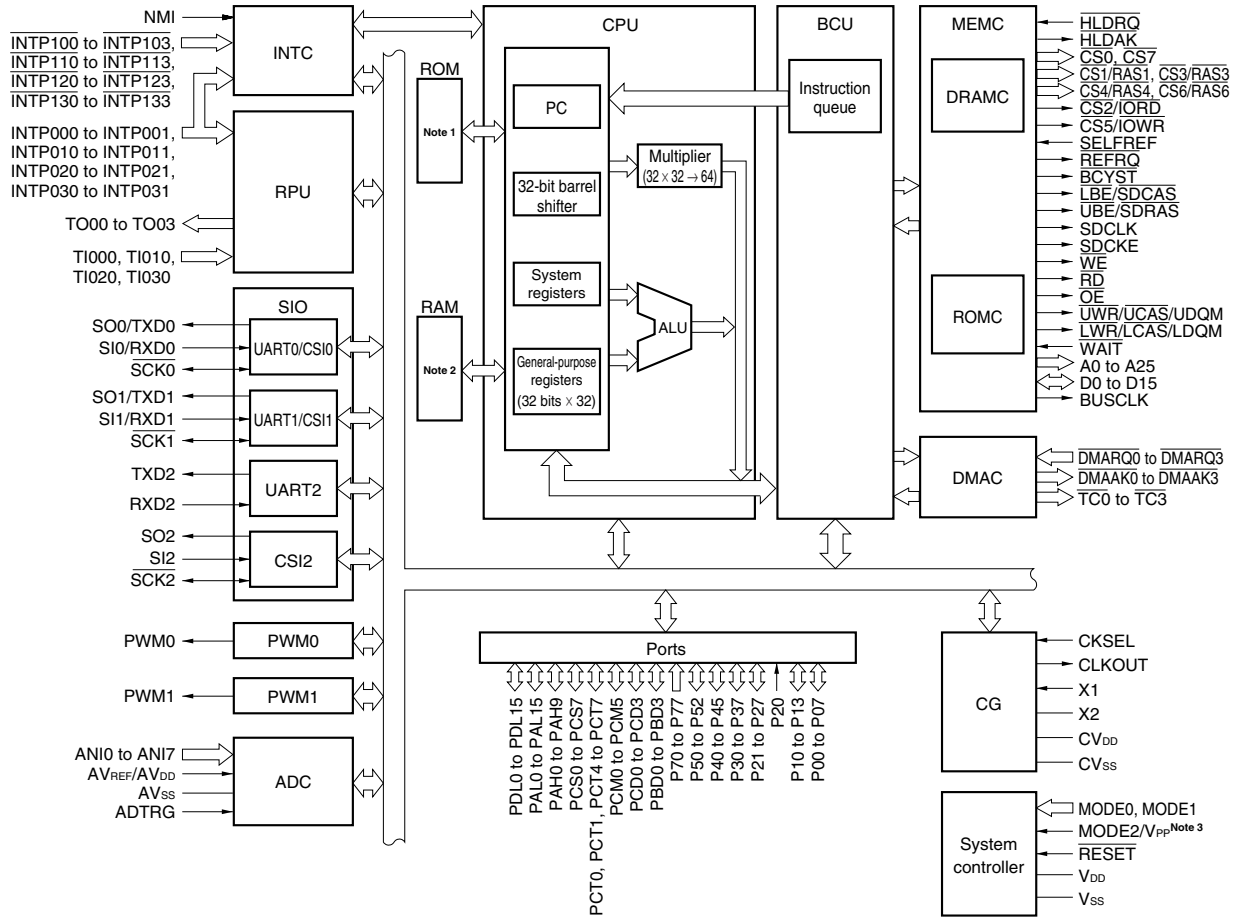
2. Items in parentheses are pin names in the μPD70F3107.

Pin Identification

A0 to A25:	Address bus	P70 to P77:	Port 7
ADTRG:	AD trigger input	PAH0 to PAH9:	Port AH
ANI0 to ANI7:	Analog input	PAL0 to PAL15:	Port AL
AV_{DD}:	Analog power supply	PBD0 to PBD3:	Port BD
AV_{REF}:	Analog reference voltage	PCD0 to PCD3:	Port CD
AV_{SS}:	Analog ground	PCM0 to PCM5:	Port CM
$\overline{BCYST}$:	Bus cycle start timing	PCS0 to PCS7:	Port CS
BUSCLK:	Bus clock output	PCT0, PCT1, :	Port CT
CKSEL:	Clock generator operating mode select	PCT4 to PCT7	
CLKOUT:	Clock output	PDL0 to PDL15:	Port DL
$\overline{CS0}$ to $\overline{CS7}$:	Chip select	PWM0, PWM1:	Pulse width modulation
CV_{DD}:	Clock generator power supply	$\overline{RAS1}$, $\overline{RAS3}$, :	Row address strobe
CV_{SS}:	Clock generator ground	$\overline{RAS4}$, $\overline{RAS6}$	
D0 to D15:	Data bus	$\overline{RD}$:	Read
$\overline{DMAAK0}$ to $\overline{DMAAK3}$:	DMA acknowledge	$\overline{REFRQ}$:	Refresh request
$\overline{DMARQ0}$ to $\overline{DMARQ3}$:	DMA request	$\overline{RESET}$:	Reset
$\overline{HLDAK}$:	Hold acknowledge	RXD0 to RXD2:	Receive data
$\overline{HLDRQ}$:	Hold request	$\overline{SCK0}$ to $\overline{SCK2}$:	Serial clock
INTP000, INTP001,	Interrupt request from peripherals	$\overline{SDCAS}$:	SDRAM column address strobe
INTP010, INTP011,		SDCKE:	SDRAM clock enable
INTP020, INTP021,		SDCLK:	SDRAM clock output
INTP030, INTP031,		$\overline{SDRAS}$:	SDRAM row address strobe
★ $\overline{INTP100}$ to $\overline{INTP103}$,		SELFREF:	Self-refresh request
★ $\overline{INTP110}$ to $\overline{INTP113}$,		SI0 to SI2:	Serial input
★ $\overline{INTP120}$ to $\overline{INTP123}$,		SO0 to SO2:	Serial output
★ $\overline{INTP130}$ to $\overline{INTP133}$		$\overline{TC0}$ to $\overline{TC3}$:	Terminal count signal
$\overline{IORD}$:	I/O read strobe	TI000, TI010, :	Timer input
$\overline{IOWR}$:	I/O write strobe	TI020, TI030:	
$\overline{LBE}$:	Lower byte enable	TO00 to TO03:	Timer output
$\overline{LCAS}$:	Lower column address strobe	TXD0 to TXD2:	Transmit data
LDQM:	Lower DQ mask enable	$\overline{UBE}$:	Upper byte enable
$\overline{LWR}$:	Lower write strobe	$\overline{UCAS}$:	Upper column address strobe
MODE0 to MODE2:	Mode	UDQM:	Upper DQ mask enable
NMI:	Non-maskable interrupt request	$\overline{UWR}$:	Upper write strobe
$\overline{OE}$:	Output enable	V_{DD}:	Power supply
P00 to P07:	Port 0	V_{PP}:	Programming power supply
P10 to P13:	Port 1	V_{SS}:	Ground
P20 to P27:	Port 2	$\overline{WAIT}$:	Wait
P30 to P37:	Port 3	$\overline{WE}$:	Write enable
P40 to P45:	Port 4	X1, X2:	Crystal
P50 to P52:	Port 5		

1.6 Function Blocks

1.6.1 Internal block diagram



Notes 1. μPD703103: ROMless
μPD703105, 703106: 128 KB (mask ROM)
μPD703107: 256 KB (mask ROM)
μPD70F3107: 256 KB (flash memory)

2. μPD703103, 703105: 4 KB
μPD703106, 703107, 70F3107: 10 KB

3. Valid for μPD70F3107 only

1.6.2 On-chip 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 a multiplier (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate processing of complex instructions.

(2) Bus control unit (BCU)

The BCU starts the required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory area 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 in the CPU.

The BCU controls a DRAM controller (DRAMC), page ROM controller (ROMC), and DMA controller (DMAC) and performs external memory access and DMA transfer.

(a) DRAM controller (DRAMC)

(i) SDRAM

The DRAM controller generates the $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, UDQM, and LDQM signals and performs access control for SDRAM.

CAS latency 2 and 3 are supported, and the burst length is fixed to 1.

A refresh function that supports the CBR refresh cycle and a dynamic self-refresh function based on an external input are also available.

(ii) EDO DRAM

The DRAM controller generates the $\overline{\text{RAS}}$, $\overline{\text{UCAS}}$, and $\overline{\text{LCAS}}$ signals (2CAS control) and performs access control for EDO DRAM.

EDO DRAM is supported, and there are two types of access: normal access (off page) and page access (on page).

A refresh function that supports the CBR refresh cycle and a dynamic self-refresh function based on an external input are also available.

(b) Page ROM controller (ROMC)

This controller supports accessing ROM that includes the page access function.

It performs address comparisons with the immediately preceding bus cycle and executes wait control for normal access (off-page)/page access (on-page). It can handle page widths of 8 to 128 bytes.

(c) DMA controller (DMAC)

This controller controls data transfer between memory and I/O instead of the CPU.

There are two address modes: flyby (1-cycle) transfer, and 2-cycle transfer. There are three bus modes, single transfer, single step transfer, and block transfer.

(3) ROM

The μPD703105 and 703106 have 128 KB of on-chip mask ROM, the μPD703107 has 256 KB of on-chip mask ROM and the μPD70F3107 has 256 KB of on-chip flash memory. The μPD703103 does not include on-chip ROM.

During instruction fetch, ROM/flash memory can be accessed from the CPU in 1-clock cycles.

If single-chip mode 0 or flash memory programming mode is set, memory mapping occurs from address 00000000H.

If single-chip mode 1 is set, memory mapping occurs from address 00100000H.

If ROMless mode is set, access is not possible.

(4) RAM

RAM is mapped from address FFFFC000H.

During data access, data can be accessed from the CPU in 1-clock cycles.

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0n0, INTP0n1, $\overline{\text{INTP1nn}}$) from on-chip peripheral I/O and external hardware (n = 0 to 3). Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed for interrupt sources.

(6) Clock generator (CG)

This clock generator supplies frequencies which are 10 times the input clock (fxx) (using an on-chip PLL) or 1/2 the input clock (when an on-chip PLL is not used) as the internal system clock (ϕ). As the input clock, an external oscillator is connected to pins X1 and X2 (only when an on-chip PLL synthesizer is used) or an external clock is input from the X1 pin.

(7) Real-time pulse unit (RPU)

This unit incorporates a 4-channel 16-bit timer/event counter and 4-channel 16-bit interval timer, and can measure pulse widths or frequency and output a programmable pulse.

(8) Serial interfaces (SIO)

The serial interfaces consist of 4 channels divided between an asynchronous serial interface (UART) and clocked serial interface (CSI). Two of these channels can be switched between UART and CSI, one channel is fixed to CSI, and the remaining channel is fixed to UART.

UART transfers data by using the TXDn and RXDn pins (n = 0 to 2).

CSI transfers data by using the SOn, SIn, and $\overline{\text{SCKn}}$ pins (n = 0 to 2).

(9) A/D converter (ADC)

This high-speed, high-resolution 10-bit A/D converter includes 8 analog input pins. Conversion is performed using the successive approximation method.

(10) PWM

Two channels for PWM signal output of 8-/9-/10-/12-bit resolution have been provided. By connecting an external low-pass filter, PWM output can be used as digital/analog conversion output. PWM is ideal for actuator control signals such as those in motors.

(11) Ports

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

Port	Port Function	Control Function
Port 0	8-bit I/O	Real-time pulse unit I/O, external interrupt input, PWM output, DMA controller input
Port 1	4-bit I/O	Real-time pulse unit I/O, external interrupt input, PWM output
Port 2	1-bit input, 7-bit I/O	NMI input, real-time pulse unit I/O, external interrupt input, DMA controller output
Port 3	8-bit I/O	Serial interface I/O, external interrupt input, A/D converter external trigger input
Port 4	6-bit I/O	Serial interface I/O
Port 5	3-bit I/O	Real-time pulse unit I/O, external interrupt input
Port 7	8-bit input	A/D converter input
Port AL	16-bit I/O	External address bus
Port AH	10-bit I/O	External address bus
★ Port DL	16-bit I/O	External data bus
Port CS	8-bit I/O	External bus interface control signal output
Port CT	6-bit I/O	External bus interface control signal output
Port CM	6-bit I/O	Wait insertion signal input, internal system clock output, external bus interface control signal I/O, self-refresh request signal input
Port CD	4-bit I/O	External bus interface control signal output
Port BD	4-bit I/O	DMA controller output

CHAPTER 2 PIN FUNCTIONS

The names and functions of the pins in the V850E/MA1 are listed below. These pins can be divided into port pins and non-port pins according to their functions.

2.1 List of Pin Functions

(1) Port pins

(1/3)

Pin Name	I/O	Function	Alternate Function
P00	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units.	PWM0
P01			TI000/INTP000
P02			INTP001
P03			TO00
P04			$\overline{\text{DMARQ0}}$/INTP100
P05			$\overline{\text{DMARQ1}}$/INTP101
P06			$\overline{\text{DMARQ2}}$/INTP102
P07			$\overline{\text{DMARQ3}}$/INTP103
P10	I/O	Port 1 4-bit I/O port Input/output can be specified in 1-bit units.	PWM1
P11			INTP010/TI010
P12			INTP011
P13			TO01
P20	Input	Port 2 P20 is an input-only port. If a valid edge is input, it operates as an NMI input. Also, the status of the NMI input is shown by bit 0 of the P2 register. P21 to P27 are a 7-bit I/O port. Input/output can be specified in 1-bit units.	NMI
P21	I/O		INTP020/TI020
P22			INTP021
P23			TO02
P24			$\overline{\text{TC0}}$/INTP110
P25			$\overline{\text{TC1}}$/INTP111
P26			$\overline{\text{TC2}}$/INTP112
P27			$\overline{\text{TC3}}$/INTP113
P30	I/O	Port 3 8-bit I/O port Input/output can be specified in 1-bit units.	SO2/$\overline{\text{INTP130}}$
P31			SI2/$\overline{\text{INTP131}}$
P32			$\overline{\text{SCK2}}$/$\overline{\text{INTP132}}$
P33			TXD2/$\overline{\text{INTP133}}$
P34			RXD2/$\overline{\text{INTP120}}$
P35			$\overline{\text{INTP121}}$
P36			$\overline{\text{INTP122}}$
P37			ADTRG/$\overline{\text{INTP123}}$

(2/3)

Pin Name	I/O	Function	Alternate Function
P40	I/O	Port 4 6-bit I/O port Input/output can be specified in 1-bit units.	TXD0/SO0
P41			RXD0/SI0
P42			$\overline{SCK0}$
P43			TXD1/SO1
P44			RXD1/SI1
P45			$\overline{SCK1}$
P50	I/O	Port 5 3-bit I/O port Input/output can be specified in 1-bit units.	INTP030/TI030
P51			INTP031
P52			TO03
P70 to P77	Input	Port 7 8-bit input-only port	ANI0 to ANI7
PBD0 to PBD3	I/O	Port BD 4-bit I/O port Input/output can be specified in 1-bit units.	$\overline{DMAAK0}$ to $\overline{DMAAK3}$
PCM0	I/O	Port CM 6-bit I/O port Input/output can be specified in 1-bit units.	$\overline{WAIT}$
PCM1			CLKOUT/BUSCLK
PCM2			$\overline{HLDAK}$
PCM3			$\overline{HLDRQ}$
PCM4			$\overline{REFRQ}$
PCM5			SELFREF
PCT0	I/O	Port CT 6-bit I/O port Input/output can be specified in 1-bit units.	$\overline{LCAS}$/$\overline{LWR}$/LDQM
PCT1			$\overline{UCAS}$/$\overline{UWR}$/UDQM
PCT4			$\overline{RD}$
PCT5			$\overline{WE}$
PCT6			$\overline{OE}$
PCT7			$\overline{BCYST}$
PCS0	I/O	Port CS 8-bit I/O port Input/output can be specified in 1-bit units.	$\overline{CS0}$
PCS1			$\overline{CS1}$/$\overline{RAS1}$
PCS2			$\overline{CS2}$/$\overline{IOWR}$
PCS3			$\overline{CS3}$/$\overline{RAS3}$
PCS4			$\overline{CS4}$/$\overline{RAS4}$
PCS5			$\overline{CS5}$/$\overline{IORD}$
PCS6			$\overline{CS6}$/$\overline{RAS6}$
PCS7			$\overline{CS7}$
PCD0	I/O	Port CD 4-bit I/O port Input/output can be specified in 1-bit units.	SDCKE
PCD1			SDCLK
PCD2			$\overline{LBE}$/$\overline{SDCAS}$
PCD3			$\overline{UBE}$/$\overline{SDRAS}$

(3/3)

Pin Name	I/O	Function	Alternate Function
PAH0 to PAH9	I/O	Port AH 8-/10-bit I/O port Input/output can be specified in 1-bit units.	A16 to A25
PAL0 to PAL15	I/O	Port AL 8-/16-bit I/O port Input/output can be specified in 1-bit units.	A0 to A15
PDL0 to PDL15	I/O	Port DL 8-/16-bit I/O port Input/output can be specified in 1-bit units.	D0 to D15

(2) Non-port pins

(1/4)

Pin Name	I/O	Function	Alternate Function
TO00	Output	Pulse signal output of timer C0 to C3	P03
TO01			P13
TO02			P23
TO03			P52
TI000	Input	External count input of timer C0 to C3	P01/INTP000
TI010			P11/INTP010
TI020			P21/INTP020
TI030			P50/INTP030
INTP000	Input	External maskable interrupt request input, or timer C0 external capture trigger input	P01/TI000
INTP001			P02
INTP010		External maskable interrupt request input, or timer C1 external capture trigger input	P11/TI010
INTP011			P12
INTP020		External maskable interrupt request input, or timer C2 external capture trigger input	P21/TI020
INTP021			P22
INTP030		External maskable interrupt request input, or timer C3 external capture trigger input	P50/TI030
INTP031			P51
★ $\overline{\text{INTP100}}$	Input	External maskable interrupt request input	P04/$\overline{\text{DMARQ0}}$
★ $\overline{\text{INTP101}}$			P05/$\overline{\text{DMARQ1}}$
★ $\overline{\text{INTP102}}$			P06/$\overline{\text{DMARQ2}}$
★ $\overline{\text{INTP103}}$			P07/$\overline{\text{DMARQ3}}$
★ $\overline{\text{INTP110}}$			P24/$\overline{\text{TC0}}$
★ $\overline{\text{INTP111}}$			P25/$\overline{\text{TC1}}$
★ $\overline{\text{INTP112}}$			P26/$\overline{\text{TC2}}$
★ $\overline{\text{INTP113}}$			P27/$\overline{\text{TC3}}$
★ $\overline{\text{INTP120}}$			P34/RXD2
★ $\overline{\text{INTP121}}$			P35
★ $\overline{\text{INTP122}}$			P36
★ $\overline{\text{INTP123}}$			P37/ADTRG
★ $\overline{\text{INTP130}}$			P30/SO2
★ $\overline{\text{INTP131}}$			P31/SI2
★ $\overline{\text{INTP132}}$			P32/$\overline{\text{SCK2}}$
★ $\overline{\text{INTP133}}$			P33/TXD2
SO0	Output	CSI0 to SCI2 serial transmission data output (3-wire)	P40/TXD0
SO1			P43/TXD1
SO2			P30/$\overline{\text{INTP130}}$

(2/4)

Pin Name	I/O	Function	Alternate Function
SI0	Input	CSI0 to CSI2 serial reception data input (3-wire)	P41/RXD0
SI1			P44/RXD1
SI2			P31/$\overline{\text{INTP131}}$
$\overline{\text{SCK0}}$	I/O	CSI0 to CSI2 serial clock I/O (3-wire)	P42
$\overline{\text{SCK1}}$			P45
$\overline{\text{SCK2}}$			P32/$\overline{\text{INTP132}}$
TXD0	Output	UART0 to UART2 serial transmission data output	P40/SO0
TXD1			P43/SO1
TXD2			P33/$\overline{\text{INTP133}}$
RXD0	Input	UART0 to UART2 serial reception data input	P41/SI0
RXD1			P44/SI1
RXD2			P34/$\overline{\text{INTP120}}$
PWM0	Output	PWM pulse signal output	P00
PWM1			P10
ANI0 to ANI7	Input	Analog inputs to A/D converter	P70 to P77
ADTRG	Input	A/D converter external trigger input	P37/$\overline{\text{INTP123}}$
$\overline{\text{DMARQ0}}$	Input	DMA request signal input	P04/$\overline{\text{INTP100}}$
$\overline{\text{DMARQ1}}$			P05/$\overline{\text{INTP101}}$
$\overline{\text{DMARQ2}}$			P06/$\overline{\text{INTP102}}$
$\overline{\text{DMARQ3}}$			P07/$\overline{\text{INTP103}}$
$\overline{\text{DMAAK0}}$	Output	DMA acknowledge signal output	PBD0
$\overline{\text{DMAAK1}}$			PBD1
$\overline{\text{DMAAK2}}$			PBD2
$\overline{\text{DMAAK3}}$			PBD3
$\overline{\text{TC0}}$	Output	DMA transfer end (terminal count) signal output	P24/$\overline{\text{INTP110}}$
$\overline{\text{TC1}}$			P25/$\overline{\text{INTP111}}$
$\overline{\text{TC2}}$			P26/$\overline{\text{INTP112}}$
$\overline{\text{TC3}}$			P27/$\overline{\text{INTP113}}$
NMI	Input	Non-maskable interrupt request signal input	P20
MODE0	Input	V850E/MA1 operating mode specification	–
MODE1			–
MODE2			V_{PP}
V_{PP}	Input	Flash memory programming power-supply application pin (μPD70F3107 only)	MODE2
$\overline{\text{WAIT}}$	Input	Control signal input that inserts a wait in the bus cycle	PCM0
$\overline{\text{HLDAK}}$	Output	Bus hold acknowledge output	PCM2
$\overline{\text{HLDRQ}}$	Input	Bus hold request input	PCM3
$\overline{\text{REFRQ}}$	Output	Refresh request signal output for DRAM	PCM4
SELFREF	Input	Self-refresh request input for DRAM	PCM5

(3/4)

Pin Name	I/O	Function	Alternate Function
$\overline{\text{LCAS}}$	Output	Column address strobe signal output for DRAM lower data	PCT0/$\overline{\text{LWR}}$/LDQM
$\overline{\text{UCAS}}$	Output	Column address strobe signal output for DRAM higher data	PCT1/$\overline{\text{UWR}}$/UDQM
$\overline{\text{LWR}}$	Output	External data lower byte write enable signal output	PCT0/$\overline{\text{LCAS}}$/LDQM
$\overline{\text{UWR}}$	Output	External data higher byte write enable signal output	PCT1/$\overline{\text{UCAS}}$/UDQM
LDQM	Output	Output disable/write mask signal output for SDRAM lower data	PCT0/$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$
UDQM	Output	Output disable/write mask signal output for SDRAM higher data	PCT1/$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$
$\overline{\text{RD}}$	Output	External data bus read strobe signal output	PCT4
$\overline{\text{WE}}$	Output	Write enable signal output for DRAM	PCT5
$\overline{\text{OE}}$	Output	Output enable signal output for DRAM	PCT6
$\overline{\text{BCYST}}$	Output	Strobe signal output that shows the start of the bus cycle	PCT7
$\overline{\text{CS0}}$	Output	Chip select signal output	PCS0
$\overline{\text{CS1}}$			PCS1/$\overline{\text{RAS1}}$
$\overline{\text{CS2}}$			PCS2/$\overline{\text{IOWR}}$
$\overline{\text{CS3}}$			PCS3/$\overline{\text{RAS3}}$
$\overline{\text{CS4}}$			PCS4/$\overline{\text{RAS4}}$
$\overline{\text{CS5}}$			PCS5/$\overline{\text{IORD}}$
$\overline{\text{CS6}}$			PCS6/$\overline{\text{RAS6}}$
$\overline{\text{CS7}}$			PCS7
$\overline{\text{RAS1}}$	Output	Row address strobe signal output for DRAM	PCS1/$\overline{\text{CS1}}$
$\overline{\text{RAS3}}$			PCS3/$\overline{\text{CS3}}$
$\overline{\text{RAS4}}$			PCS4/$\overline{\text{CS4}}$
$\overline{\text{RAS6}}$			PCS6/$\overline{\text{CS6}}$
$\overline{\text{IOWR}}$	Output	DMA write strobe signal output	PCS2/$\overline{\text{CS2}}$
$\overline{\text{IORD}}$	Output	DMA read strobe signal output	PCS5/$\overline{\text{CS5}}$
SDCKE	Output	SDRAM clock enable signal output	PCD0
SDCLK	Output	SDRAM clock signal output	PCD1
$\overline{\text{SDCAS}}$	Output	Column address strobe signal output for SDRAM	PCD2/$\overline{\text{LBE}}$
$\overline{\text{SDRAS}}$	Output	Row address strobe signal output for SDRAM	PCD3/$\overline{\text{UBE}}$
$\overline{\text{LBE}}$	Output	External data bus lower byte enable signal output	PCD2/$\overline{\text{SDCAS}}$
$\overline{\text{UBE}}$	Output	External data bus higher byte enable signal output	PCD3/$\overline{\text{SDRAS}}$
D0 to D15	I/O	16-bit data bus for external memory	PDL0 to PDL15
A0 to A15	Output	26-bit address bus for external memory	PAL0 to PAL15
A16 to A25			PAH0 to PAH9
$\overline{\text{RESET}}$	Input	System reset input	–
X1	Input	Connects the crystal resonator for system clock oscillation. In the case of an external source supplying the clock, it is input to X1.	–
X2	–		–
CLKOUT	Output	System clock output	PCM1/BUSCLK
BUSCLK	Output	Bus clock output	PCM1/CLKOUT

(4/4)

Pin Name	I/O	Function	Alternate Function
CKSEL	Input	Input specifying the clock generator's operating mode	–
AV_{REF}	Input	Reference voltage applied to A/D converter	AV_{DD}
AV_{DD}	–	Positive power supply for A/D converter	AV_{REF}
AV_{SS}	–	Ground potential for A/D converter	–
CV_{DD}	–	Positive power supply for dedicated clock generator	–
CV_{SS}	–	Ground potential for dedicated clock generator	–
V_{DD}	–	Positive power supply	–
V_{SS}	–	Ground potential	–

2.2 Pin Status

The status of each pin after reset, in power-save mode (software STOP, IDLE, HALT modes), and during DMA transfer, refresh, and bus hold (TH) is shown below.

Operating Status / Pin	Reset (Single-Chip Mode 0)	Reset (Single-Chip Mode 1, ROMless Mode 0,1)	IDLE Mode/Software STOP Mode	HALT Mode/During DMA Transfer, Refresh	Bus Hold (TH)
A0 to A15 (PAL0 to PAL15)	Hi-Z	Hi-Z	Hi-Z	Operating	Hi-Z
A16 to A25 (PAH0 to PAH9)	Hi-Z	Hi-Z	Hi-Z	Operating	Hi-Z
D0 to D15 (PDL0 to PDL15)	Hi-Z	Hi-Z	Hi-Z	Operating	Hi-Z
$\overline{CS0}$ to $\overline{CS7}$ (PCS0 to PCS7)	Hi-Z	Hi-Z	SELF	Operating	Hi-Z
$\overline{RAS1}$, $\overline{RAS3}$, $\overline{RAS4}$, $\overline{RAS6}$ (PCS1, PCS3, PCS4, PCS6)	–	–	CBR	Operating	Hi-Z
$\overline{IOWR}$ (PCS2)	–	–	H	Operating	Hi-Z
$\overline{IORD}$ (PCS5)	–	–	H	Operating	Hi-Z
$\overline{LWR}$, $\overline{UWR}$ (PCT0, PCT1)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{LCAS}$, $\overline{UCAS}$ (PCT0, PCT1)	–	–	CBR	Operating	Hi-Z
LDQM, UDQM (PCT0, PCT1)	–	–	H	Operating	Hi-Z
$\overline{RD}$ (PCT4)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{WE}$ (PCT5)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{OE}$ (PCT6)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{BCYST}$ (PCT7)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{WAIT}$ (PCM0)	Hi-Z	Hi-Z	–	Operating	–
CLKOUT (PCM1)	Hi-Z	Operating	L	Operating	Operating
BUSCLK (PCM1)	–	–	L	Operating	Operating
$\overline{HLDAK}$ (PCM2)	Hi-Z	Hi-Z	H	Operating	L
$\overline{HLDRQ}$ (PCM3)	Hi-Z	Hi-Z	–	Operating	Operating
$\overline{REFRQ}$ (PCM4)	Hi-Z	Hi-Z	CBR	Operating	Operating
SELFREF (PCM5)	Hi-Z	Hi-Z	–	Operating	–
SDCKE (PCD0)	Hi-Z	Hi-Z	L	Operating	Operating
SDCLK (PCD1)	Hi-Z	Hi-Z	L	Operating	Operating
$\overline{SDCAS}$ (PCD2)	–	–	SELF	Operating	Hi-Z
$\overline{LBE}$ (PCD2)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{SDRAS}$ (PCD3)	–	–	SELF	Operating	Hi-Z
$\overline{UBE}$ (PCD3)	Hi-Z	Hi-Z	H	Operating	Hi-Z
$\overline{DMAAK0}$ to $\overline{DMAAK3}$ (PBD0 to PBD3)	Hi-Z	Hi-Z	H	Operating	H

Remark Hi-Z: High-impedance
Hold: Status during immediately preceding external bus cycle hold
H: High-level output
L: Low-level output
—: No sampling of input
CBR: A DRAM refresh state
SELF: Self-refresh state when pins are connected to SDRAM

2.3 Description of Pin Functions

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

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

Besides functioning as an I/O port, in the control mode, these pins operate as I/O for the real-time pulse unit (RPU), external interrupt request inputs, a PWM output, and DMA request inputs.

The operation mode can be set to port or control mode in 1-bit units, specified by the port 0 mode control register (PMC0).

(a) Port mode

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

(b) Control mode

P00 to P07 can be set to port/control mode in 1-bit units using the PMC0 register.

(i) PWM0 (Pulse width modulation) ··· output

This pin outputs the PWM pulse signal.

(ii) TI000 (Timer input) ··· input

This is the external count clock input pin for timer C0.

(iii) TO00 (Timer output) ··· output

This pin outputs the pulse signals for timer C0.

(iv) INTP000, INTP001 (Interrupt request from peripherals) ··· input

These are external interrupt request input pins and the external capture trigger input pins for timer C0.

(v) $\overline{\text{INTP100}}$ to $\overline{\text{INTP103}}$ (Interrupt request from peripherals) ··· input

These are external interrupt request input pins.

(vi) $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ (DMA request) ··· input

These are DMA service request signals. They correspond to DMA channels 0 to 3, respectively, and operate independently of each other. The priority order is fixed to $\overline{\text{DMARQ0}}$ > $\overline{\text{DMARQ1}}$ > $\overline{\text{DMARQ2}}$ > $\overline{\text{DMARQ3}}$.

These signals are sampled at the falling edge of the CLKOUT signal. Maintain an active level until a DMA request is acknowledged.

(2) P10 to P13 (Port 1) ··· 3-state I/O

P10 to P13 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, in the control mode, these pins operate as I/O for the real-time pulse unit (RPU), external interrupt request inputs, and a PWM output.

The operation mode can be set to port or control mode in 1-bit units, specified by the port 1 mode control register (PMC1).

(a) Port mode

P10 to P13 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(b) Control mode

P10 to P13 can be set to port/control mode in 1-bit units using the PMC1 register.

(i) PWM1 (Pulse width modulation) ··· output

This pin outputs the PWM pulse signal.

(ii) TI010 (Timer input) ··· input

This is the external count clock input pin for timer C1.

(iii) TO01 (Timer output) ··· output

This pin outputs the pulse signal for timer C1.

(iv) INTP010, INTP011 (Interrupt request from peripherals) ··· input

These are external interrupt request input pins and the external capture trigger input pins for timer C1.

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

P20 is an input-only pin. P21 to P27 function as a 7-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, in the control mode, these pins operate as I/O for the real-time pulse unit (RPU), external interrupt request inputs, and DMA transfer termination outputs (terminal count).

The operation mode can be set to port or control mode in 1-bit units, specified by the port 2 mode control register (PMC2).

(a) Port mode

P21 to P27 can be set to input or output in 1-bit units using the port 2 mode register (PM2). P20 is an input-only port, and if a valid edge is input, it operates as an NMI input.

(b) Control mode

P21 to P27 can be set to port/control mode in 1-bit units using the PMC2 register.

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

This is the non-maskable interrupt request input pin.

(ii) TI020 (Timer input) ··· input

This is the external count clock input pin for timer C2.

(iii) TO02 (Timer output) ··· output

This pin outputs the pulse signal for timer C2.

(iv) INTP020, INTP021 (Interrupt request from peripherals) ··· input

These are external interrupt request input pins and the external capture trigger input pins for timer C2.

(v) $\overline{\text{INTP110}}$ to $\overline{\text{INTP113}}$ (Interrupt request from peripherals) ··· input

These are external interrupt request input pins.

(vi) $\overline{\text{TC0}}$ to $\overline{\text{TC3}}$ (Terminal count) ··· output

These are signals from the DMA controller indicating that DMA transfer is complete. These signals become active for 1 clock at the rising edge of the CLKOUT signal.

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

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

Besides functioning as an I/O port, in the control mode, these pins operate as I/O for the serial interfaces (CSI2, UART2), external interrupt request inputs, and the A/D converter external trigger input.

The operation mode can be set to port or control mode in 1-bit units, specified by the port 3 mode control register (PMC3).

(a) Port mode

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

(b) Control mode

P30 to P37 can be set to port/control mode in 1-bit units using the PMC3 register.

(i) TXD2 (Transmit data) ··· output

This pin outputs the serial transmit data of UART2.

(ii) RXD2 (Receive data) ··· input

This pin inputs the serial receive data of UART2.

(iii) SO2 (Serial output) ··· output

This pin outputs the serial transmit data of CSI2.

(iv) SI2 (Serial input) ··· input

This pin inputs the serial receive data of CSI2.

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

This is the CSI2 serial clock I/O pin.

(vi) $\overline{\text{INTP120}}$ to $\overline{\text{INTP123}}$, $\overline{\text{INTP130}}$ to $\overline{\text{INTP133}}$ (Interrupt request from peripherals) ··· input

These are external interrupt request input pins.

(vii) ADTRG (AD trigger input) ··· input

This is the external trigger input pin for the AD converter.

(5) P40 to P45 (Port 4) ··· 3-state I/O

P40 to P45 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, in the control mode, these pins operate as I/O for the serial interfaces (UART0/CSI0, UART1/CSI1).

The operation mode can be set to port or control mode in 1-bit units, specified by the port 4 mode control register (PMC4).

(a) Port mode

P40 to P45 can be set to input or output in 1-bit units using the port 4 mode register (PM4).

(b) Control mode

P40 to P45 can be set to port/control mode in 1-bit units using the PMC4 register.

(i) TXD0, TXD1 (Transmit data) ··· output

These pins output UART0, UART1 serial transmit data.

(ii) RXD0, RXD1 (Receive data) ··· input

These pins input UART0, UART1 serial receive data.

(iii) SO0, SO1 (Serial output) ··· output

These pins output CSI0, CSI1 serial transmit data.

(iv) SI0, SI1 (Serial input) ··· input

These pins input CSI0, CSI1 serial receive data.

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

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

(6) P50 to P52 (Port 5) ··· 3-state I/O

P50 to P52 function as a 3-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, in the control mode, these pins operate as I/O for the real-time pulse unit (RPU) and external interrupt request inputs.

The operation mode can be set to port or control mode in 1-bit units, specified by the port 5 mode control register (PMC5).

(a) Port mode

P50 to P52 can be set to input or output in 1-bit units using the port 5 mode register (PM5).

(b) Control mode

P50 to P52 can be set to port/control mode in 1-bit units using the PMC5 register.

(i) TI030 (Timer input) ··· input

This is the external count clock input pin for timer C3.

(ii) TO03 (Timer output) ··· output

This pin outputs the pulse signal for timer C3.

(iii) INTP030, INTP031 (Interrupt request from peripherals) ··· input

These are external interrupt request input pins and the external capture trigger input pins for timer C3.

(7) P70 to P77 (Port 7) ··· 3-state I/O

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

Besides functioning as a port, in the control mode, these pins operate as analog inputs for the A/D converter. However, the input ports and analog input pins cannot be switched.

(a) Port mode

P70 to P77 are input-only pins.

(b) Control mode

P70 to P77 have alternate functions as pins ANI0 to ANI7, but these alternate functions are not switchable.

(i) ANI0 to ANI7 (Analog input) ··· 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.

(8) PBD0 to PBD3 (Port BD) ··· 3-state I/O

PBD0 to PBD3 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, in the control mode, these pins operate as DMA acknowledge outputs.

The operation mode can be set to port or control in 1-bit units, specified by the port BD mode control register (PMCBD).

(a) Port mode

PBD0 to PBD3 can be set to input or output in 1-bit units using the port BD mode register (PMBD).

(b) Control mode

PBD0 to PBD3 can be set to port/control mode in 1-bit units using the PMCBD register.

(i) $\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$ (DMA acknowledge) ··· output

These signals show that a DMA service request was granted. They correspond to DMA channel 0 to 3, respectively, and operate independently of each other.

These signals become active only when external memory is being accessed. When DMA transfers are being executed between internal RAM and internal peripheral I/O, they do not become active.

These signals are activated at the falling edge of the CLKOUT signal in the T0, T1R, T1FH state of the DMA cycle, and maintained at an active level during DMA transfers.

(9) PCM0 to PCM5 (Port CM) ··· 3-state I/O

PCM0 to PCM5 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode, these pins operate as the wait insertion signal input, system clock output, bus hold control signal, refresh request signal output for DRAM, and self-refresh request signal input.

The operation mode can be set to port or control in 1-bit units, specified by the port CM mode control register (PMCCM).

(a) Port mode

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

(b) Control mode

PCM0 to PCM5 can be set to port/control mode in 1-bit units using the PMCCM register.

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

This is the control signal input pin at which a data wait is inserted in the bus cycle. The $\overline{\text{WAIT}}$ signal can be input asynchronously to the CLKOUT signal. When the CLKOUT signal falls, sampling is executed. When the set/hold time is not terminated within the sampling timing, wait insertion may not be executed.

(ii) CLKOUT (Clock output) ··· output

This is the internal system clock output pin. In single-chip mode 0, because port mode is entered during the reset period, output does not occur from the CLKOUT pin. CLKOUT output can be executed by setting the port CM mode control register (PMCCM) and the port CM function control register (PFCCM).

(iii) BUSCLK (Bus clock output) ··· output

This is the bus clock output pin. The bus clock operates at the operating frequency of 1/2 the internal system clock by setting the bus cycle period control register (BCP). To execute BUSCLK output, set the port CM mode control register (PMCCM) and the port CM function control register (PFCCM).

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

In this mode, this pin is the acknowledge signal output pin that indicates the high impedance status for the address bus, data bus, and control bus when the V850E/MA1 receives a bus hold request.

While this signal is active, the impedance of the address bus, data bus and control bus becomes high and the bus mastership is transferred to the external bus master.

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

In this mode, this pin is the input pin through which an external device requests the V850E/MA1 to release the address bus, data bus, and control bus. The $\overline{\text{HLDRQ}}$ signal can be input asynchronously to the CLKOUT signal. When this pin is active, the address bus, data bus, and control bus are set to the high impedance status. This occurs either when the V850E/MA1 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.

In order to make the bus hold state secure, keep the $\overline{\text{HLDRQ}}$ signal active until the $\overline{\text{HLDAK}}$ signal is output.

(vi) $\overline{\text{REFRQ}}$ (Refresh request) ··· output

This is the refresh request signal for DRAM.

In cases when the address is decoded by an external circuit to increase the connected DRAM, or in cases when external SIMM's are connected, this signal is used for RAS control during the refresh cycle.

This signal becomes active during the refresh cycle. Also, during bus hold, it becomes active when a refresh request is generated and informs the external bus master that a refresh request was generated.

(vii) SELFREF (Self-refresh request) ··· input

This is a self-refresh request signal input for DRAM.

The internal ROM and internal RAM can be accessed even in the self-refresh cycle. However, access to a peripheral I/O register or external device is held pending until the self-refresh cycle is cancelled.

(10) PCT0, PCT1, PCT4 to PCT7 (Port CT) ··· 3-state I/O

PCT0, PCT1, PCT4 to PCT7 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode, these pins operate as control signal outputs for when memory is expanded externally.

The operation mode can be set to port or control mode in 1-bit units, specified by the port CT mode control register (PMCCT).

(a) Port mode

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

(b) Control mode

PCT0, PCT1, PCT4 to PCT7 can be set to port/control mode in 1-bit units using the PMCCT register.

(i) $\overline{\text{LCAS}}$ (Lower column address strobe) ··· 3-state output

This is the column address strobe signal for DRAM and the strobe signal for the CBR refresh cycle. For the data bus, the lower byte is valid.

(ii) $\overline{\text{UCAS}}$ (Upper column address strobe) ··· 3-state output

This is the column address strobe signal for DRAM and the strobe signal for the CBR refresh cycle. For the data bus, the higher byte is valid.

(iii) $\overline{\text{LWR}}$ (Lower byte write strobe) ··· 3-state output

This strobe signal shows whether the bus cycle currently being executed is a write cycle for the SRAM, external ROM, or external peripheral I/O area.

For the data bus, the lower byte becomes valid. If the bus cycle is a lower memory write, it becomes active at the falling edge of the CLKOUT signal in the T1 state and becomes inactive at the falling edge of the CLKOUT signal in the T2 state.

(iv) $\overline{\text{UWR}}$ (Upper byte write strobe) ··· 3-state output

This strobe signal shows whether the bus cycle currently being executed is a write cycle for the SRAM, external ROM, or external peripheral I/O area.

For the data bus, the higher byte becomes valid. If the bus cycle is a higher memory write, it becomes active at the falling edge of the CLKOUT signal in the T1 state and becomes inactive at the falling edge of the CLKOUT signal in the T2 state.

(v) LDQM (Lower DQ mask enable) ··· 3-state output

This is a control signal for the data bus to SDRAM. For the data bus, the lower byte is valid. This signal carries out SDRAM output disable control during a read operation, and SDRAM byte mask control during a write operation.

(vi) UDQM (Upper DQ mask enable) ··· 3-state output

This is a control signal for the data bus to SDRAM. For the data bus, the higher byte is valid. This signal carries out SDRAM output disable control during a read operation, and SDRAM byte mask control during a write operation.

(vii) $\overline{RD}$ (Read strobe) ··· 3-state output

This strobe signal shows that the bus cycle currently being executed is a read cycle for the SRAM, external ROM, external peripheral I/O, or page ROM area. In the idle state (TI), it becomes inactive.

(viii) $\overline{WE}$ (Write enable) ··· 3-state output

This signal shows that the bus cycle currently being executed is a write cycle for the DRAM area. In the idle state (TI), it becomes inactive.

(ix) $\overline{OE}$ (Output enable) ··· 3-state output

This signal shows that the bus cycle currently being executed is a read cycle for the DRAM area. In the idle state (TI), it becomes inactive.

(x) $\overline{BCYST}$ (Bus cycle start timing) ··· 3-state output

This outputs a status signal showing the start of the bus cycle. It becomes active for 1-clock cycle from the start of each cycle. In the idle state (TI), it becomes inactive.

(11) PCS0 to PCS7 (Port CS) ··· 3-state I/O

PCS0 to PCS7 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode, these pins operate as control signal outputs for when memory and peripheral I/O are expanded externally.

The operation mode can be set to port or control mode in 1-bit units, specified by the port CS mode control register (PMCCS).

(a) Port mode

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

(b) Control mode

PCS0 to PCS7 can be set to port/control mode in 1-bit units using the PMCCS register.

(i) $\overline{CS0}$ to $\overline{CS7}$ (Chip select) ··· 3-state output

These are the chip select signals for the SRAM, external ROM, external peripheral I/O, and page ROM area.

The $\overline{CSn}$ signal is assigned to memory block n (n = 0 to 7).

It becomes active while the bus cycle that accesses the corresponding memory block is activated.

In the idle state (TI), it becomes inactive.

(ii) $\overline{RAS1}$, $\overline{RAS3}$, $\overline{RAS4}$, $\overline{RAS6}$ (Row address strobe) ··· 3-state output

These are the row address strobe signals for the DRAM area and the strobe signal for the refresh cycle.

The $\overline{RASn}$ signal is assigned to memory block n (n = 1, 3, 4, 6).

During on-page disable, after the DRAM access bus cycle ends, it becomes inactive.

During on-page enable, even after the DRAM access bus cycle ends, it remains in the active state.

During the reset period and during a bus hold period, it is in the high-impedance state, so connect it to V_{DD} via a resistor.

(iii) $\overline{IOWR}$ (I/O write) ··· 3-state output

This is the write strobe signal for external I/O during DMA flyby transfer. It indicates whether the bus cycle currently being executed is a write cycle for external I/O during flyby transfer, or a write cycle for the SRAM area.

Note that if the IOEN bit of the BCP register is set (1), this signal can be output even in the normal SRAM, external ROM, or external I/O cycle.

(iv) $\overline{IORD}$ (I/O read) ··· 3-state output

This is the read strobe signal for external I/O during DMA flyby transfer. It indicates whether the bus cycle currently being executed is a read cycle for external I/O during flyby transfer, or a read cycle for the SRAM area.

Note that if the IOEN bit of the BCP register is set (1), this signal can be output even in the normal SRAM, external ROM, or external I/O cycle.

(12) PCD0 to PCD3 (Port CD) ··· 3-state I/O

PCD0 to PCD3 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in control mode, these pins operate as control signal outputs for when the memory and peripheral I/O are expanded externally.

The operation mode can be set to port or control mode in 1-bit units, specified by the port CD mode control register (PMCCD).

(a) Port mode

PCD0 to PCD3 can be set to input or output in 1-bit units using the port CD mode register (PMCD).

(b) Control mode

PCD0 to PCD3 can be set to port or control mode in 1-bit units using the PMCCD register.

(i) SDCKE (SDRAM clock enable) ··· 3-state output

This is the SDRAM clock enable output signal. It becomes inactive in self-refresh and standby mode.

(ii) SDCLK (SDRAM clock output) ··· 3-state output

★ This is an SDRAM dedicated clock output signal. The same frequency as the internal system clock is output.

(iii) $\overline{\text{SDCAS}}$ (SDRAM column address strobe) ··· 3-state output

This is a command output signal for SDRAM.

(iv) $\overline{\text{SDRAS}}$ (SDRAM row address strobe) ··· 3-state output

This is a command output signal for SDRAM.

(v) $\overline{\text{LBE}}$ (Lower byte enable) ··· 3-state output

This is the signal that enables the lower byte of the external data bus.

(vi) $\overline{\text{UBE}}$ (Upper byte enable) ··· 3-state output

This is the signal that enables the higher byte of the external data bus.

(13) PAH0 to PAH9 (Port AH) ··· 3-state I/O

PAH0 to PAH9 function as an 8- or 10-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in control mode (external expansion mode), these pins operate as an address bus (A16 to A25) for when the memory is expanded externally.

The operation mode can be set to port or control mode in 1-bit units, specified by the port AH mode control register (PMCAH).

(a) Port mode

PAH0 to PAH9 can be set to input or output in 1-bit units using the port AH mode register (PMAH).

(b) Control mode

PAH0 to PAH9 can be set to function alternately as A16 to A25 using the PMCAH register.

(i) A16 to A25 (Address) ··· 3-state output

These are the address output pins of the higher 10 bits of the address bus's 26-bit address when the external memory is accessed.

The output changes in synchronization with the fall of the CLKOUT signal in the T1 state. In the idle state (TI), the address of the bus cycle immediately before is retained.

(14) PAL0 to PAL15 (Port AL) ··· 3-state I/O

PAL0 to PAL15 function as an 8- or 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in control mode (external expansion mode), these pins operate as an address bus (A0 to A15) for when the memory is expanded externally.

The operation mode can be set to port or control mode in 1-bit units, specified by the port AL mode control register (PMCAL).

(a) Port mode

PAL0 to PAL15 can be set to input or output in 1-bit units using the port AL mode register (PMAL).

(b) Control mode

PAL0 to PAL15 can be set to function alternately as A0 to A15 using the PMCAL register.

(i) A0 to A15 (Address) ··· 3-state output

These are the address output pins of the lower 16 bits of the address bus's 26-bit address when the external memory is accessed.

The output changes in synchronization with the fall of the CLKOUT signal in the T1 state. In the idle state (TI), the address of the bus cycle immediately before is retained.

(15) PDL0 to PDL15 (Port DL) ··· 3-state I/O

PDL0 to PDL15 function as an 8- or 16-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as a port, in control mode (external expansion mode), these pins operate as a data bus (D0 to D15) for when the memory is expanded externally.

The operation mode can be set to port or control mode in 1-bit units, specified by the port DL mode control register (PMCDL).

(a) Port mode

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

(b) Control mode

PDL0 to PDL15 can be set to function alternately as D0 to D15 using the PMCDL register.

(i) D0 to D15 (Data) ··· 3-state I/O

These pins constitute a data bus for when the external memory is accessed. These are 16-bit data I/O bus pins.

The output changes in synchronization with the rise of the CLKOUT signal in the T1 state. In the idle state (TI), these pins become high impedance.

(16) CKSEL (Clock generator operating mode select) ··· input

This is an input pin used to specify the clock generator's operating mode.

(17) MODE0 to MODE2 (Mode) ··· input

These are input pins used to specify the operating mode.

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

$\overline{\text{RESET}}$ is a signal that is input asynchronously and 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 release a standby mode (HALT, IDLE, or software STOP).

(19) X1, X2 (Crystal)

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

(20) CV_{DD} (Power supply for clock generator)

This pin supplies positive power to the clock generator.

(21) CV_{SS} (Ground for clock generator)

This is the ground pin for the clock generator.

(22) V_{DD} (Power supply)

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

(23) V_{SS} (Ground)

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

(24) AV_{DD} (Analog power supply)

This is the analog positive power supply pin for the A/D converter.

(25) AV_{SS} (Analog ground)

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

(26) AV_{REF} (Analog reference voltage) ··· input

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

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

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

This pin is used for the μPD70F3107.

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

It is recommended that 1 to 10 kΩ resistors be used when connecting to V_{DD} or V_{SS} via resistors.

(1/2)

Pin Name	I/O Circuit Type	Recommended Connection
P00/PWM0	5	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
P01/INTP000/TI000	5-AC	
P02/INTP001		
P03/TO00	5	
P04/$\overline{\text{DMARQ0}}$/$\overline{\text{INTP100}}$ to P07/$\overline{\text{DMARQ3}}$/$\overline{\text{INTP103}}$	5-AC	
P10/PWM1	5	
P11/INTP010/TI010, P12/INTP011	5-AC	
P13/TO01	5	
P20/NMI	2	Connect to V_{SS} directly.
P21/INTP020/TI020, P22/INTP021	5-AC	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
P23/TO02	5	
P24/$\overline{\text{TC0}}$/$\overline{\text{INTP110}}$ to P27/$\overline{\text{TC3}}$/$\overline{\text{INTP113}}$	5-AC	
P30/SO2/$\overline{\text{INTP130}}$		
P31/SI2/$\overline{\text{INTP131}}$		
P32/$\overline{\text{SCK2}}$/$\overline{\text{INTP132}}$		
P33/TXD2/$\overline{\text{INTP133}}$		
P34/RXD2/$\overline{\text{INTP120}}$		
P35/$\overline{\text{INTP121}}$		
P36/$\overline{\text{INTP122}}$		
P37/ADTRG/$\overline{\text{INTP123}}$		
P40/TXD0/SO0	5	
P41/RXD0/SI0	5-AC	
P42/$\overline{\text{SCK0}}$		
P43/TXD1/SO1	5	
P44/RXD1/SI1	5-AC	
P45/$\overline{\text{SCK1}}$		
P50/INTP030/TI030, P51/INTP031		
P52/TO03	5	
P70/ANI0 to P77/ANI7	9	Connect to V_{SS} directly.
PBD0/$\overline{\text{DMAAK0}}$ to PBD3/$\overline{\text{DMAAK3}}$	5	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
PCM0/$\overline{\text{WAIT}}$	5	Input: Independently connect to V_{DD} via a resistor
PCM1/CLKOUT/BUSCLK	5	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open

(2/2)

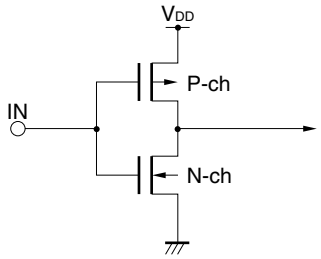
Pin Name	I/O Circuit Type	Recommended Connection
PCM2/$\overline{\text{HLDAK}}$	5	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
PCM3/$\overline{\text{HLDRQ}}$	5	Input: Independently connect to V_{DD} via a resistor
PCM4/$\overline{\text{REFRQ}}$	5	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
PCM5/SELFREF	5	Input: Independently connect to V_{SS} via a resistor
PCT0/$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM	5	Input: Independently connect to V_{DD} or V_{SS} via a resistor Output: Leave open
PCT1/$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM		
PCT4/$\overline{\text{RD}}$		
PCT5/$\overline{\text{WE}}$		
PCT6/$\overline{\text{OE}}$		
PCT7/$\overline{\text{BCYST}}$		
PCS0/$\overline{\text{CS0}}$		
PCS1/$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$		
PCS2/$\overline{\text{CS2}}$/$\overline{\text{IOWR}}$		
PCS3/$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$		
PCS4/$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$		
PCS5/$\overline{\text{CS5}}$/$\overline{\text{IORD}}$		
PCS6/$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$		
PCS7/$\overline{\text{CS7}}$		
PCD0/SDCKE		
PCD1/SDCLK		
PCD2/$\overline{\text{LBE}}$/$\overline{\text{SDCAS}}$		
PCD3/$\overline{\text{UBE}}$/$\overline{\text{SDRAS}}$		
PAH0/A16 to PAH9/A25		
PAL0/A0 to PAL15/A15		
PDL0/D0 to PDL15/D15		
MODE0, MODE1	2	–
MODE2[Note 1]		
MODE2/V_{PP}[Note 2]		
★ $\overline{\text{RESET}}$		–
★ CKSEL	1	–
AV_{SS}	–	Connect to V_{SS}.
AV_{DD}/AV_{REF}	–	Connect to V_{DD}.

Notes 1. μPD703103, 703105, 703106, 703107 only.

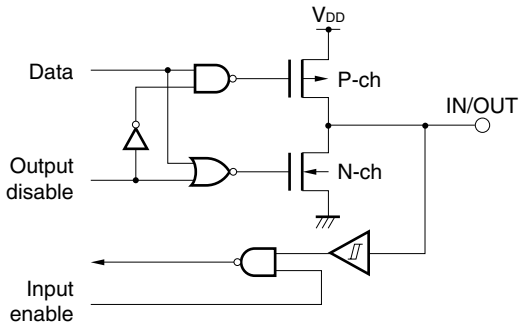
2. μPD70F3107 only

2.5 Pin I/O Circuits

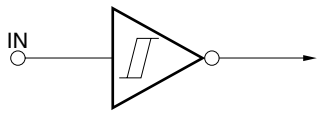
Type 1



Type 5-AC

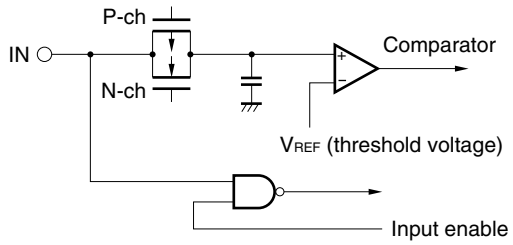


Type 2

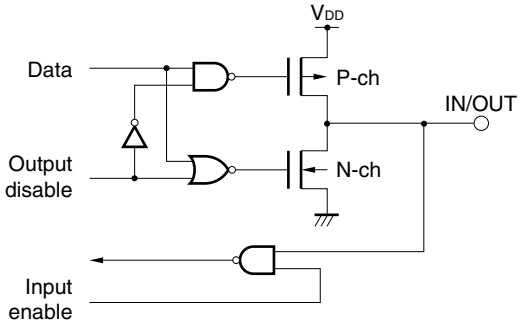


Schmitt-triggered input with hysteresis characteristics

Type 9



Type 5



CHAPTER 3 CPU FUNCTION

The CPU of the V850E/MA1 is based on RISC architecture and executes almost all the instructions in one clock cycle using 5-stage pipeline control.

3.1 Features

- Minimum instruction cycle: 20 ns (@ 50 MHz internal operation)
- Memory space Program space: 64 MB linear
  Data space: 4 GB linear
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiply/divide instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Long/short instruction format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The registers of the V850E/MA1 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32-bit width.

For details, refer to **V850E1 User's Manual Architecture**.

(1) Program register set

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

31	0
PC	(Program counter)

(2) System register set

31	0
EIPC	(Status saving register during interrupt)
EIPSW	(Status saving register during interrupt)

FEPC	(Status saving register during NMI)
FEPSW	(Status saving register during NMI)

ECR	(Interrrupt source register)

PSW	(Program status word)

CTPC	(Status saving register during CALLT execution)
CTPSW	(Status saving register during CALLT execution)

DBPC	(Status saving register during exception/debug trap)
DBPSW	(Status saving register during exception/debug trap)

CTBP	(CALLT base pointer)

3.2.1 Program register set

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

(1) General-purpose registers

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

However, r0 and r30 are implicitly used by instructions, and care must be exercised when using these registers. r0 is a register that always holds 0, and is used for operations using 0 and offset 0 addressing. r30 is used, by means of the SLD and SST instructions, as a base pointer for when memory is accessed. Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used. r2 may be used by the real-time OS. If the real-time OS does not use r2, it can be used as a variable register.

Table 3-1. Program Registers

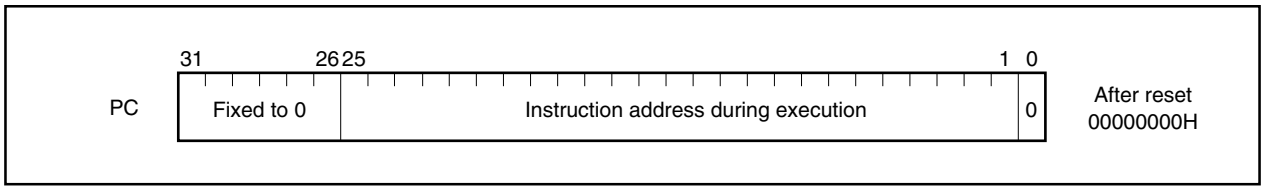
Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate data
r2	Address/data variable register (when r2 is not used by the real-time OS)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area (where program code is located)
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

(2) Program counter

This register holds the instruction address during program execution. The lower 26 bits of this register are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to 26, it is ignored.

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

Figure 3-1. Program Counter (PC)



3.2.2 System register set

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

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

Table 3-2. System Register Numbers

No.	System Register Name	Operand Specification	
		LDSR Instruction	STSR Instruction
0	Status saving register during interrupt (EIPC)[Note 1]	○	○
1	Status saving register during interrupt (EIPSW)	○	○
2	Status saving register during NMI (FEPC)	○	○
3	Status saving register during NMI (FEPSW)	○	○
4	Interrupt source register (ECR)	×	○
5	Program status word (PSW)	○	○
6 to 15	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×
16	Status saving register during CALLT execution (CTPC)	○	○
17	Status saving register during CALLT execution (CTPSW)	○	○
18	Status saving register during exception/debug trap (DBPC)	○[Note 2]	○
19	Status saving register during exception/debug trap (DBPSW)	○[Note 2]	○
20	CALLT base pointer (CTBP)	○	○
21 to 31	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×

Notes 1. Because this register has only one set, to approve multiple interrupts, it is necessary to save this register by program.

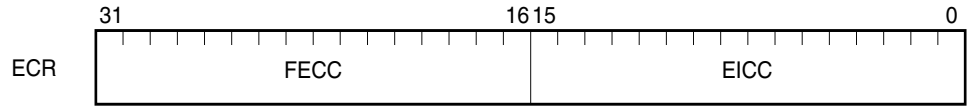
2. Access is only possible when the DBTRAP instruction is executed.

Caution Even if bit 0 of EIPC, FEPC, or CTPC is set to 1 with the LDSR instruction, bit 0 will be ignored when the program is returned by the RETI instruction after interrupt servicing (because bit 0 of the PC is fixed to 0). When setting the value of EIPC, FEPC, or CTPC, use an even value (bit 0 = 0).

Remark ○: Access allowed

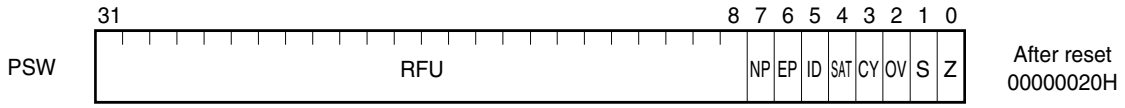
×: Access prohibited

Figure 3-2. Interrupt Source Register (ECR)



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

Figure 3-3. Program Status Word (PSW)



Bit position	Flag	Function
31 to 8	RFU	Reserved field (fixed to 0).
7	NP	Indicates that non-maskable interrupt (NMI) servicing is in progress. This flag is set when an NMI is acknowledged, and disables multiple interrupts. 0: NMI servicing not under execution. 1: NMI servicing under execution.
6	EP	Indicates that exception processing is in progress. This flag is set when an exception is generated. Moreover, interrupt requests can be acknowledged when this bit is set. 0: Exception processing not under execution. 1: Exception processing under execution.
5	ID	Indicates whether a maskable interrupt request can be acknowledged or not. 0: Interrupt enabled. 1: Interrupt disabled.
4	SAT[Note]	Indicates that the operation result of a saturated operation processing instruction is saturated due to overflow. Due to the cumulative flag, if the operation result is saturated by the saturation operation instruction, this bit is set (1), but is not cleared (0) even if the operation results of subsequent instructions are not saturated. To clear (0) this bit, load data in PSW. Note that in a general arithmetic operation, this bit is neither set (1) nor cleared (0). 0: Not saturated. 1: Saturated.
3	CY	This flag is set if a carry or borrow occurs as the result of an operation (if a carry or borrow does not occur, it is reset). 0: Carry or borrow does not occur. 1: Carry or borrow occurs.
2	OV[Note]	This flag is set if an overflow occurs during operation (if an overflow does not occur, it is reset). 0: Overflow does not occur. 1: Overflow occurs.
1	S[Note]	This flag is set if the result of an operation is negative (it is reset if the result is positive). 0: The operation result was positive or 0. 1: The operation result was negative.
0	Z	This flag is set if the result of an operation is zero (if the result is not zero, it is reset). 0: The operation result was not 0. 1: The operation result was 0.

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

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

3.3 Operating Modes

3.3.1 Operating modes

The V850E/MA1 has the following operating modes. Mode specification is carried out using the MODE0 to MODE2 pins.

(1) Normal operation mode

(a) Single-chip modes 0, 1

Access to the internal ROM is enabled.

In single-chip mode 0, after system reset is cleared, each pin related to the bus interface enters the port mode, program execution branches to the reset entry address of the internal ROM, and instruction processing starts. By setting the PMCAL, PMCAH, PMCDL, PMCCS, PMCCT, PMCCM, and PMCCD registers to control mode by instruction, an external device can be connected to the external memory area.

In single-chip mode 1, after system reset is cleared, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and instruction processing starts.

The internal ROM area is mapped from address 100000H.

(b) ROMless modes 0, 1

After system reset is cleared, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and instruction processing starts. Fetching of instructions and data access for internal ROM becomes impossible.

In ROMless mode 0, the data bus is a 16-bit data bus and in ROMless mode 1, the data bus is an 8-bit data bus.

(2) Flash memory programming mode (μPD70F3107 only)

If this mode is specified, it becomes possible for the flash programmer to run a program to the on-chip flash memory.

The initial value of the register differs depending on the mode.

Operating Mode		PMCAL	PMCAH	PMCDL	PMCCS	PMCCT	PMCCM	PMCCD	BSC
Normal operation mode	ROMless mode 0	FFFFH	03FFH	FFFFH	FFH	F3H	3FH	0FH	5555H
	ROMless mode 1	FFFFH	03FFH	FFFFH	FFH	F3H	3FH	0FH	0000H
	Single-chip mode 0	0000H	0000H	0000H	00H	00H	00H	00H	5555H
	Single-chip mode 1	FFFFH	03FFH	FFFFH	FFH	F3H	3FH	0FH	5555H

3.3.2 Operating mode specification

The operating mode is specified according to the status of the MODE0 to MODE2 pins. In an application system fix the specification of these pins and do not change them during operation. Operation is not guaranteed if these pins are changed during operation.

(a) μPD703103

MODE2	MODE1	MODE0	Operating Mode		Remarks
L	L	L	Normal operation mode	ROMless mode 0	16-bit data bus
L	L	H		ROMless mode 1	8-bit data bus
Other than above			Setting prohibited		

(b) μPD703105, 703106, 703107

MODE2	MODE1	MODE0	Operating Mode		Remarks
L	L	L	Normal operation mode	ROMless mode 0	16-bit data bus
L	L	H		ROMless mode 1	8-bit data bus
L	H	L		Single-chip mode 0	Internal ROM area is allocated from address 000000H.
L	H	H		Single-chip mode 1	Internal ROM area is allocated from address 100000H.
Other than above			Setting prohibited		

(c) μPD70F3107

MODE2/ V_{PP}	MODE1	MODE0	Operating Mode		Remarks
0 V	L	L	Normal operation mode	ROMless mode 0	16-bit data bus
0 V	L	H		ROMless mode 1	8-bit data bus
0 V	H	L		Single-chip mode 0	Internal ROM area is allocated from address 000000H.
0 V	H	H		Single-chip mode 1	Internal ROM area is allocated from address 100000H.
7.8 V	H	H/L	Flash memory programming mode		—
Other than above			Setting prohibited		

Remarks L: Low-level input
H: High-level input

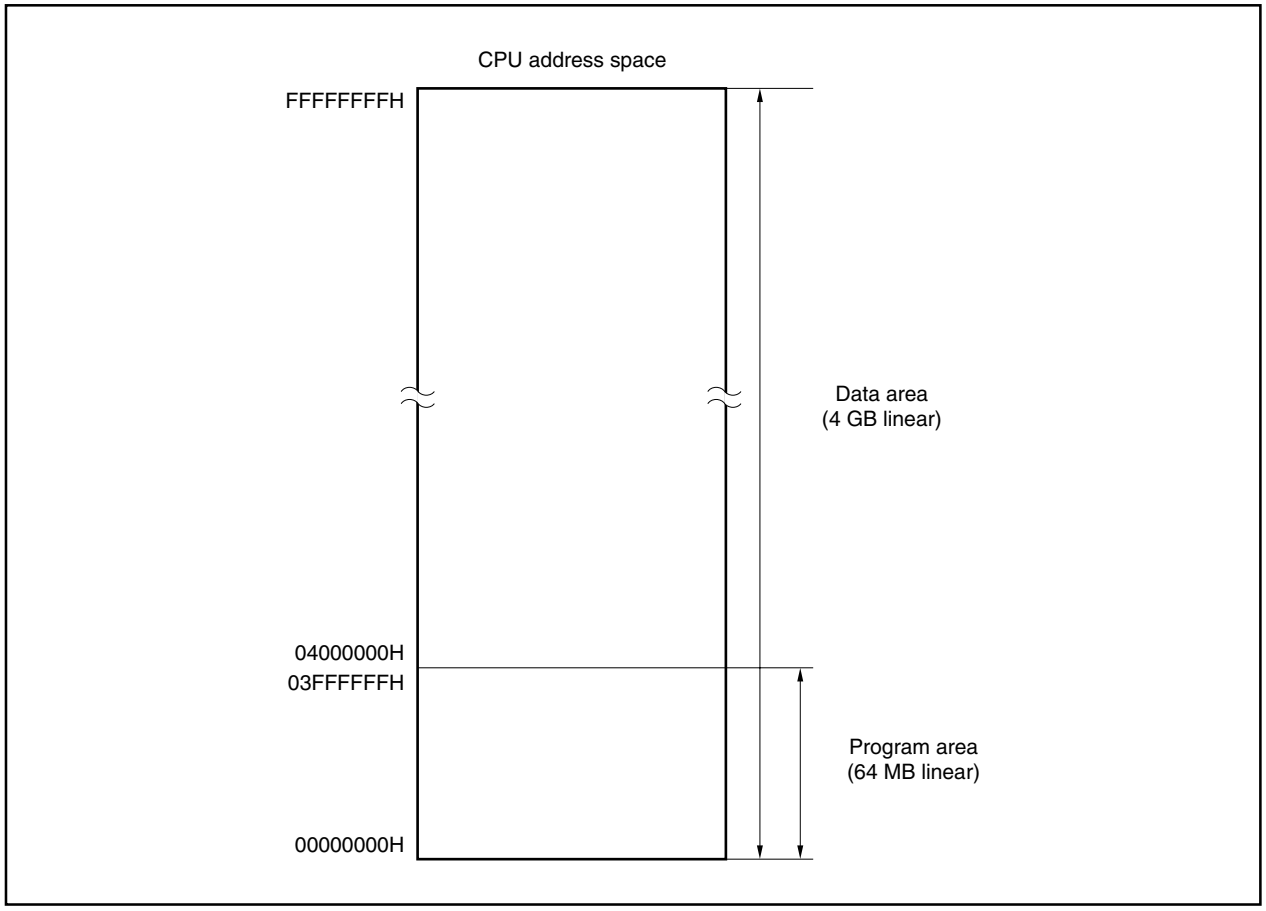
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/MA1 is of 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). Also, in instruction address addressing, a maximum of 64 MB of linear address space (program space) is supported.

Figure 3-4 shows the CPU address space.

Figure 3-4. CPU Address Space

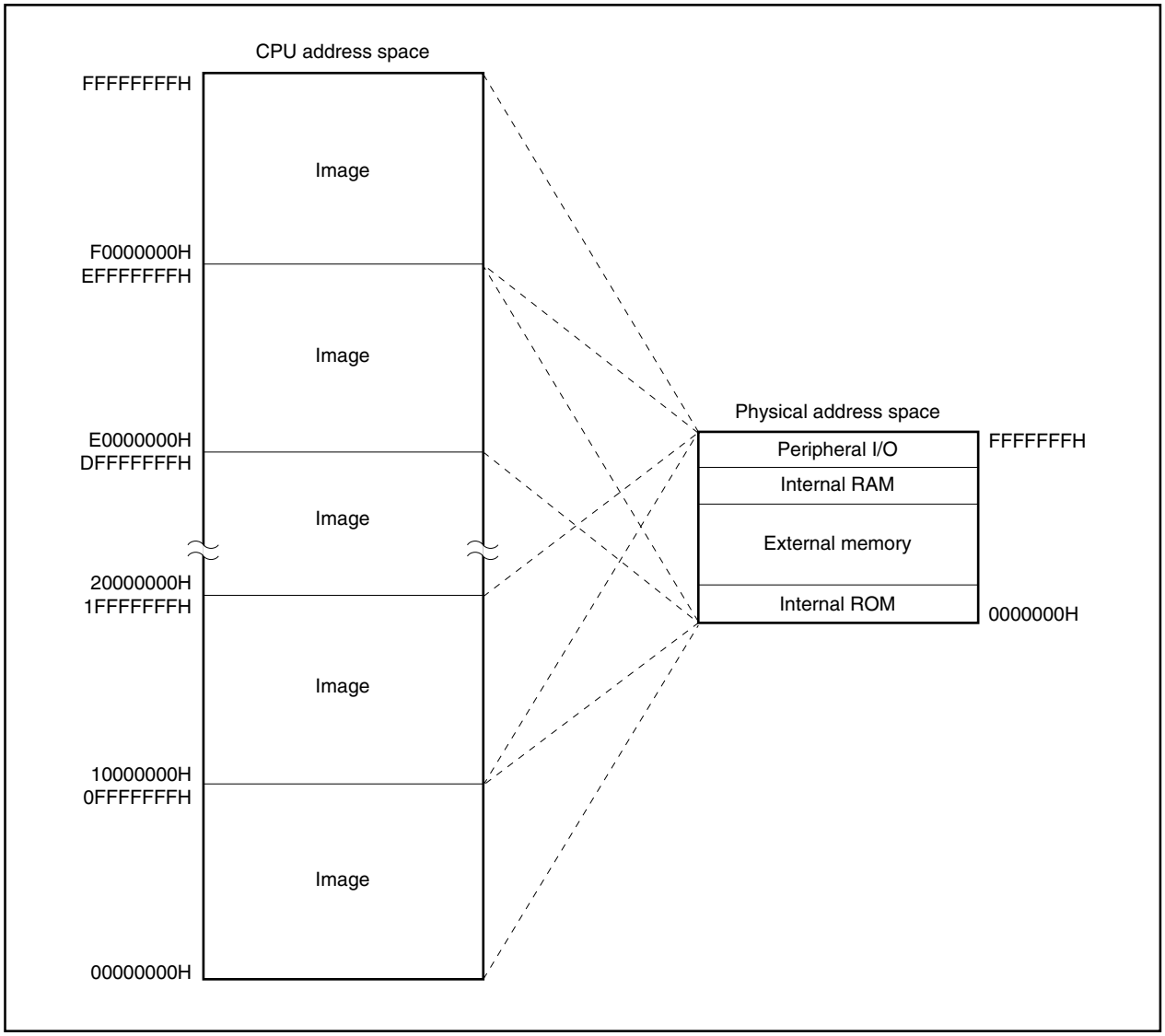


3.4.2 Image

A 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. In actuality, the same 256 MB physical address space is accessed regardless of the values of bits 31 to 28 of the CPU address. Figure 3-5 shows the image of the virtual addressing space.

Physical address x0000000H can be seen as CPU address 00000000H, and in addition, can be seen as address 10000000H, address 20000000H, … , address E0000000H, or address F0000000H.

Figure 3-5. Images on Address Space



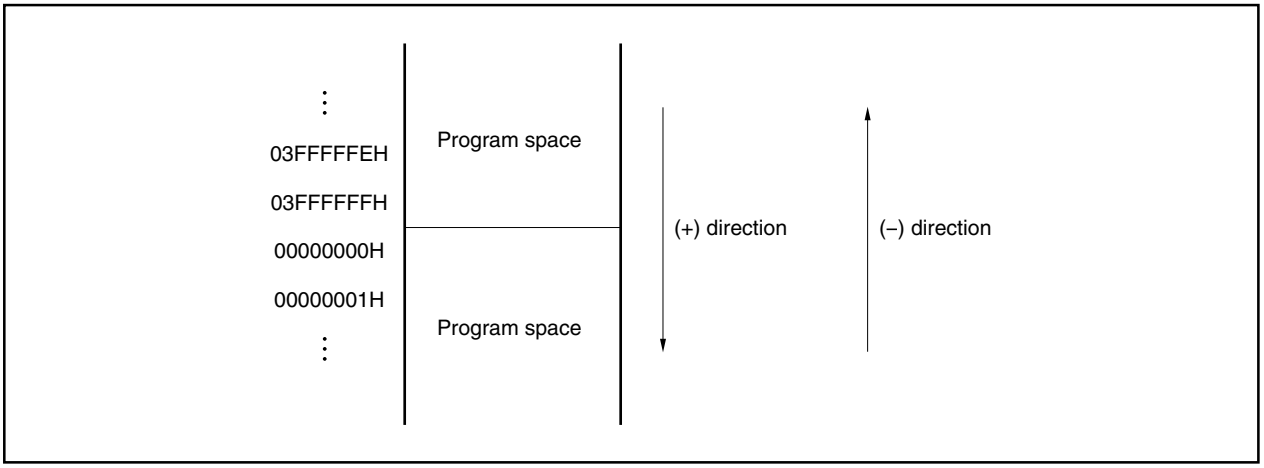
3.4.3 Wrap-around of CPU address space

(1) Program space

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

Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address 03FFFFFFH become contiguous addresses. Wrap-around refers to a situation like this whereby the lower-limit address and upper-limit address become contiguous.

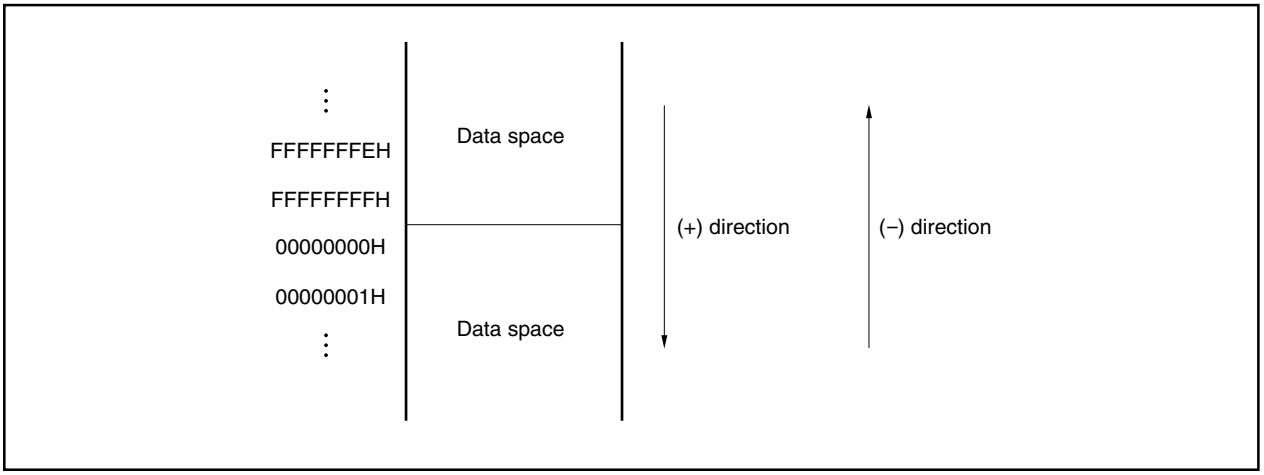
Caution The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. No instruction can be fetched from this area because this area is defined as peripheral I/O area. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

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

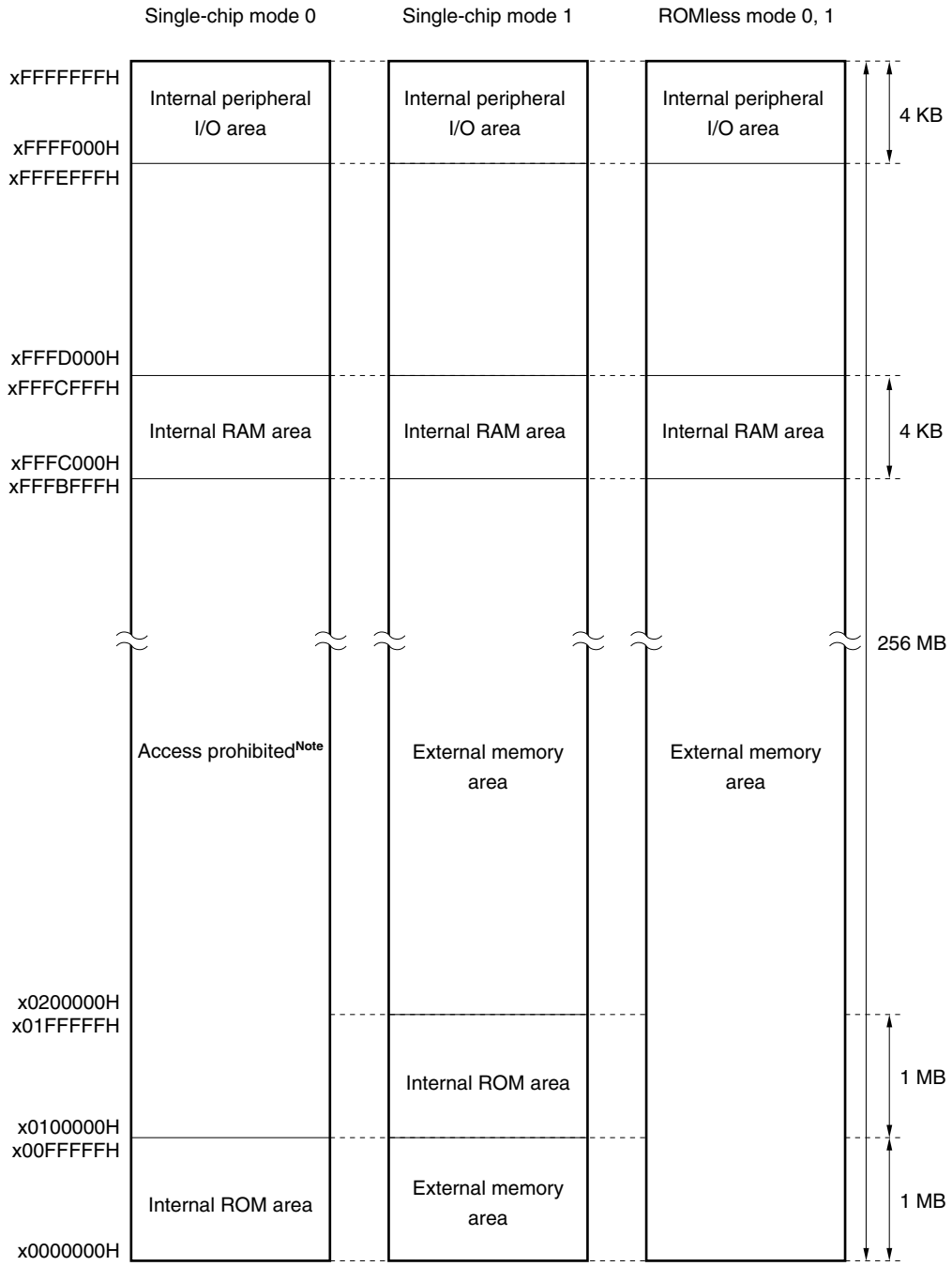
Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address FFFFFFFFH are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/MA1 reserves areas as shown in Figures 3-6 and 3-7. The mode is specified by the MODE0 to MODE2 pins.

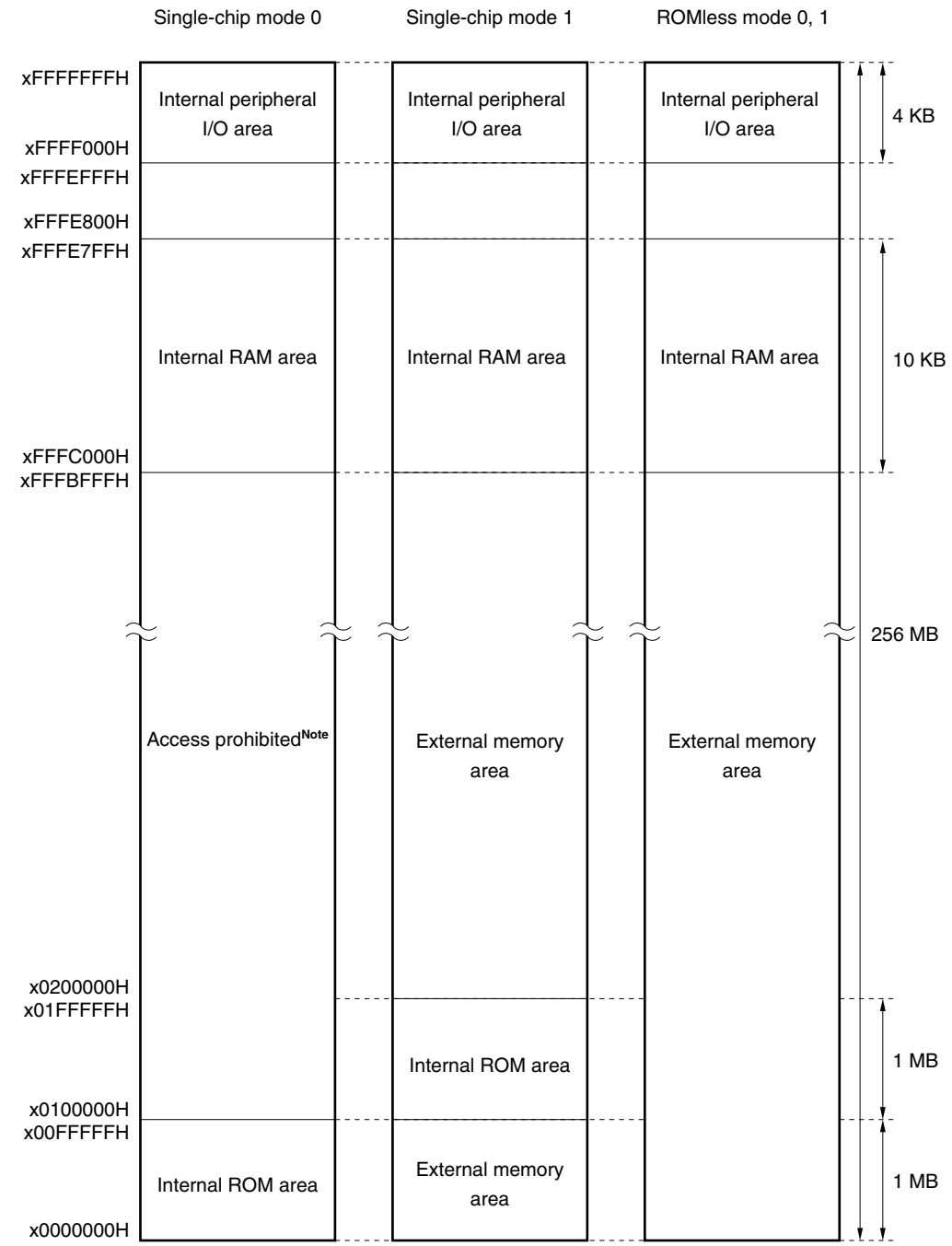
Figure 3-6. Memory Map (μPD703103, 703105)



Note By setting the PMCAL, PMCAH, PMCDL, PMCCS, PMCCT, PMCCM, and PMCCD registers to control mode, this area can be used as external memory area.

Remark For the μPD703103, only ROMless modes 0 and 1 are supported as the operating mode.

Figure 3-7. Memory Map (μPD703106, 703107, 70F3107)



Note By setting the PMCAL, PMCAH, PMCDL, PMCCS, PMCCT, PMCCM, and PMCCD registers to control mode, this area can be used as external memory area.

3.4.5 Area

(1) Internal ROM area

(a) Memory map (μPD703105, 703106, 703107, 70F3107)

1 MB of internal ROM area, addresses 00000H to FFFFFH, is reserved.

<1> μPD703105, 703106

128 KB are provided at the following addresses as physical internal ROM (mask ROM).

- In single-chip mode 0: Addresses 000000H to 01FFFFH
- In single-chip mode 1: Addresses 100000H to 11FFFFH

<2> μPD703107

256 KB are provided at the following addresses as physical internal ROM (mask ROM).

- In single-chip mode 0: Addresses 000000H to 03FFFFH
- In single-chip mode 1: Addresses 100000H to 13FFFFH

<3> μPD70F3107

256 KB are provided at the following addresses as physical internal ROM (flash memory).

- In single-chip mode 0: Addresses 000000H to 03FFFFH
- In single-chip mode 1: Addresses 100000H to 13FFFFH

(b) Interrupt/exception table

The V850E/MA1 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 acknowledged, execution jumps to the handler address, and the program written in that memory is executed. Table 3-3 shows the sources of interrupts/exceptions, and the corresponding addresses.

Remark When in ROMless modes 0 and 1, in single-chip mode 1, or in the case of the μPD703103, in order to restore correct operation after reset, provide a handler address to the reset routine at address 0 of the external memory.

Table 3-3. Interrupt/Exception Table (1/2)

Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000000H	RESET
00000010H	NMI
00000040H	TRAP0n (n = 0 to F)
00000050H	TRAP1n (n = 0 to F)
00000060H	ILGOP/DBG0
00000080H	INTOV00
00000090H	INTOV01
000000A0H	INTOV02
000000B0H	INTOV03
000000C0H	INTP000/INTM000
000000D0H	INTP001/INTM001
000000E0H	INTP010/INTM010
000000F0H	INTP011/INTM011
00000100H	INTP020/INTM020
00000110H	INTP021/INTM021
00000120H	INTP030/INTM030
00000130H	INTP031/INTM031
00000140H	INTP100
00000150H	INTP101
00000160H	INTP102
00000170H	INTP103
00000180H	INTP110
00000190H	INTP111
000001A0H	INTP112
000001B0H	INTP113
000001C0H	INTP120
000001D0H	INTP121
000001E0H	INTP122
000001F0H	INTP123
00000200H	INTP130
00000210H	INTP131
00000220H	INTP132
00000230H	INTP133
00000240H	INTCMD0
00000250H	INTCMD1
00000260H	INTCMD2
00000270H	INTCMD3

Table 3-3. Interrupt/Exception Table (2/2)

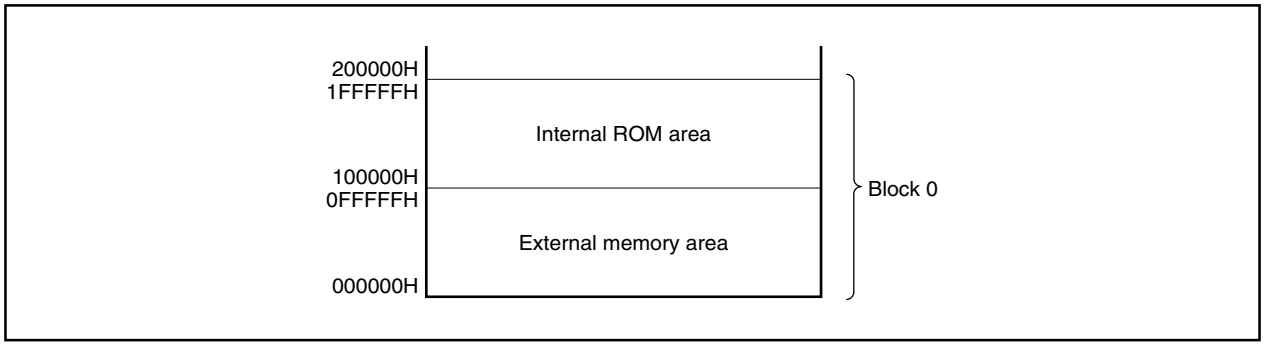
Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000280H	INTDMA0
00000290H	INTDMA1
000002A0H	INTDMA2
000002B0H	INTDMA3
000002C0H	INTCSI0
000002D0H	INTSER0
000002E0H	INTSR0
000002F0H	INTST0
00000300H	INTCSI1
00000310H	INTSER1
00000320H	INTSR1
00000330H	INTST1
00000340H	INTCSI2
00000350H	INTSER2
00000360H	INTSR2
00000370H	INTST2
00000380H	INTAD

(c) Internal ROM area relocation function

If set in single-chip mode 1, the internal ROM area is located beginning from address 100000H, so booting from external memory becomes possible.

Therefore, in order to resume correct operation after reset, provide a handler address to the reset routine at address 0 of the external memory.

Figure 3-8. Internal ROM Area in Single-Chip Mode 1



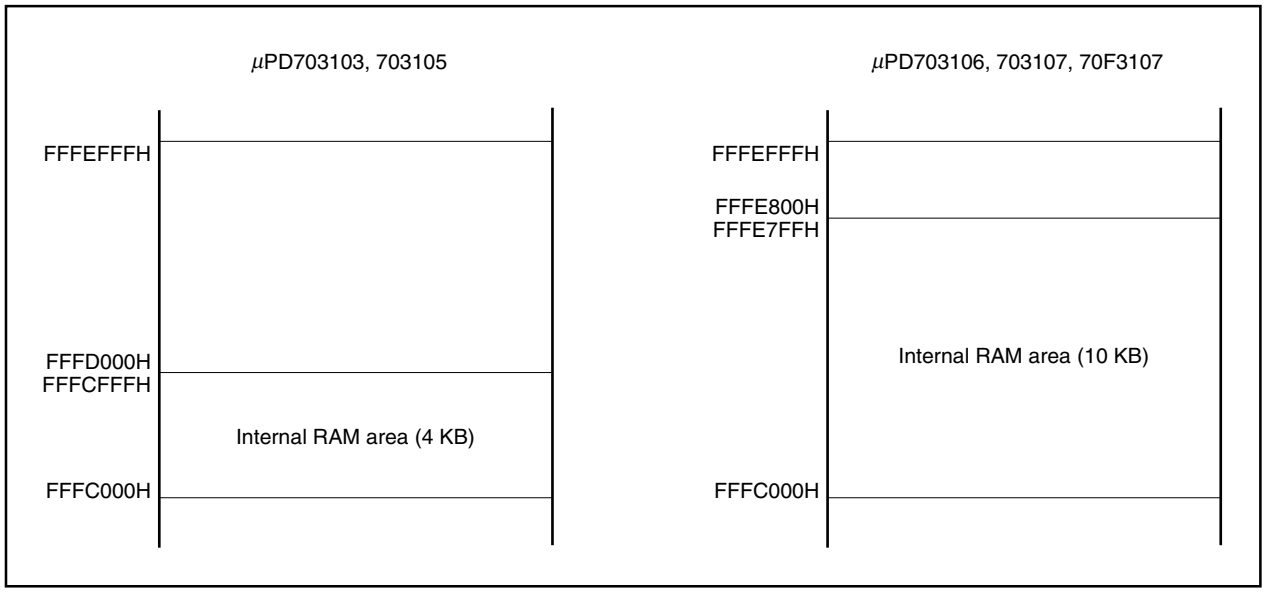
(2) Internal RAM area

The 12 KB of addresses FFFC000H to FFFEFFFH are reserved for the internal RAM area.

The 12 KB area of 3FFC000H to 3FFEFFFH can be seen as an image of FFFC000H to FFFEFFFH. When the internal RAM area is used as the program space, access addresses 3FFC000H to 3FFEFFFH.

In the μPD703103 and 703105, the 4 KB of addresses FFFC000H to FFFCFFFH are provided as physical internal RAM.

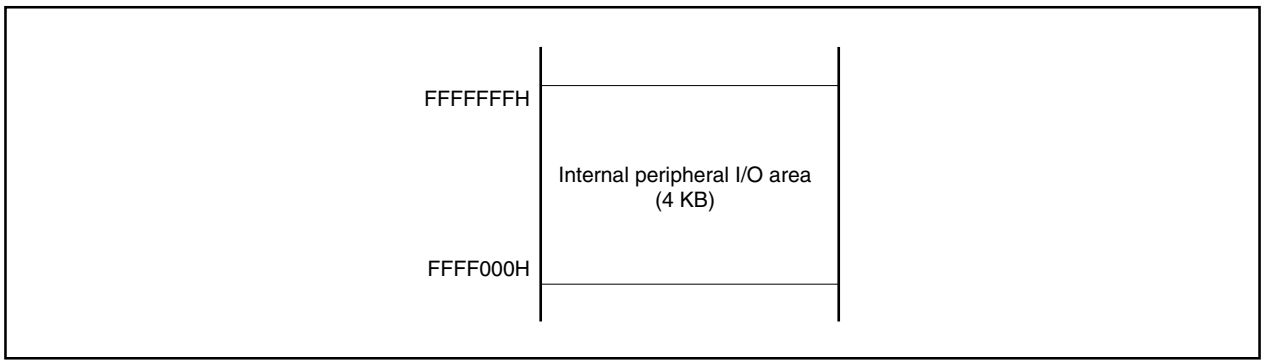
In the μPD703106, 703107, and 70F3107, the 10 KB of addresses FFFC000H to FFFE7FFH are provided as physical internal RAM.



(3) Internal peripheral I/O area

4 KB of memory, addresses FFFF000H to FFFFFFFH, are provided as an internal peripheral I/O area.
An image of addresses FFFF000H to FFFFFFFH can be seen at addresses 3FFF000H to 3FFFFFFH[Note].

★ **Note** Addresses 3FFF000H to 3FFFFFFH are access-prohibited. To access the internal peripheral I/O of this area, specify addresses FFFF000H to FFFFFFFH.



Peripheral I/O registers associated with the operating mode specification and the state monitoring for the internal peripheral I/O are all memory-mapped to the internal peripheral I/O area. Program fetches cannot be executed from this area.

Cautions 1. In the V850E/MA1, no registers exist which are capable of word access, but if a register is word accessed, halfword access is performed twice in the order of lower address, ★ **then higher address of the word area, disregarding the lower 2 bits of the address.**

2. For registers in which byte access is possible, if halfword access is executed, the higher 8 bits become undefined during the read operation, and the lower 8 bits of data are written to the register during the write operation.

3. 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. Addresses 3FFF000H to 3FFFFFFH cannot be specified as the source/destination address of DMA transfer. Be sure to use addresses FFFF000H to FFFFFFFH for the source/destination address of DMA transfer.

(4) External memory area

256 MB are available for external memory area. The lower 64 MB can be used as program/data area and the higher 192 MB as data area.

When in single-chip mode 0:	x0100000H to xFFFBFFFH
When in single-chip mode 1:	x0000000H to x00FFFFFH, x0200000H to xFFFBFFFH
When in ROMless modes 0 and 1:	x0000000H to xFFFBFFFH

Access to the external memory area uses the chip select signal assigned to each memory block (which is carried out in the CS unit set by chip area select control registers 0 and 1 (CSC0, CSC1)).
Note that the internal ROM, internal RAM, and internal peripheral I/O areas cannot be accessed as external memory areas.

3.4.6 External memory expansion

By setting the port n mode control register (PMCn) to control mode, an external memory device can be connected to the external memory space using each pin of ports AL, AH, DL, CS, CT, CM, and CD. Each register is set by selecting control mode for each pin of these ports using PMCn (n = AL, AH, DL, CS, CT, CM, CD).

Note that the status after reset differs as shown below in accordance with the operating mode specification set by pins MODE0 to MODE2 (refer to **3.3 Operating Modes** for details of the operating modes).

(a) In the case of ROMless mode 0

Because each pin of ports AL, AH, DL, CS, CT, CM, and CD enters control mode following a reset, external memory can be used without making changes to the port n mode control register (PMCn) (the external data bus width is 16 bits).

(b) In the case of ROMless mode 1

Because each pin of ports AL, AH, DL, CS, CT, CM, and CD enters control mode following a reset, external memory can be used without making changes to the port n mode control register (PMCn) (the external data bus width is 8 bits).

(c) In the case of single-chip mode 0

After reset, since the internal ROM area is accessed, each pin of ports AL, AH, DL, CS, CT, CM, and CD enters the port mode and external devices cannot be used.

To use external memory, set the port n mode control register (PMCn).

(d) In the case of single-chip mode 1

The internal ROM area is allocated from address 100000H. As a result, because each pin of ports AL, AH, DL, CS, CT, CM, and CD enters control mode following a reset, external memory can be used without making changes to the port n mode control register (PMCn) (the external data bus width is 16 bits).

Remark n = AL, AH, DL, CS, CT, CM, CD

3.4.7 Recommended use of address space

The architecture of the V850E/MA1 requires that a register that serves as a pointer be secured for address generation in operand data accessing of data space. Operand data access from instruction can be directly executed at the address in this pointer register ±32 KB. However, because the general-purpose registers that can be used as a pointer register are limited, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved.

(1) Program space

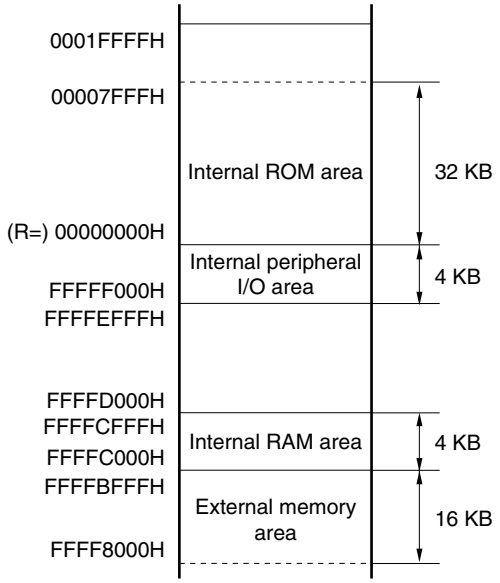
Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Therefore, a contiguous 64 MB space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

When the internal RAM area is used as program space, access addresses 3FFC000H to 3FFEFFFH.

(2) Data space

With the V850E/MA1, a 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. The highest bit (bit 25) of this 26-bit address is assigned as an address sign-extended to 32 bits.

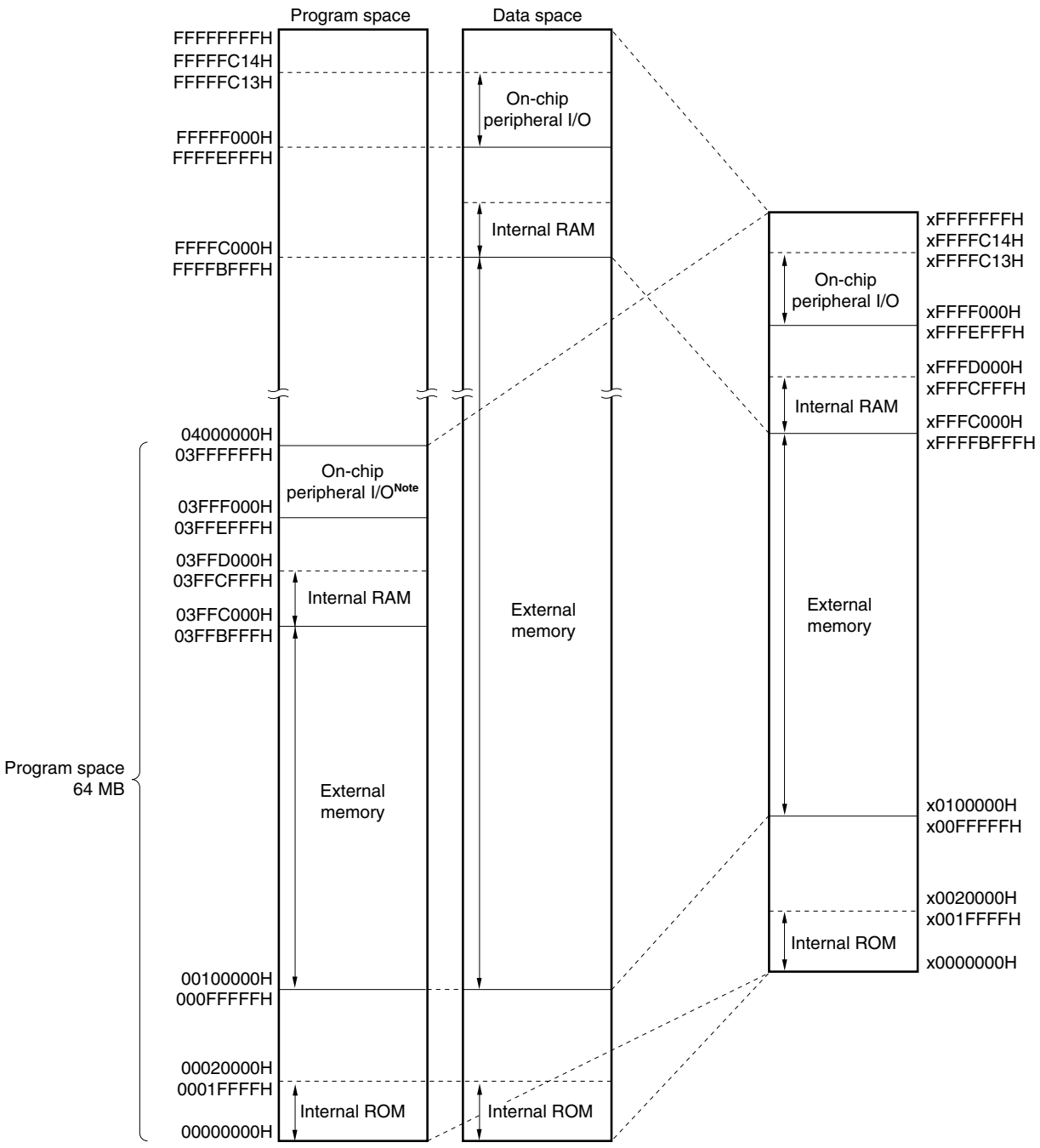
Example Application of wrap-around (μPD703105)



When R = r0 (zero register) is specified with the LD/ST disp16 [R] instruction, an addressing range of 00000000H ±32 KB can be referenced with the sign-extended disp 16. By mapping the external memory in the 16 KB area in the figure, all resources including internal 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 3-9. Recommended Memory Map



★ **Note** This area is access-prohibited. To access the on-chip peripheral I/O of this area, specify addresses FFFF000H to FFFFFFFH.

Remarks 1. The arrows indicate the recommended area.

2. This is a recommended memory map when the μPD703105 is set to single-chip mode 0, and used in external expansion mode.

3.4.8 Peripheral I/O registers

(1/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF000H	Port AL	PAL	R/W			○	Undefined
FFFFF000H	Port ALL	PALL	R/W	○	○		Undefined
FFFFF001H	Port ALH	PALH	R/W	○	○		Undefined
FFFFF002H	Port AH	PAH	R/W			○	Undefined
FFFFF002H	Port AHL	PAHL	R/W	○	○		Undefined
FFFFF003H	Port AHH	PAHH	R/W	○	○		Undefined
FFFFF004H	Port DL	PDL	R/W			○	Undefined
FFFFF004H	Port DLL	PDLL	R/W	○	○		Undefined
FFFFF005H	Port DLH	PDLH	R/W	○	○		Undefined
FFFFF008H	Port CS	PCS	R/W	○	○		Undefined
FFFFF00AH	Port CT	PCT	R/W	○	○		Undefined
FFFFF00CH	Port CM	PCM	R/W	○	○		Undefined
FFFFF00EH	Port CD	PCD	R/W	○	○		Undefined
FFFFF012H	Port BD	PBD	R/W	○	○		Undefined
FFFFF020H	Port AL mode register	PMAL	R/W			○	FFFFH
FFFFF020H	Port AL mode register L	PMALL	R/W	○	○		FFH
FFFFF021H	Port AL mode register H	PMALH	R/W	○	○		FFH
FFFFF022H	Port AH mode register	PMAH	R/W			○	FFFFH
FFFFF022H	Port AH mode register L	PMAHL	R/W	○	○		FFH
FFFFF023H	Port AH mode register H	PMAHH	R/W	○	○		FFH
FFFFF024H	Port DL mode register	PMDL	R/W			○	FFFFH
FFFFF024H	Port DL mode register L	PMDLL	R/W	○	○		FFH
FFFFF025H	Port DL mode register H	PMDLH	R/W	○	○		FFH
FFFFF028H	Port CS mode register	PMCS	R/W	○	○		FFH
FFFFF02AH	Port CT mode register	PMCT	R/W	○	○		FFH
FFFFF02CH	Port CM mode register	PMCM	R/W	○	○		FFH
FFFFF02EH	Port CD mode register	PMCD	R/W	○	○		FFH
FFFFF032H	Port BD mode register	PMBD	R/W	○	○		FFH
FFFFF040H	Port AL mode control register	PMCAL	R/W			○	0000H/FFFFH
FFFFF040H	Port AL mode control register L	PMCALL	R/W	○	○		00H/FFH
FFFFF041H	Port AL mode control register H	PMCALH	R/W	○	○		00H/FFH
FFFFF042H	Port AH mode control register	PMCAH	R/W			○	0000H/03FFH
FFFFF042H	Port AH mode control register L	PMCAHL	R/W	○	○		00H/FFH
FFFFF043H	Port AH mode control register H	PMCAHH	R/W	○	○		00H/03H

(2/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF044H	Port DL mode control register	PMCDL	R/W			○	0000H/FFFFH
FFFFF044H	Port DL mode control register L	PMCDLL	R/W	○	○		00H/FFH
FFFFF045H	Port DL mode control register H	PMCDLH	R/W	○	○		00H/FFH
FFFFF048H	Port CS mode control register	PMCCS	R/W	○	○		00H/FFH
FFFFF049H	Port CS function control register	PFCCS	R/W	○	○		00H
FFFFF04AH	Port CT mode control register	PMCCT	R/W	○	○		00H/F3H
FFFFF04CH	Port CM mode control register	PMCCM	R/W	○	○		00H/3FH
FFFFF04DH	Port CM function control register	PFCCM	R/W	○	○		00H
FFFFF04EH	Port CD mode control register	PMCCD	R/W	○	○		00H/0FH
FFFFF04FH	Port CD function control register	PFCCD	R/W	○	○		00H
FFFFF052H	Port BD mode control register	PMCBD	R/W	○	○		00H
FFFFF060H	Chip area select control register 0	CSC0	R/W			○	2C11H
FFFFF062H	Chip area select control register 1	CSC1	R/W			○	2C11H
FFFFF066H	Bus size configuration register	BSC	R/W			○	0000H/5555H
FFFFF068H	Endian configuration register	BEC	R/W			○	0000H
FFFFF06EH	System wait control register	VSWC	R/W		○		77H
FFFFF080H	DMA source address register 0L	DSA0L	R/W			○	Undefined
FFFFF082H	DMA source address register 0H	DSA0H	R/W			○	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L	R/W			○	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H	R/W			○	Undefined
FFFFF088H	DMA source address register 1L	DSA1L	R/W			○	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H	R/W			○	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L	R/W			○	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H	R/W			○	Undefined
FFFFF090H	DMA source address register 2L	DSA2L	R/W			○	Undefined
FFFFF092H	DMA source address register 2H	DSA2H	R/W			○	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L	R/W			○	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H	R/W			○	Undefined
FFFFF098H	DMA source address register 3L	DSA3L	R/W			○	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H	R/W			○	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L	R/W			○	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H	R/W			○	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0	R/W			○	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1	R/W			○	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2	R/W			○	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3	R/W			○	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0	R/W			○	0000H

(3/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF0D2H	DMA addressing control register 1	DADC1	R/W			○	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2	R/W			○	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3	R/W			○	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0	R/W	○	○		00H
FFFFF0E2H	DMA channel control register 1	DCHC1	R/W	○	○		00H
FFFFF0E4H	DMA channel control register 2	DCHC2	R/W	○	○		00H
FFFFF0E6H	DMA channel control register 3	DCHC3	R/W	○	○		00H
FFFFF0F0H	DMA disable status register	DDIS	R		○		00H
FFFFF0F2H	DMA restart register	DRST	R/W		○		00H
FFFFF100H	Interrupt mask register 0	IMR0	R/W			○	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L	R/W	○	○		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H	R/W	○	○		FFH
FFFFF102H	Interrupt mask register 1	IMR1	R/W			○	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L	R/W	○	○		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H	R/W	○	○		FFH
FFFFF104H	Interrupt mask register 2	IMR2	R/W			○	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L	R/W	○	○		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H	R/W	○	○		FFH
FFFFF106H	Interrupt mask register 3	IMR3	R/W			○	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L	R/W	○	○		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H	R/W	○	○		FFH
FFFFF110H	Interrupt control register	OVIC00	R/W	○	○		47H
FFFFF112H	Interrupt control register	OVIC01	R/W	○	○		47H
FFFFF114H	Interrupt control register	OVIC02	R/W	○	○		47H
FFFFF116H	Interrupt control register	OVIC03	R/W	○	○		47H
FFFFF118H	Interrupt control register	P00IC0	R/W	○	○		47H
FFFFF11AH	Interrupt control register	P00IC1	R/W	○	○		47H
FFFFF11CH	Interrupt control register	P01IC0	R/W	○	○		47H
FFFFF11EH	Interrupt control register	P01IC1	R/W	○	○		47H
FFFFF120H	Interrupt control register	P02IC0	R/W	○	○		47H
FFFFF122H	Interrupt control register	P02IC1	R/W	○	○		47H
FFFFF124H	Interrupt control register	P03IC0	R/W	○	○		47H
FFFFF126H	Interrupt control register	P03IC1	R/W	○	○		47H
FFFFF128H	Interrupt control register	P10IC0	R/W	○	○		47H
FFFFF12AH	Interrupt control register	P10IC1	R/W	○	○		47H
FFFFF12CH	Interrupt control register	P10IC2	R/W	○	○		47H
FFFFF12EH	Interrupt control register	P10IC3	R/W	○	○		47H

(4/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF130H	Interrupt control register	P11IC0	R/W	O	O		47H
FFFFF132H	Interrupt control register	P11IC1	R/W	O	O		47H
FFFFF134H	Interrupt control register	P11IC2	R/W	O	O		47H
FFFFF136H	Interrupt control register	P11IC3	R/W	O	O		47H
FFFFF138H	Interrupt control register	P12IC0	R/W	O	O		47H
FFFFF13AH	Interrupt control register	P12IC1	R/W	O	O		47H
FFFFF13CH	Interrupt control register	P12IC2	R/W	O	O		47H
FFFFF13EH	Interrupt control register	P12IC3	R/W	O	O		47H
FFFFF140H	Interrupt control register	P13IC0	R/W	O	O		47H
FFFFF142H	Interrupt control register	P13IC1	R/W	O	O		47H
FFFFF144H	Interrupt control register	P13IC2	R/W	O	O		47H
FFFFF146H	Interrupt control register	P13IC3	R/W	O	O		47H
FFFFF148H	Interrupt control register	CMICD0	R/W	O	O		47H
FFFFF14AH	Interrupt control register	CMICD1	R/W	O	O		47H
FFFFF14CH	Interrupt control register	CMICD2	R/W	O	O		47H
FFFFF14EH	Interrupt control register	CMICD3	R/W	O	O		47H
FFFFF150H	Interrupt control register	DMAIC0	R/W	O	O		47H
FFFFF152H	Interrupt control register	DMAIC1	R/W	O	O		47H
FFFFF154H	Interrupt control register	DMAIC2	R/W	O	O		47H
FFFFF156H	Interrupt control register	DMAIC3	R/W	O	O		47H
FFFFF158H	Interrupt control register	CSIIC0	R/W	O	O		47H
FFFFF15AH	Interrupt control register	SEIC0	R/W	O	O		47H
FFFFF15CH	Interrupt control register	SRIC0	R/W	O	O		47H
FFFFF15EH	Interrupt control register	STIC0	R/W	O	O		47H
FFFFF160H	Interrupt control register	CSIIC1	R/W	O	O		47H
FFFFF162H	Interrupt control register	SEIC1	R/W	O	O		47H
FFFFF164H	Interrupt control register	SRIC1	R/W	O	O		47H
FFFFF166H	Interrupt control register	STIC1	R/W	O	O		47H
FFFFF168H	Interrupt control register	CSIIC2	R/W	O	O		47H
FFFFF16AH	Interrupt control register	SEIC2	R/W	O	O		47H
FFFFF16CH	Interrupt control register	SRIC2	R/W	O	O		47H
FFFFF16EH	Interrupt control register	STIC2	R/W	O	O		47H
FFFFF170H	Interrupt control register	ADIC	R/W	O	O		47H
FFFFF1FAH	In-service priority register	ISPR	R	O	O		00H
FFFFF1FCH	Command register	PRCMD	W		O		Undefined
FFFFF1FEH	Power-save control register	PSC	R/W	O	O		00H
FFFFF200H	A/D converter mode register 0	ADM0	R/W	O	O		00H

(5/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF201H	A/D converter mode register 1	ADM1	R/W		○		07H
FFFFF202H	A/D converter mode register 2	ADM2	R/W	○	○		00H
FFFFF210H	A/D conversion result register 0 (10 bits)	ADCR0	R			○	0000H
FFFFF212H	A/D conversion result register 1 (10 bits)	ADCR1	R			○	0000H
FFFFF214H	A/D conversion result register 2 (10 bits)	ADCR2	R			○	0000H
FFFFF216H	A/D conversion result register 3 (10 bits)	ADCR3	R			○	0000H
FFFFF218H	A/D conversion result register 4 (10 bits)	ADCR4	R			○	0000H
FFFFF21AH	A/D conversion result register 5 (10 bits)	ADCR5	R			○	0000H
FFFFF21CH	A/D conversion result register 6 (10 bits)	ADCR6	R			○	0000H
FFFFF21EH	A/D conversion result register 7 (10 bits)	ADCR7	R			○	0000H
FFFFF220H	A/D conversion result register 0H (8 bits)	ADCR0H	R		○		00H
FFFFF221H	A/D conversion result register 1H (8 bits)	ADCR1H	R		○		00H
FFFFF222H	A/D conversion result register 2H (8 bits)	ADCR2H	R		○		00H
FFFFF223H	A/D conversion result register 3H (8 bits)	ADCR3H	R		○		00H
FFFFF224H	A/D conversion result register 4H (8 bits)	ADCR4H	R		○		00H
FFFFF225H	A/D conversion result register 5H (8 bits)	ADCR5H	R		○		00H
FFFFF226H	A/D conversion result register 6H (8 bits)	ADCR6H	R		○		00H
FFFFF227H	A/D conversion result register 7H (8 bits)	ADCR7H	R		○		00H
FFFFF400H	Port 0	P0	R/W	○	○		Undefined
FFFFF402H	Port 1	P1	R/W	○	○		Undefined
FFFFF404H	Port 2	P2	R/W	○	○		Undefined
FFFFF406H	Port 3	P3	R/W	○	○		Undefined
FFFFF408H	Port 4	P4	R/W	○	○		Undefined
FFFFF40AH	Port 5	P5	R/W	○	○		Undefined
FFFFF40EH	Port 7	P7	R/W	○	○		Undefined
FFFFF420H	Port 0 mode register	PM0	R/W	○	○		FFH
FFFFF422H	Port 1 mode register	PM1	R/W	○	○		FFH
FFFFF424H	Port 2 mode register	PM2	R/W	○	○		FFH
FFFFF426H	Port 3 mode register	PM3	R/W	○	○		FFH
FFFFF428H	Port 4 mode register	PM4	R/W	○	○		FFH
FFFFF42AH	Port 5 mode register	PM5	R/W	○	○		FFH
FFFFF440H	Port 0 mode control register	PMC0	R/W	○	○		00H
FFFFF442H	Port 1 mode control register	PMC1	R/W	○	○		00H
FFFFF444H	Port 2 mode control register	PMC2	R/W	○	○		01H
FFFFF446H	Port 3 mode control register	PMC3	R/W	○	○		00H
FFFFF448H	Port 4 mode control register	PMC4	R/W	○	○		00H
FFFFF44AH	Port 5 mode control register	PMC5	R/W	○	○		00H

(6/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF460H	Port 0 function control register	PFC0	R/W	O	O		00H
FFFFF464H	Port 2 function control register	PFC2	R/W	O	O		00H
FFFFF466H	Port 3 function control register	PFC3	R/W	O	O		00H
FFFFF468H	Port 4 function control register	PFC4	R/W	O	O		00H
FFFFF480H	Bus cycle type configuration register 0	BCT0	R/W			O	8888H
FFFFF482H	Bus cycle type configuration register 1	BCT1	R/W			O	8888H
FFFFF484H	Data wait control register 0	DWC0	R/W			O	7777H
FFFFF486H	Data wait control register 1	DWC1	R/W			O	7777H
FFFFF488H	Bus cycle control register	BCC	R/W			O	FFFFH
FFFFF48AH	Address setup wait control register	ASC	R/W			O	FFFFH
FFFFF48CH	Bus cycle period control register	BCP	R/W		O		00H
FFFFF49AH	Page-ROM configuration register	PRC	R/W			O	7000H
FFFFF49EH	Refresh wait control register	RWC	R/W		O		00H
FFFFF4A4H	DRAM configuration register 1	SCR1	R/W			O	3FC1H
	SDRAM configuration register 1		R/W			O	0000H
FFFFF4A6H	Refresh control register 1	RFS1	R/W			O	0000H
	SDRAM refresh control register 1		R/W			O	0000H
FFFFF4ACH	DRAM configuration register 3	SCR3	R/W			O	3FC1H
	SDRAM configuration register 3		R/W			O	0000H
FFFFF4AEH	Refresh control register 3	RFS3	R/W			O	0000H
	SDRAM refresh control register 3		R/W			O	0000H
FFFFF4B0H	DRAM configuration register 4	SCR4	R/W			O	3FC1H
	SDRAM configuration register 4		R/W			O	0000H
FFFFF4B2H	Refresh control register 4	RFS4	R/W			O	0000H
	SDRAM refresh control register 4		R/W			O	0000H
FFFFF4B8H	DRAM configuration register 6	SCR6	R/W			O	3FC1H
	SDRAM configuration register 6		R/W			O	0000H
FFFFF4BAH	Refresh control register 6	RFS6	R/W			O	0000H
	SDRAM refresh control register 6		R/W			O	0000H
FFFFF540H	Timer D0	TMD0	R			O	0000H
FFFFF542H	Compare register D0	CMD0	R/W			O	0000H
FFFFF544H	Timer mode control register D0	TMCD0	R/W	O	O		00H
FFFFF550H	Timer D1	TMD1	R			O	0000H
FFFFF552H	Compare register D1	CMD1	R/W			O	0000H
FFFFF554H	Timer mode control register D1	TMCD1	R/W	O	O		00H
FFFFF560H	Timer D2	TMD2	R			O	0000H
FFFFF562H	Compare register D2	CMD2	R/W			O	0000H

(7/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF564H	Timer mode control register D2	TMCD2	R/W	○	○		00H
FFFFF570H	Timer D3	TMD3	R			○	0000H
FFFFF572H	Compare register D3	CMD3	R/W			○	0000H
FFFFF574H	Timer mode control register D3	TMCD3	R/W	○	○		00H
FFFFF600H	Timer C0	TMC0	R			○	0000H
FFFFF602H	Capture/compare register C00	CCC00	R/W			○	0000H
FFFFF604H	Capture/compare register C01	CCC01	R/W			○	0000H
FFFFF606H	Timer mode control register C00	TMCC00	R/W	○	○		00H
FFFFF608H	Timer mode control register C01	TMCC01	R/W		○		20H
FFFFF609H	Valid edge select register C0	SESC0	R/W		○		00H
FFFFF610H	Timer C1	TMC1	R			○	0000H
FFFFF612H	Capture/compare register C10	CCC10	R/W			○	0000H
FFFFF614H	Capture/compare register C11	CCC11	R/W			○	0000H
FFFFF616H	Timer mode control register C10	TMCC10	R/W	○	○		00H
FFFFF618H	Timer mode control register C11	TMCC11	R/W		○		20H
FFFFF619H	Valid edge select register C1	SESC1	R/W		○		00H
FFFFF620H	Timer C2	TMC2	R			○	0000H
FFFFF622H	Capture/compare register C20	CCC20	R/W			○	0000H
FFFFF624H	Capture/compare register C21	CCC21	R/W			○	0000H
FFFFF626H	Timer mode control register C20	TMCC20	R/W	○	○		00H
FFFFF628H	Timer mode control register C21	TMCC21	R/W		○		20H
FFFFF629H	Valid edge select register C2	SESC2	R/W		○		00H
FFFFF630H	Timer C3	TMC3	R			○	0000H
FFFFF632H	Capture/compare register C30	CCC30	R/W			○	0000H
FFFFF634H	Capture/compare register C31	CCC31	R/W			○	0000H
FFFFF636H	Timer mode control register C30	TMCC30	R/W	○	○		00H
FFFFF638H	Timer mode control register C31	TMCC31	R/W		○		20H
FFFFF639H	Valid edge select register C3	SESC3	R/W		○		00H
FFFFF800H	Peripheral command register	PHCMD	W		○		Undefined
FFFFF802H	Peripheral status register	PHS	R/W	○	○		00H
FFFFF810H	DMA trigger factor register 0	DTFR0	R/W	○	○		00H
FFFFF812H	DMA trigger factor register 1	DTFR1	R/W	○	○		00H
FFFFF814H	DMA trigger factor register 2	DTFR2	R/W	○	○		00H
FFFFF816H	DMA trigger factor register 3	DTFR3	R/W	○	○		00H
FFFFF820H	Power-save mode register	PSMR	R/W	○	○		00H
FFFFF822H	Clock control register	CKC	R/W		○		00H
FFFFF824H	Lock register	LOCKR	R	○	○		0xH

(8/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFF880H	External interrupt mode register 0	INTM0	R/W	○	○		00H
FFFFF882H	External interrupt mode register 1	INTM1	R/W		○		00H
FFFFF884H	External interrupt mode register 2	INTM2	R/W		○		00H
FFFFF886H	External interrupt mode register 3	INTM3	R/W		○		00H
FFFFF888H	External interrupt mode register 4	INTM4	R/W		○		00H
FFFFF8A0H	DMA terminal count output control register	DTOC	R/W		○		01H
FFFFF8D4H	Flash programming mode control register	FLPMC	R/W	○	○		08H/0CH/00H
FFFFF900H	Clocked serial interface mode register 0	CSIM0	R/W	○	○		00H
FFFFF901H	Clocked serial interface clock select register 0	CSIC0	R/W		○		00H
FFFFF902H	Serial I/O shift register 0	SIO0	R		○		00H
FFFFF903H	Receive-only serial I/O shift register 0	SIOE0	R		○		00H
FFFFF904H	Clocked serial interface transmit buffer register 0	SOTB0	R/W		○		00H
FFFFF910H	Clocked serial interface mode register 1	CSIM1	R/W	○	○		00H
FFFFF911H	Clocked serial interface clock select register 1	CSIC1	R/W		○		00H
FFFFF912H	Serial I/O shift register 1	SIO1	R		○		00H
FFFFF913H	Receive-only serial I/O shift register 1	SIOE1	R		○		00H
FFFFF914H	Clocked serial interface transmit buffer register 1	SOTB1	R/W		○		00H
FFFFF920H	Clocked serial interface mode register 2	CSIM2	R/W	○	○		00H
FFFFF921H	Clocked serial interface clock select register 2	CSIC2	R/W		○		00H
FFFFF922H	Serial I/O shift register 2	SIO2	R		○		00H
FFFFF923H	Receive-only serial I/O shift register 2	SIOE2	R		○		00H
FFFFF924H	Clocked serial interface transmit buffer register 2	SOTB2	R/W		○		00H
FFFFFA00H	Asynchronous serial interface mode register 0	ASIM0	R/W	○	○		01H
FFFFFA02H	Receive buffer register 0	RXB0	R		○		FFH
FFFFFA03H	Asynchronous serial interface status register 0	ASIS0	R		○		00H
FFFFFA04H	Transmit buffer register 0	TXB0	R/W		○		FFH
FFFFFA05H	Asynchronous serial interface transmit status register 0	ASIF0	R	○	○		00H
FFFFFA06H	Clock select register 0	CKSR0	R/W		○		00H
FFFFFA07H	Baud rate generator control register 0	BRGC0	R/W		○		FFH
FFFFFA10H	Aynchronous serial interface mode register 1	ASIM1	R/W	○	○		01H
FFFFFA12H	Receive buffer register 1	RXB1	R		○		FFH
FFFFFA13H	Asynchronous serial interface status register 1	ASIS1	R		○		00H
FFFFFA14H	Transmit buffer register 1	TXB1	R/W		○		FFH
FFFFFA15H	Aynchronous serial interface transmit status register 1	ASIF1	R	○	○		00H
FFFFFA16H	Clock select register 1	CKSR1	R/W		○		00H
FFFFFA17H	Baud rate generator control register 1	BRGC1	R/W		○		FFH

(9/9)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FFFFFA20H	Aynchronous serial interface mode register 2	ASIM2	R/W	○	○		01H
FFFFFA22H	Receive buffer register 2	RXB2	R		○		FFH
FFFFFA23H	Asynchronous serial interface status register 2	ASIS2	R		○		00H
FFFFFA24H	Transmit buffer register 2	TXB2	R/W		○		FFH
FFFFFA25H	Asynchronous serial interface transmit status register 2	ASIF2	R	○	○		00H
FFFFFA26H	Clock select register 2	CKSR2	R/W		○		00H
FFFFFA27H	Baud rate generator control register 2	BRGC2	R/W		○		FFH
FFFFFC00H	PWM control register 0	PWMC0	R/W	○	○		40H
FFFFFC02H	PWM buffer register 0	PWMB0	R/W			○	0000H
FFFFFC10H	PWM control register 1	PWMC1	R/W	○	○		40H
FFFFFC12H	PWM buffer register 1	PWMB1	R/W			○	0000H

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 V850E/MA1 has three specific registers, the power-save control register (PSC) (refer to **9.5.2 (3) Power-save control register (PSC)**), clock control register (CKC) (refer to **9.3.4 Clock control register (CKC)**), and flash programming mode control register (FLPMC). Disable DMA transfer when writing to a specific register.

There are also two protection registers supporting write operations for specific registers to avoid an unexpected stoppage of the application system due to erroneous program execution. These two registers are the command register (PRCMD) and peripheral command register (PHCMD) (refer to **9.5.2 (2) Command register (PRCMD)** and **9.3.3 Peripheral command register (PHCMD)**).

★ **3.4.10 System wait control register (VSWC)**

The system wait control register (VSWC) is a register that controls the bus access wait for the on-chip peripheral I/O registers.

Access to on-chip peripheral I/O registers is made in 3 clocks (without wait), however, in the V850E/MA1 waits may be required depending on the operation frequency. Set the values described in the table below to the VSWC in accordance with the operation frequency used.

This register can be read/written in 8-bit units (address: FFFFF06EH, initial value: 77H).

Operation Frequency (ϕ)	Set Value of VSWC	Number of Waits for On-chip Peripheral I/O Register Access
4 MHz ≤ ϕ < 25 MHz	11H	2
25 MHz ≤ ϕ < 50 MHz	12H	3
ϕ = 50 MHz	13H	4

3.4.11 Cautions

When using the V850E/MA1, the following registers must be set in the beginning.

- System wait control register (VSWC)
  (See **3.4.10 System wait control register (VSWC)**)
- Clock control register (CKC)
  (See **9.3.4 Clock control register (CKC)**)

After setting VSWC and CKC, set other registers if necessary.

To use the external bus, initialize each register in the following sequence after setting the above registers.

<1> Set each pin to the control mode by setting each port-related register.
<2> Select a chip select space by using chip area select control register n (CSCn) (n = 0 or 1).
<3> Specify the type of memory of each chip select space by using bus cycle type configuration register n (BCTn).

CHAPTER 4 BUS CONTROL FUNCTION

The V850E/MA1 is provided with an external bus interface function by which external I/O and memories, such as ROM and RAM, can be connected.

4.1 Features

- 16-bit/8-bit data bus sizing function
- 8-space chip select function
- Wait function
  - Programmable wait function, through which up to 7 wait states can be inserted for each memory block
  - External wait function via $\overline{\text{WAIT}}$ pin
- Idle state insertion function
- Bus mastership arbitration function
- Bus hold function
- External device connection enabled via bus control/port alternate function pins

4.2 Bus Control Pins

The following pins are used for connection to external devices.

Bus Control Pin (Function When in Control Mode)	Function When in Port Mode	Register for Port/Control Mode Switching
Data bus (D0 to D15)	PDL0 to PDL15 (Port DL)	PMCDL
Address bus (A0 to A15)	PAL0 to PAL15 (Port AL)	PMCAL
Address bus (A16 to A25)	PAH0 to PAH9 (Port AH)	PMCAH
Chip select ($\overline{\text{CS0}}$ to $\overline{\text{SC7}}$, $\overline{\text{RAS1}}$, $\overline{\text{RAS3}}$, $\overline{\text{RAS4}}$, $\overline{\text{RAS6}}$, $\overline{\text{IOWR}}$, $\overline{\text{IORD}}$)	PCS0 to PCS7 (Port CS)	PMCCS
SDRAM sync control (SDCKE, SDCLK)	PCD0, PCD1 (Port CD)	PMCCD
Byte access control/SDRAM control ($\overline{\text{LBE}}$/$\overline{\text{SDCAS}}$, $\overline{\text{UBE}}$/$\overline{\text{SDRAS}}$)	PCD2, PCD3 (Port CD)	
Read/write control ($\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM, $\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM, $\overline{\text{RD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$)	PCT0, PCT1, PCT4 to PCT6 (Port CT)	PMCCT
Bus cycle start ($\overline{\text{BCYST}}$)	PCT7 (Port CT)	
External wait control ($\overline{\text{WAIT}}$)	PCM0 (Port CM)	PMCCM
Internal system clock (CLKOUT)	PCM1 (Port CM)	
Bus hold control ($\overline{\text{HLDRQ}}$, $\overline{\text{HLDAK}}$)	PCM2, PCM3 (Port CM)	
DRAM refresh control ($\overline{\text{REFRQ}}$)	PCM4 (Port CM)	
Self-refresh control (SELFREF)	PCM5 (Port CM)	

Remark In the case of single-chip mode 1 and ROMless modes 0 and 1, when the system is reset, each bus control pin becomes unconditionally valid. (However, D8 to D15 are valid only in single-chip mode 1 and ROMless mode 0.)

4.2.1 Pin status during internal ROM, internal RAM, and peripheral I/O access

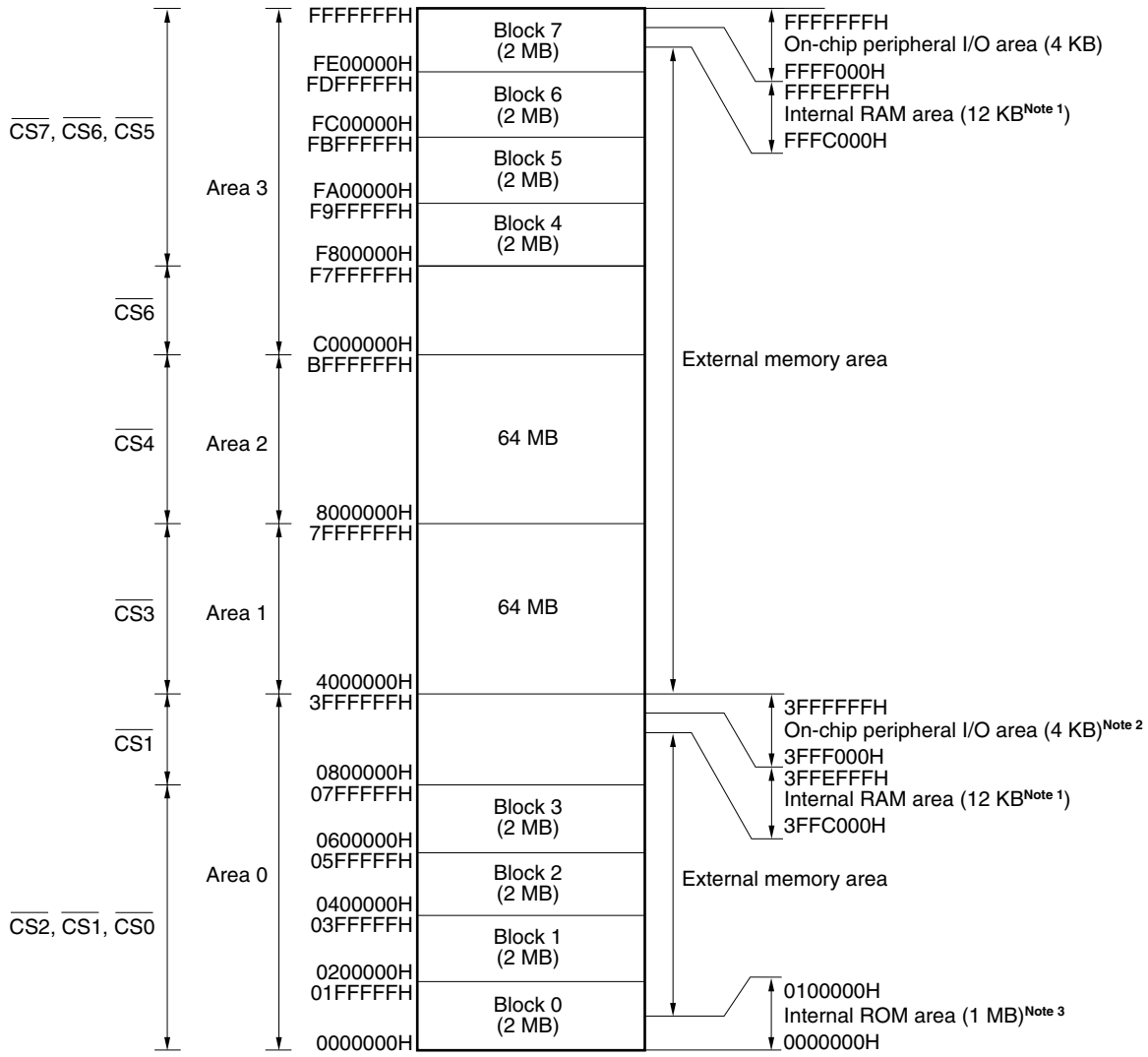
While accessing internal ROM and RAM, the address bus becomes undefined, and the data bus and external bus control signals are not output and enter the high-impedance state.

While accessing peripheral I/O, the address bus outputs the address data of the on-chip peripheral I/O currently being accessed, and the data bus control signals are not output and enter the high-impedance state. The external bus control signals become inactive.

4.3 Memory Block Function

The 256 MB memory space is divided into memory blocks of 2 MB and 64 MB units. The programmable wait function and bus cycle operation mode can be independently controlled for each block.

The area that can be used as program area is the 64 MB space of addresses 0000000H to 3FFFFFFH.



Notes 1. μPD703103, 703105: 4 KB
μPD703106, 703107, 70F3107: 10 KB

★ **2.** This area is access-prohibited. To access the on-chip peripheral I/O of this area, specify addresses FFFF000H to FFFFFFFH.

3. When in single-chip mode 1 and ROMless modes 0 and 1, this becomes an external memory area. When in single-chip mode 1, addresses 0100000H to 01FFFFFH become an internal ROM area.

4.3.1 Chip select control function

Of the 256 MB memory area, the lower 8 MB (0000000H to 07FFFFFH) and the higher 8 MB (F800000H to FFFFFFFH) can be divided into 2 MB memory blocks by chip area select control registers 0 and 1 (CSC0, CSC1) to control the chip select signal.

The memory area can be effectively used by dividing it into memory blocks using the chip select control function. The priority order is described below.

(1) Chip area select control registers 0, 1 (CSC0, CSC1)

These registers can be read/written in 16-bit units and become valid by setting each bit to 1.

If different chip select signal outputs are set to the same block, the priority order is controlled as follows.

CSC0: Peripheral I/O area: $\overline{CS0}$ > $\overline{CS2}$ > $\overline{CS1}$

CSC1: Peripheral I/O area: $\overline{CS7}$ > $\overline{CS5}$ > $\overline{CS6}$

If both the CS0n and CS2n bits of the CSC0 register are set to 0, $\overline{CS1}$ is output to the corresponding block (n = 0 to 3).

Similarly, if both the CS5n and CS7n bits of the CSC1 register are set to 0, $\overline{CS6}$ is output to the corresponding block (n = 0 to 3).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CSC0	CS33	CS32	CS31	CS30	CS23	CS22	CS21	CS20	CS13	CS12	CS11	CS10	CS03	CS02	CS01	CS00	FFFFF060H	2C11H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CSC1	CS43	CS42	CS41	CS40	CS53	CS52	CS51	CS50	CS63	CS62	CS61	CS60	CS73	CS72	CS71	CS70	FFFFF062H	2C11H

Bit position	Bit name	Function
15 to 0	CSnm (n = 0 to 7) (m = 0 to 3)	Chip Select Chip select is enabled by setting the CSnm bit to 1.

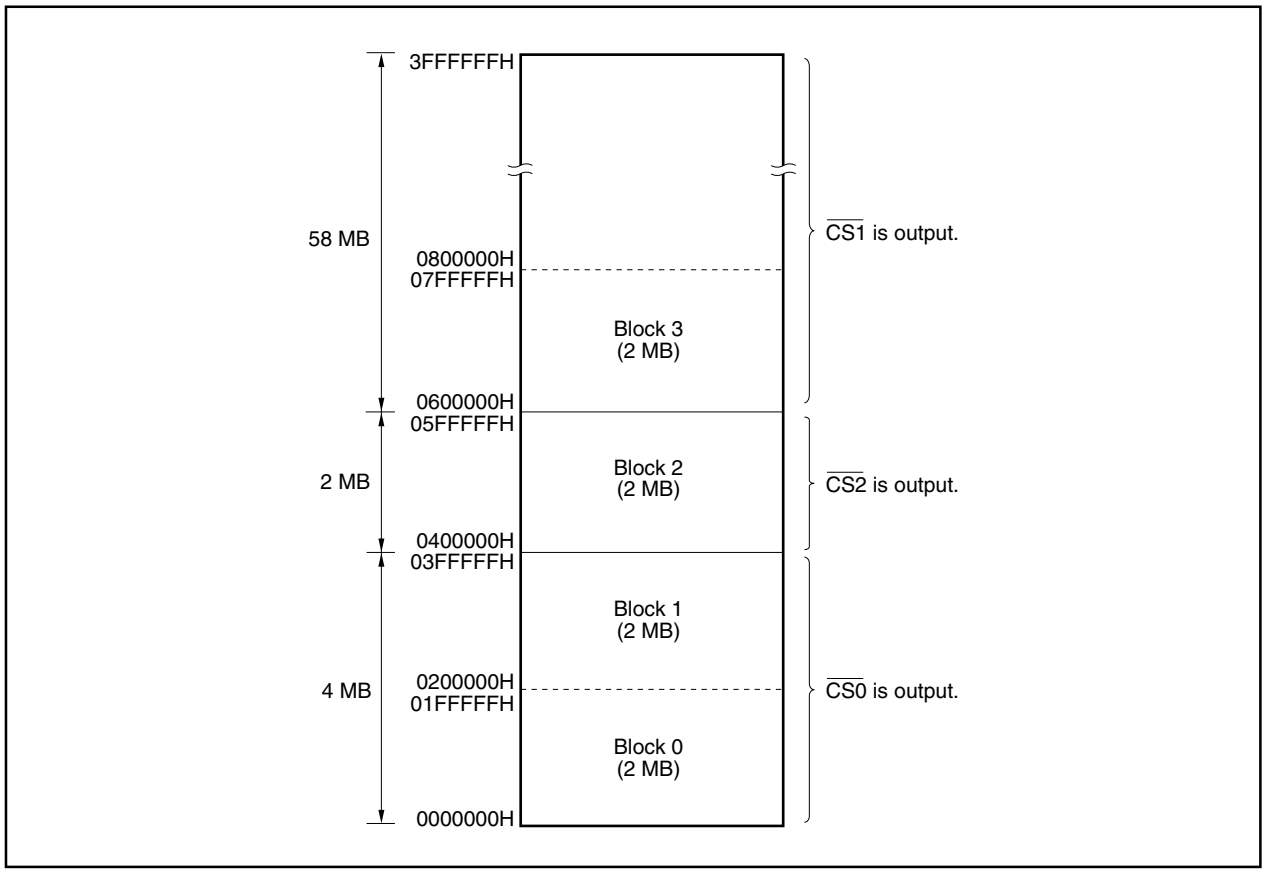
CSnm	CS operation
CS00	$\overline{CS0}$ output during block 0 access
CS01	$\overline{CS0}$ output during block 1 access.
CS02	$\overline{CS0}$ output during block 2 access.
CS03	$\overline{CS0}$ output during block 3 access.
CS10 to CS13	Setting has no meaning.
CS20	$\overline{CS2}$ output during block 0 access.
CS21	$\overline{CS2}$ output during block 1 access.
CS22	$\overline{CS2}$ output during block 2 access.
CS23	$\overline{CS2}$ output during block 3 access.
CS30 to CS33	Setting has no meaning.
CS40 to CS43	Setting has no meaning.
CS50	$\overline{CS5}$ output during block 7 access.
CS51	$\overline{CS5}$ output during block 6 access.
CS52	$\overline{CS5}$ output during block 5 access.
CS53	$\overline{CS5}$ output during block 4 access.
CS60 to CS63	Setting has no meaning.
CS70	$\overline{CS7}$ output during block 7 access.
CS71	$\overline{CS7}$ output during block 6 access.
CS72	$\overline{CS7}$ output during block 5 access.
CS73	$\overline{CS7}$ output during block 4 access.

The following diagram shows the $\overline{CS}$ signal that is enabled for area 0 when the CSC0 register is set to 0703H.

When the CSC0 register is set to 0703H, $\overline{CS0}$ and $\overline{CS2}$ are output to block 0 and block 1, but since $\overline{CS0}$ has priority over $\overline{CS2}$, $\overline{CS0}$ is output if the addresses of block 0 and block 1 are accessed.

If the address of block 3 is accessed, both the CS03 and CS23 bits of the CSC0 register are 0, and $\overline{CS1}$ is output.

Figure 4-1. Example When CSC0 Register Is Set to 0703H



4.4 Bus Cycle Type Control Function

In the V850E/MA1, the following external devices can be connected directly to each memory block.

- SRAM, external ROM, external I/O
- Page ROM
- EDO DRAM
- SDRAM

Connected external devices are specified by bus cycle type configuration registers 0 and 1 (BCT0 and BCT1).

4.4.1 Bus cycle type configuration registers 0, 1 (BCT0, BCT1)

(1) Bus cycle type configuration registers 0, 1 (BCT0, BCT1)

These registers can be read/written in 16-bit units.

Caution Write to the BCT0 and BCT1 registers after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BCT0 and BCT1 registers is complete. However, it is possible to access external memory areas whose initialization settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BCT0	ME3	0	BT31	BT30	ME2	0	0	BT20	ME1	0	BT11	BT10	ME0	0	0	BT00	FFFFF480H	8888H
$\overline{CSn}$ signal			$\overline{CS3}$					$\overline{CS2}$			$\overline{CS1}$					$\overline{CS0}$		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BCT1	ME7	0	0	BT70	ME6	0	BT61	BT60	ME5	0	0	BT50	ME4	0	BT41	BT40	FFFFF482H	8888H
$\overline{CSn}$ signal				$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$		

Bit position	Bit name	Function
15, 11, 7, 3 (BCT0), 15, 11, 7, 3 (BCT1)	MEn (n = 0 to 7)	Memory Controller Enable Sets memory controller operation enable for each chip select. MEn: 0 = Operation disabled; 1 = Operation enabled
8, 0 (BCT0), 12, 4 (BCT1)	BTn0 (n = 0, 2, 5, 7)	Bus Cycle Type Specifies the device to be connected to the $\overline{CSn}$ signal. BTn0: 0 = SRAM, external I/O; 1 = Page ROM
13, 12, 5, 4 (BCT0), 9, 8, 1, 0 (BCT1)	BTn1, BTn0 (n = 1, 3, 4, 6)	Bus Cycle Type Specifies the device to be connected to the $\overline{CSn}$ signal. BTn1, BTn0: 0, 0 = SRAM, external I/O; 0, 1 = Page ROM; 1, 0 = EDO DRAM; 1, 1 = SDRAM

4.5 Bus Access

4.5.1 Number of access clocks

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

Bus Cycle Configuration / Resource (Bus Width)	Instruction Fetch		Operand Data Access	
	Random Access	Sequential Access	Normal Access	Sequential Access
Internal ROM (32 bits)	2 Note 1	1	5	–
Internal RAM (32 bits)	1 or 2 Note 2	–	1	–

Notes 1. In the case of branch instructions

2. See **3.4.5 (2) Internal RAM area** when using internal RAM as program space.

Remark Unit: Clock/access

4.5.2 Bus sizing function

The bus sizing function controls the data bus width for each CS space. The data bus width is specified by using the bus size configuration register (BSC).

(1) Bus size configuration register (BSC)

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

Cautions 1. Write to the BSC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BSC register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

2. When the data bus width is specified as 8 bits, only the signals shown below become active.

$\overline{\text{LWR}}$: When accessing SRAM, external ROM, or external I/O (write cycle)

$\overline{\text{LCAS}}$: When accessing EDO DRAM

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset[Note]
BSC	0	BS70	0	BS60	0	BS50	0	BS40	0	BS30	0	BS20	0	BS10	0	BS00	FFFFF066H	0000H/5555H
$\overline{\text{CS}}$n signal		$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$		

Note When in single-chip mode 0, 1: 5555H
When in ROMless mode 0: 5555H
When in ROMless mode 1: 0000H

Bit position	Bit name	Function
14, 12, 10, 8, 6, 4, 2, 0	BSn0 (n = 0 to 7)	Data Bus Width Sets the data bus width of the CSn space.

BSn0	Data bus width of CSn space
0	8 bits
1	16 bits

4.5.3 Endian control function

The endian control function can be used to set processing of word data in memory using either the big endian method or the little endian method for each CS space selected with the chip select signals ($\overline{CS0}$ to $\overline{CS7}$). Switching of the endian method is specified using the endian configuration register (BEC).

Caution In the following areas, the data processing method is fixed to little endian, so the setting of the BEC register is invalid.

- **On-chip peripheral I/O area**
- **Internal ROM area**
- **Internal RAM area**
- **Fetch area for external memory**

(1) Endian configuration register (BEC)

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

Be sure to set bits 15, 13, 11, 9, 7, 5, 3, and 1 to 0. If they are set to 1, the operation is not guaranteed.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BEC	0	BE70	0	BE60	0	BE50	0	BE40	0	BE30	0	BE20	0	BE10	0	BE00	FFFFF068H	0000H
$\overline{CS}$n signal		$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$		

Bit position	Bit name	Function
14, 12, 10, 8, 6, 4, 2, 0	BEn0 (n = 0 to 7)	Big Endian Specifies the endian method.

BEn0	Endian control
0	Little endian method
1	Big endian method

Figure 4-2. Big Endian Addresses Within Word

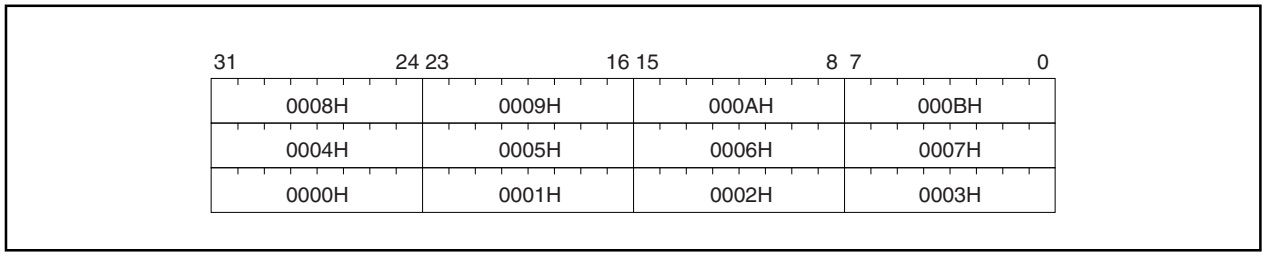
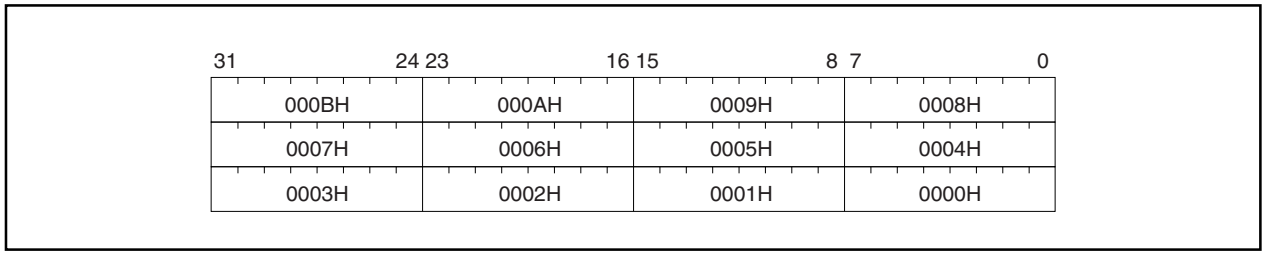


Figure 4-3. Little Endian Addresses Within Word



4.5.4 Big endian method usage restrictions in NEC development tools

(1) When using a debugger (ID850)

The big endian method is supported only in the memory window display.

(2) When using a compiler (CA850)

(a) Restrictions in C language

(i) There are restrictions for variables allocated to/located in the big endian space, as shown below.

- union cannot be used.
- bitfield cannot be used.
- Access with cast (changing access size) cannot be used.
- Variables with initial values cannot be used.

(ii) It is necessary to specify the following optimization inhibit options because optimization may cause a change in the access size.

- For global optimization part (opt850)… -Wo, -XTb
- For optimization depending on model part (impr850)… -Wi, +arg_reg_opt=OFF, +stld_trans_opt=OFF

The specification of the optimization inhibit options shown above is not necessary, however, if the access is not an access with cast or with masking/shifting[Note].

Note This is on the condition that a pattern that may cause the following optimization is not used. However, because it is very difficult for users to check the patterns completely in cases such as when several patterns are mixed (especially for optimization depending on model part), it is recommended that the optimization inhibit options shown above be specified.

[Related global optimization part]

- 1-bit set using bit or
  int i;
  i ^=1;
- 1-bit clear using bit and
  i &= ~1;
- 1-bit not using bit xor
  i ^= 1;
- 1-bit test using bit and
  if(i & 1);

[Related optimization depending on model part]

Accessing the same variable in a different size

- Cast
- Mask
- Shift

Example

```
int i, *ip;
char c;
  ⋮
c=*((char*)ip);
  ⋮
c = 0xff & i;
  ⋮
i = (i<<24) >>24;
```

(b) Restrictions in assembly language

For variables located in the big endian space, a quasi directive that secures an area of other than byte size (.hword, .word, .float, .shword) cannot be used.

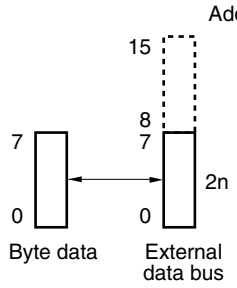
4.5.5 Bus width

The V850E/MA1 accesses peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access. All data is accessed in order starting from the lower order side.

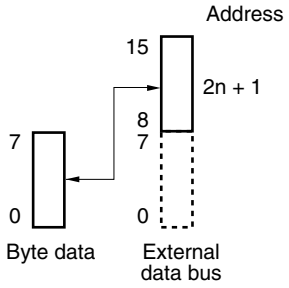
(1) Byte access (8 bits)

(a) When the data bus width is 16 bits (little endian)

<1> Access to even address (2n)

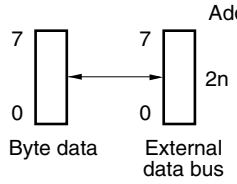


<2> Access to odd address (2n + 1)

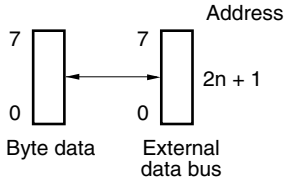


(b) When the data bus width is 8 bits (little endian)

<1> Access to even address (2n)

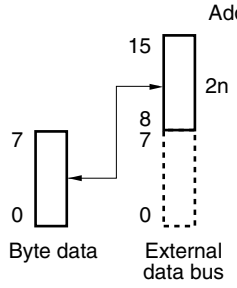


<2> Access to odd address (2n + 1)

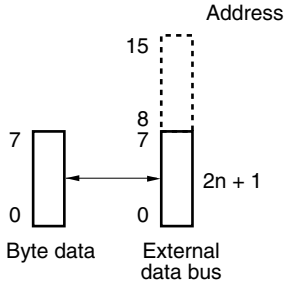


(c) When the data bus width is 16 bits (big endian)

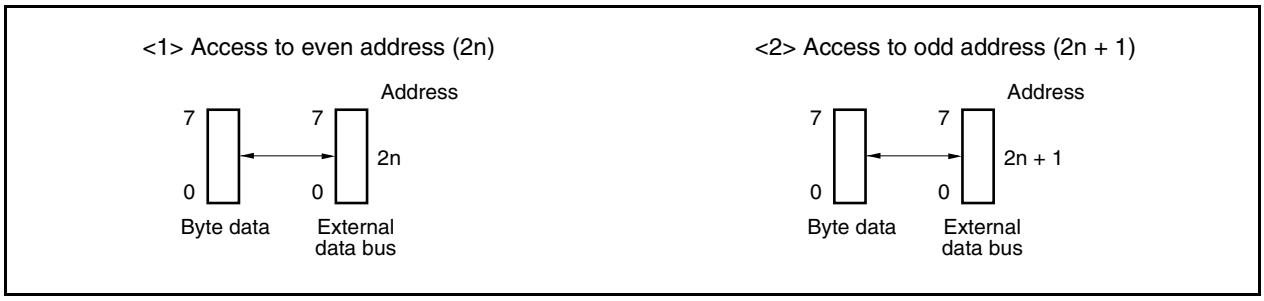
<1> Access to even address (2n)



<2> Access to odd address (2n + 1)

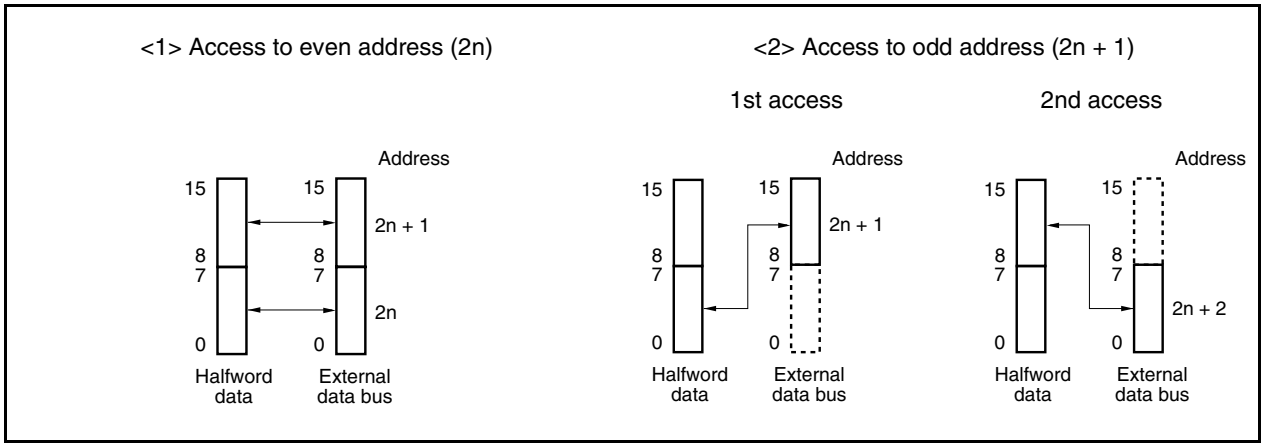


(d) When the data bus width is 8 bits (big endian)

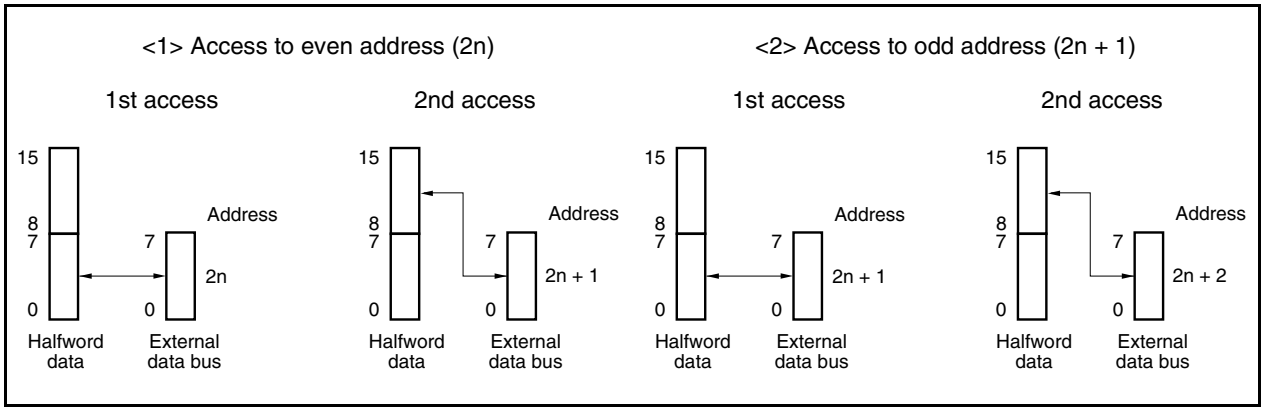


(2) Halfword access (16 bits)

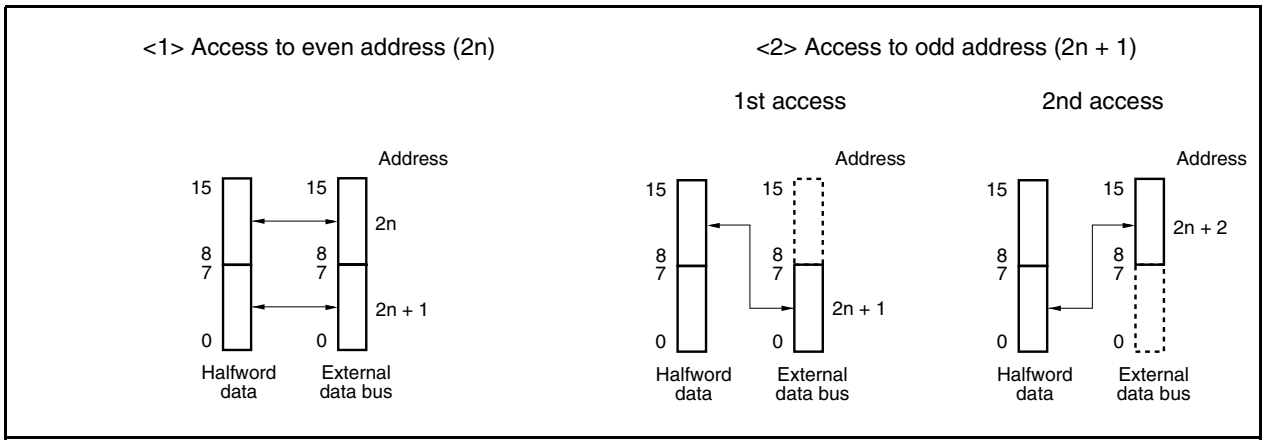
(a) When the bus width is 16 bits (little endian)



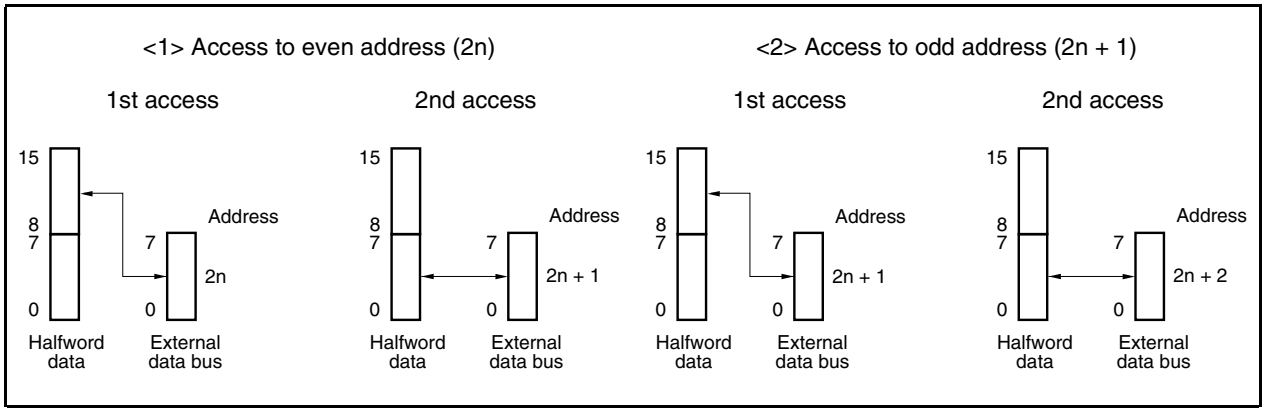
(b) When the data bus width is 8 bits (little endian)



(c) When the data bus width is 16 bits (big endian)



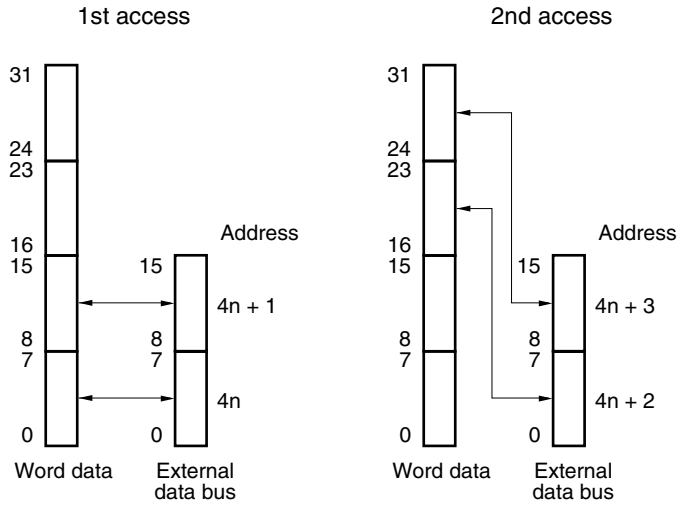
(d) When the data bus width is 8 bits (big endian)



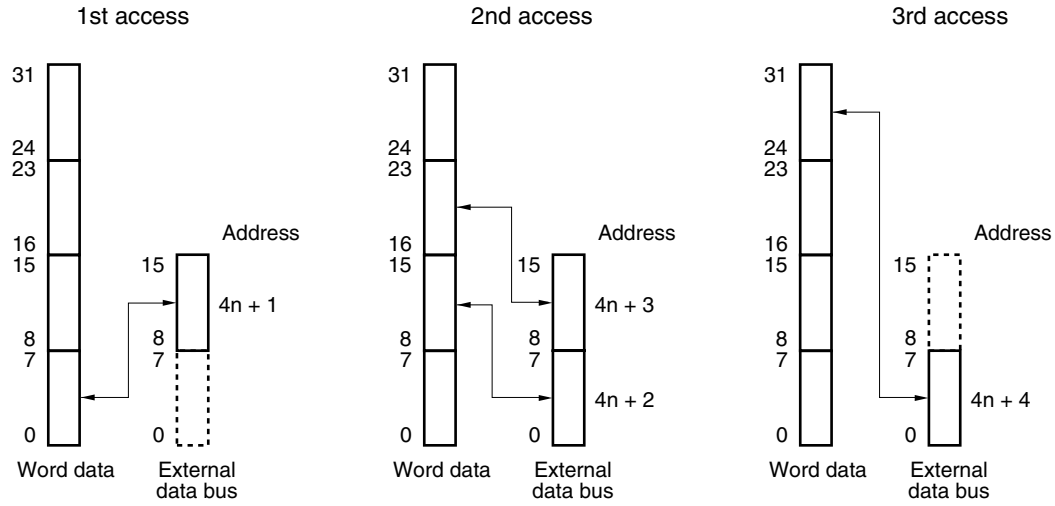
(3) Word access (32 bits)

(a) When the bus width is 16 bits (little endian) (1/2)

<1> Access to address 4n

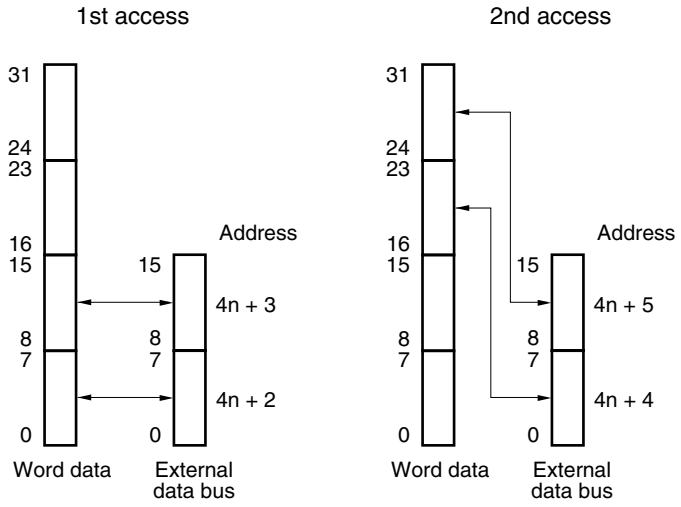


<2> Access to address 4n + 1

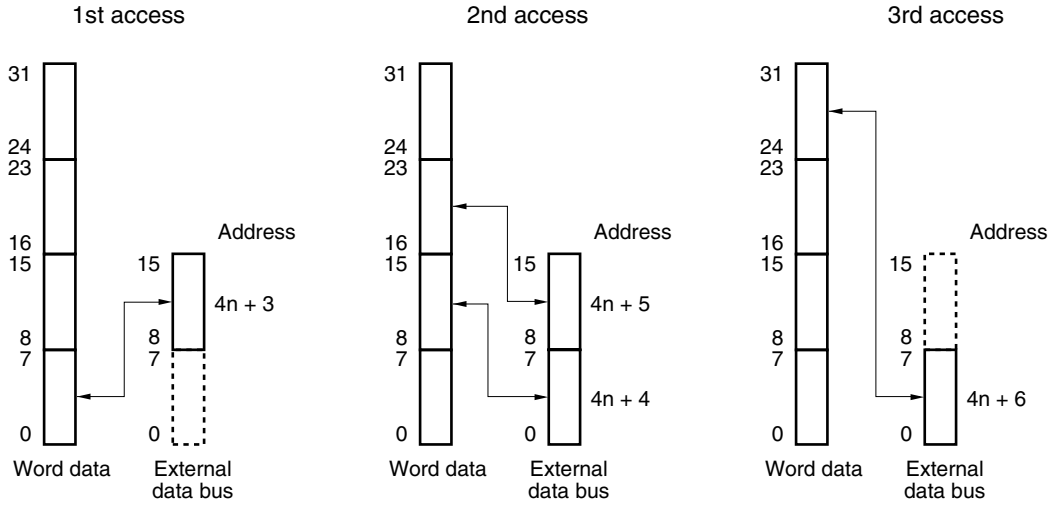


(a) When the bus width is 16 bits (little endian) (2/2)

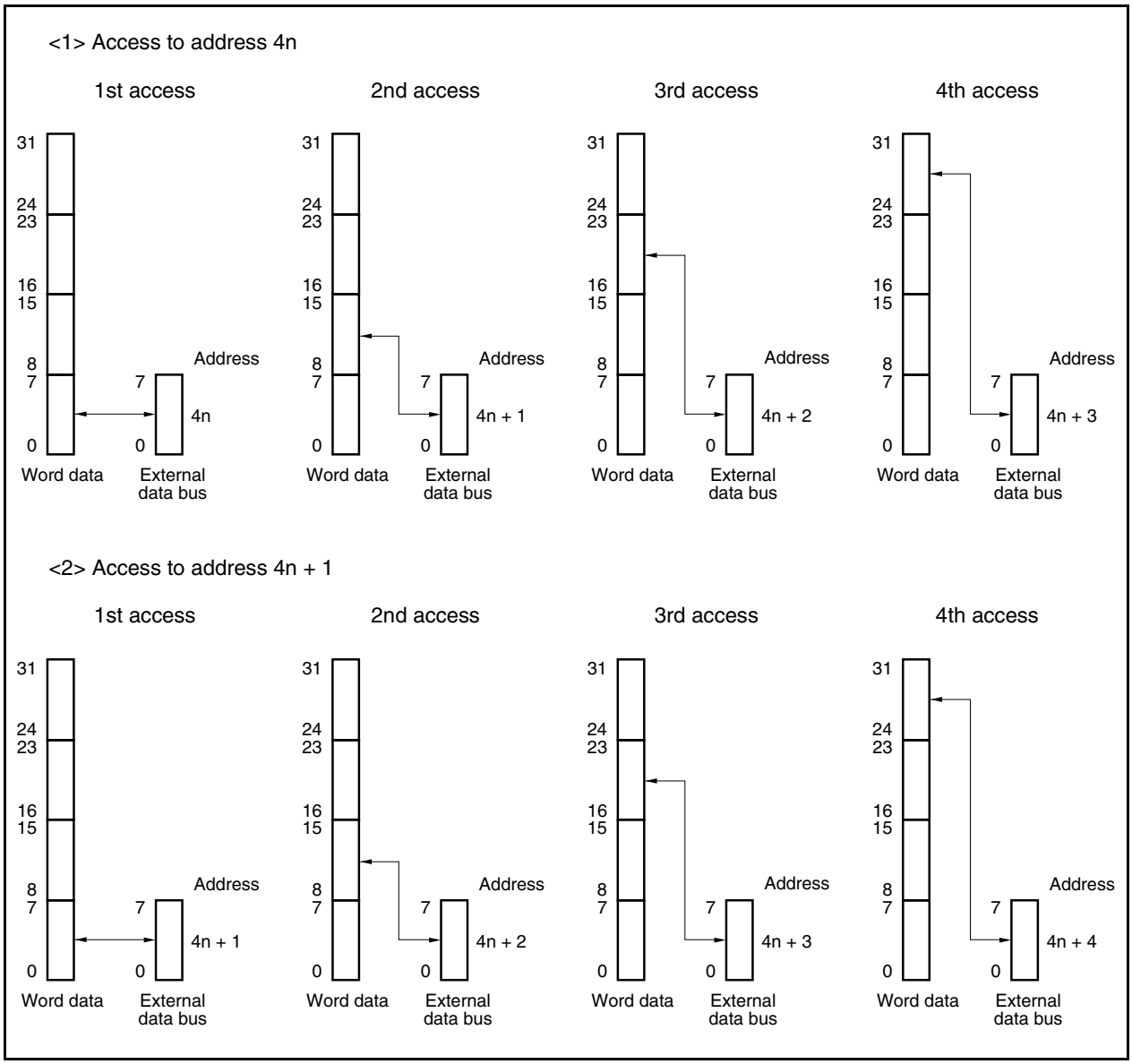
<3> Access to address 4n + 2



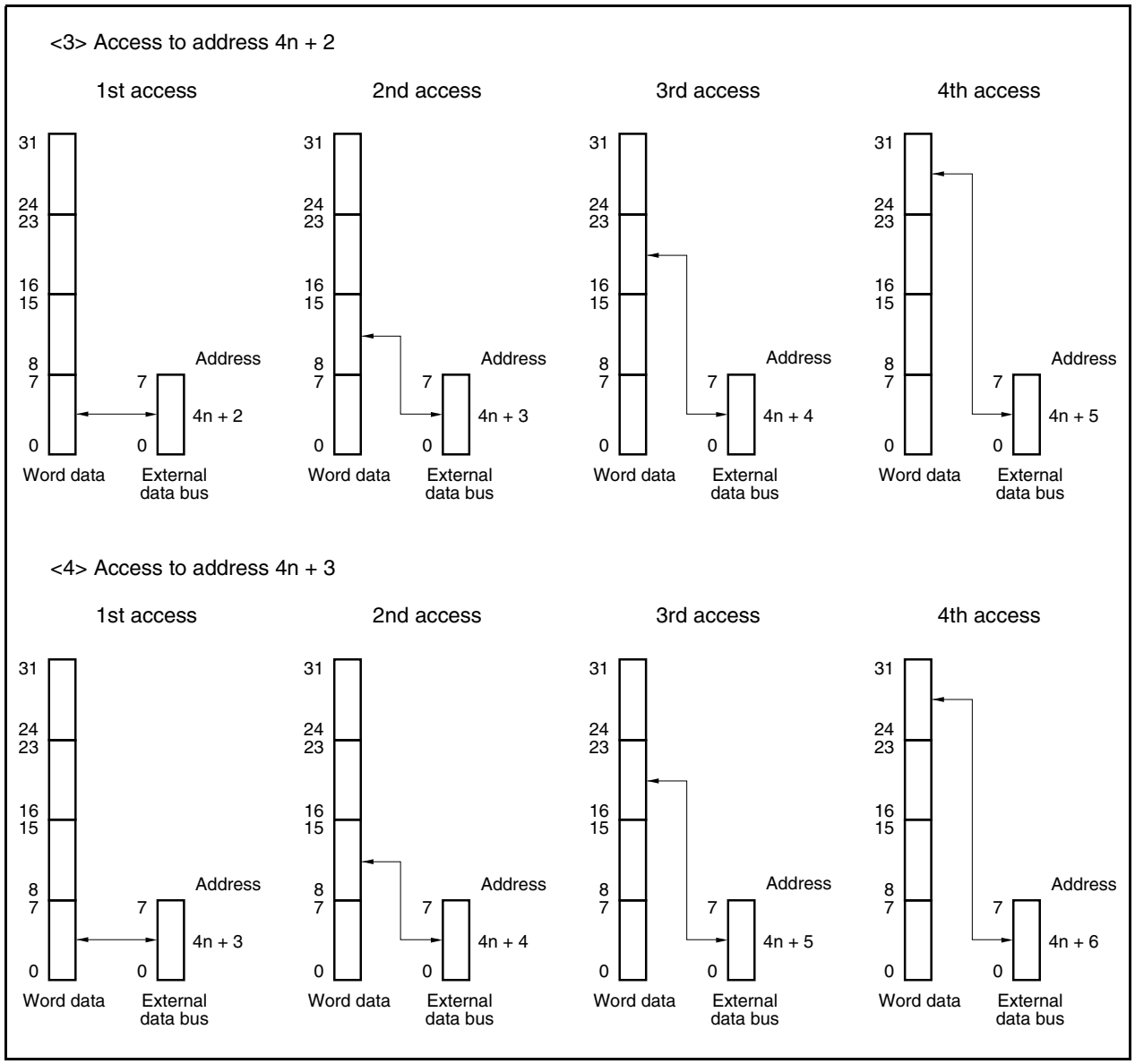
<4> Access to address 4n + 3



(b) When the data bus width is 8 bits (little endian) (1/2)

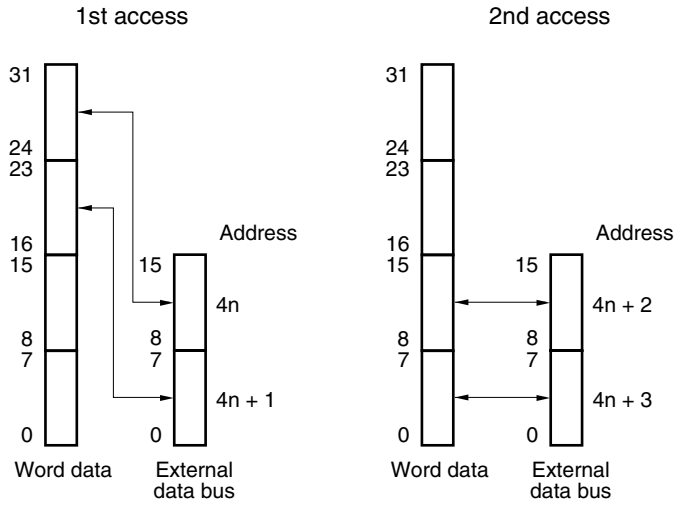


(b) When the data bus width is 8 bits (little endian) (2/2)

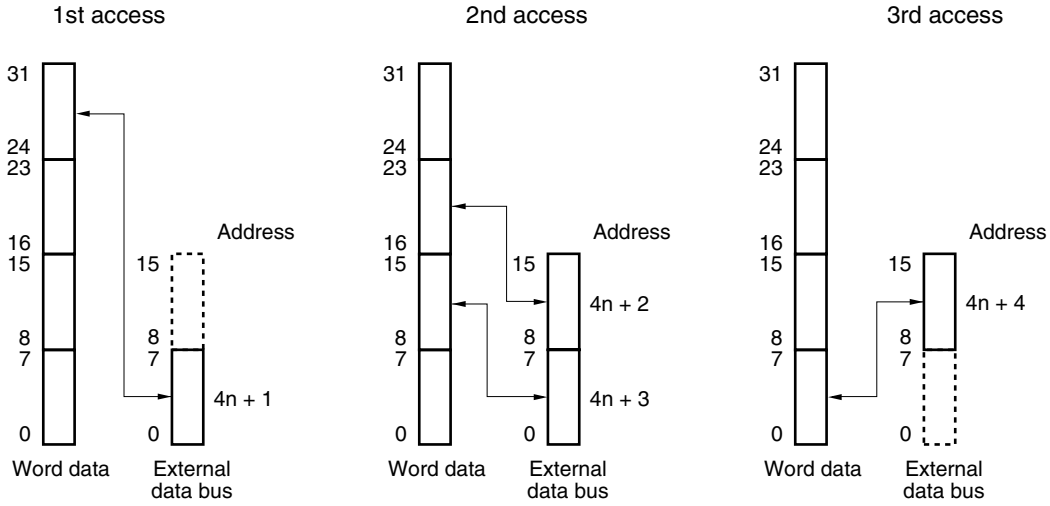


(c) When the data bus width is 16 bits (big endian) (1/2)

<1> Access to address 4n

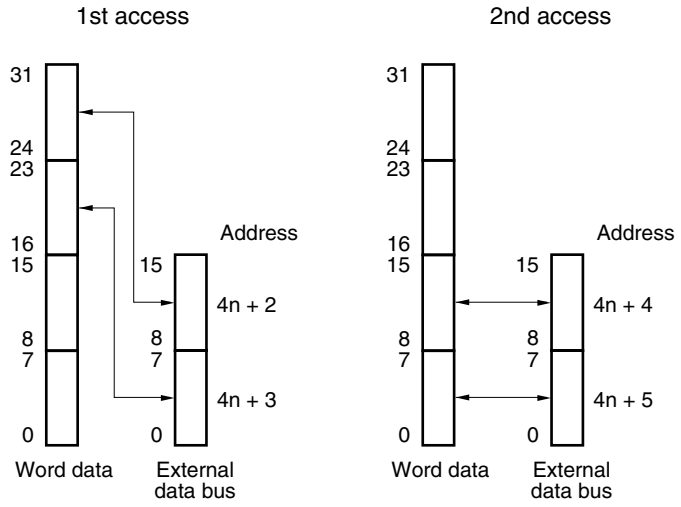


<2> Access to address 4n + 1

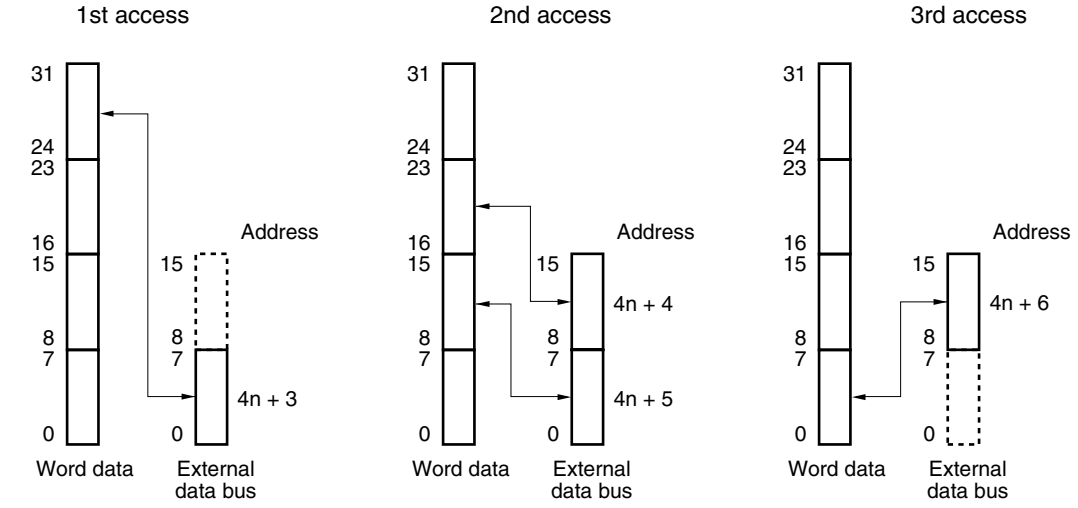


(c) When the data bus width is 16 bits (big endian) (2/2)

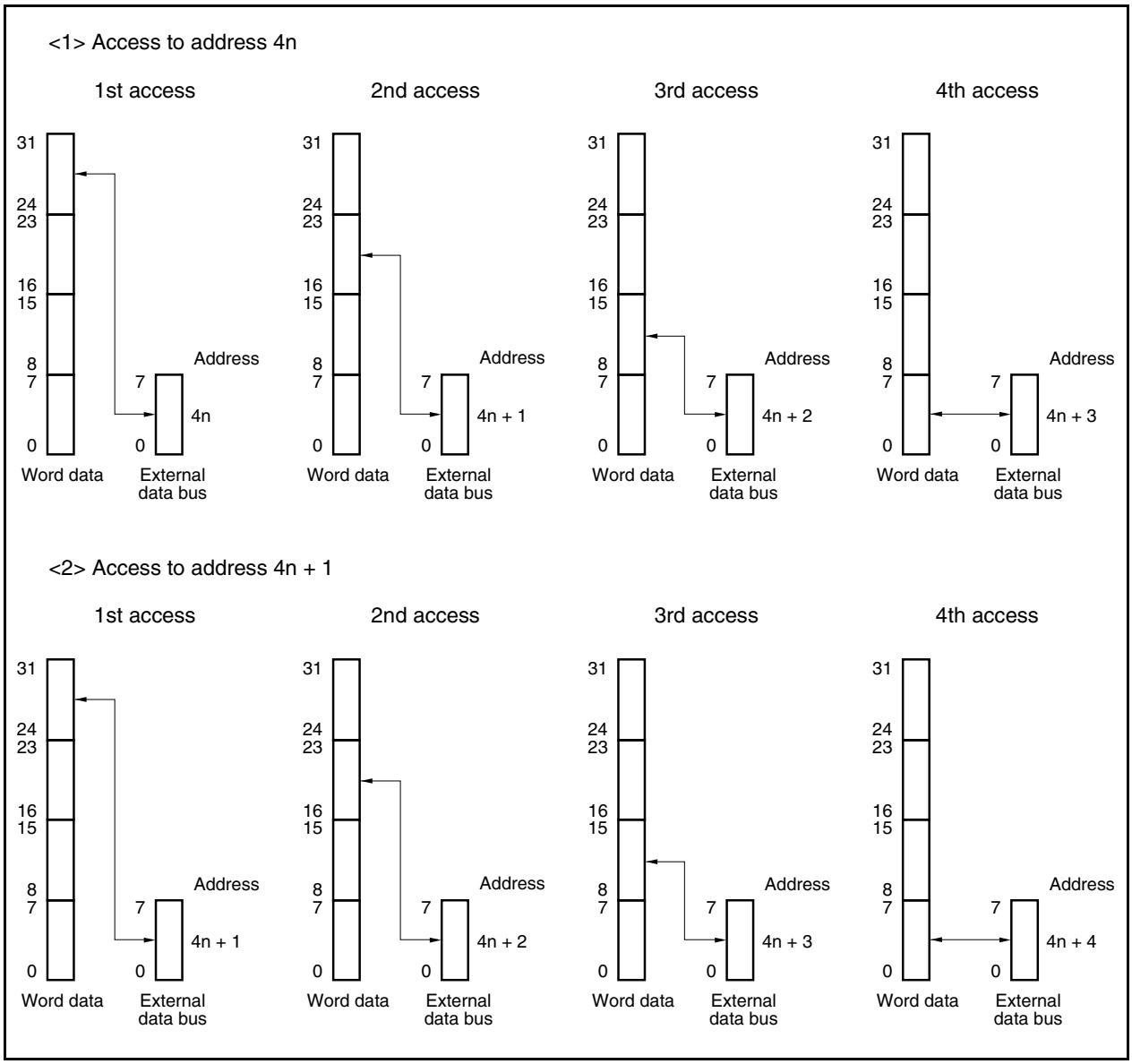
<3> Access to address 4n + 2



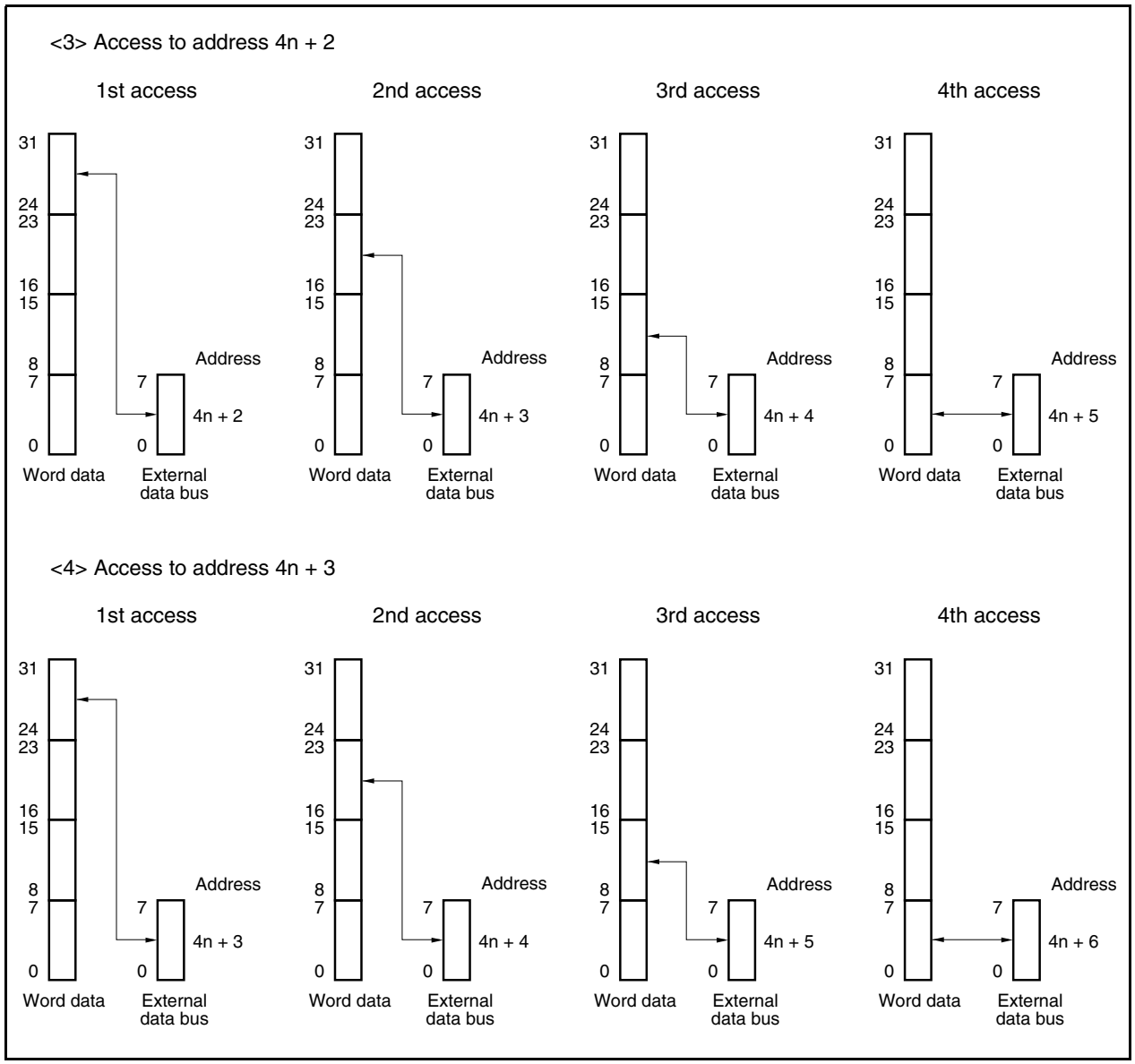
<4> Access to address 4n + 3



(d) When the data bus width is 8 bits (big endian) (1/2)



(d) When the data bus width is 8 bits (big endian) (2/2)



4.6 Wait Function

4.6.1 Programmable wait function

(1) Data wait control registers 0, 1 (DWC0, DWC1)

To facilitate interfacing with low-speed memory and I/Os, it is possible to insert up to 7 data wait states in the starting bus cycle for each CS space.

The number of wait states can be specified by program using data wait control registers 0 and 1 (DWC0, DWC1). Just after system reset, all blocks have 7 data wait states inserted.

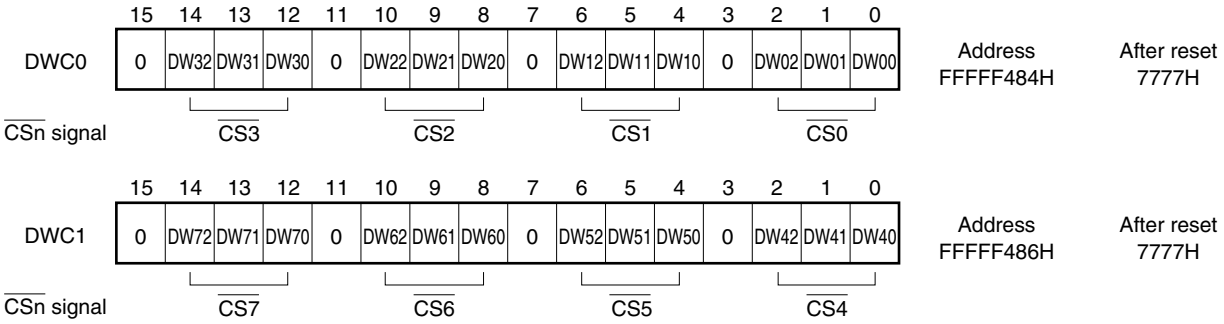
These registers can be read/written in 16-bit units.

Cautions 1. The internal ROM area and internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. The on-chip peripheral I/O area is also not subject to programmable wait states, with wait control performed by each peripheral function only.

2. In the following cases, the settings of registers DWC0 and DWC1 are invalid (wait control is performed by each memory controller).

- **Page ROM on-page access**
- **EDO DRAM access**
- **SDRAM access**

3. Write to the DWC0 and DWC1 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the DWC0 and DWC1 registers is complete. However, it is possible to access external memory areas whose initialization settings are complete.



Bit position	Bit name	Function
14 to 12, 10 to 8, 6 to 4, 2 to 0	DWn2 to DWn0 (n = 0 to 7)	Data Wait Specifies the number of wait states inserted in the CSn space.

DWn2	DWn1	DWn0	Number of wait states inserted in CSn space
0	0	0	Not inserted
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

(2) Address setup wait control register (ASC)

The V850E/MA1 allows insertion of address setup wait states before the SRAM/page ROM cycle (the setting of the ASC register in the EDO DRAM/SDRAM cycle is invalid).

The number of address setup wait states can be set with the ASC register for each CS space.

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

Caution During an address setup wait, the $\overline{\text{WAIT}}$ pin-based external wait function is disabled.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ASC	AC71	AC70	AC61	AC60	AC51	AC50	AC41	AC40	AC31	AC30	AC21	AC20	AC11	AC10	AC01	AC00	FFFFF48AH	FFFFH
$\overline{\text{CS}}$n signal	$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$			

Bit position	Bit name	Function
15 to 0	ACn1, ACn0 (n = 0 to 7)	Address Cycle Specifies the number of address setup wait states inserted before the SRAM/page ROM cycle for each CS space.

ACn1	ACn0	Number of wait states
0	0	Not inserted
0	1	1
1	0	2
1	1	3

(3) Bus cycle period control register (BCP)

In the V850E/MA1, the bus cycle period can be doubled during SRAM, external ROM, external I/O, page ROM, and EDO DRAM access. The bus cycle period is controlled using the BCP register. When the BCP bit of the BCP register is set to 1, the external bus operates at one half the frequency of the internal system clock.

Use the BUSCLK pin to output the half clock of the internal system clock. Specify the bus cycle period as "Double" with the BCP bit of the BCP register, then set the port CM mode control register (PMCCM) and port CM function control register (PFCCM).

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

Cautions 1. During a flyby DMA transfer for SRAM, external ROM, or external I/O, the $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ signals are always output, irrespective of the IOEN bit setting.
In page ROM and EDO DRAM cycles, on the other hand, the IOEN bit setting has no meaning.

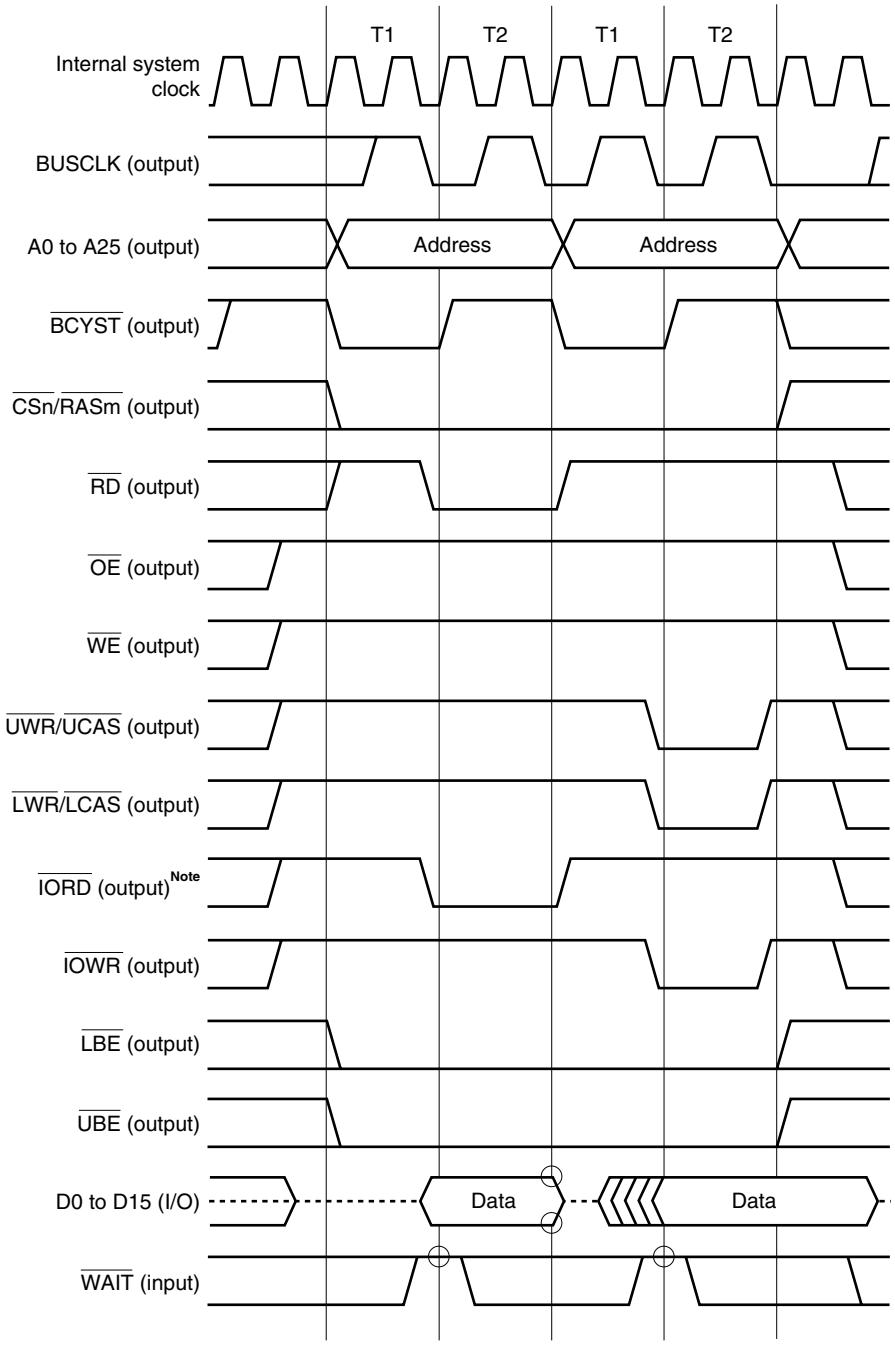
2. Write to the BCP register after reset, and then do not change the set values.

3. If the CLKOUT output mode is selected for the PCM1 pin by using the PMCCM register when the bus cycle period is doubled (BCP = 1), the bus cycle is half the frequency of the internal system clock, but the same frequency as the internal system clock is output from the PCM1 pin.

	7	6	5	4	3	2	1	0	Address	After reset
BCP	BCP	0	0	0	IOEN	0	0	0	FFFFF48CH	00H

Bit position	Bit name	Function
7	BCP	Bus Cycle Period Specifies the length of the bus cycle period. BCP: 0 — Bus cycle period: Normal BCP: 1 — Bus cycle period: Double
3	IOEN	$\overline{\text{IORD}}$, $\overline{\text{IOWR}}$ Enable Specifies whether to enable/disable the operation of $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ in SRAM, external ROM, and external I/O cycles. IOEN: 0 — Disables the operation of $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ in SRAM, external ROM, and external I/O cycles. IOEN: 1 — Enables the operation of $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ in SRAM, external ROM, and external I/O cycles.

Figure 4-4. Timing Example of Access to SRAM, External ROM, and External I/O (Read → Write)



Note When the IOEN bit of the BCP register is set to 1.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

4.6.2 External wait function

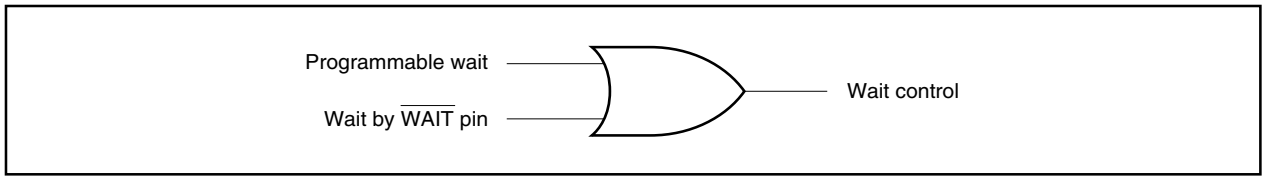
When an extremely slow device, I/O, or asynchronous system is connected, an arbitrary number of wait states can be inserted in the bus cycle by the external wait pin ($\overline{\text{WAIT}}$) for synchronization with the external device.

Just as with programmable waits, accessing internal ROM, internal RAM, and on-chip peripheral I/O areas cannot be controlled by external waits.

The external $\overline{\text{WAIT}}$ signal can be input asynchronously to CLKOUT and is sampled at the falling edge of the clock in the T1 and TW states of a bus cycle. If the setup/hold time in the sampling timing is not satisfied, the wait state may or may not be inserted in the next state.

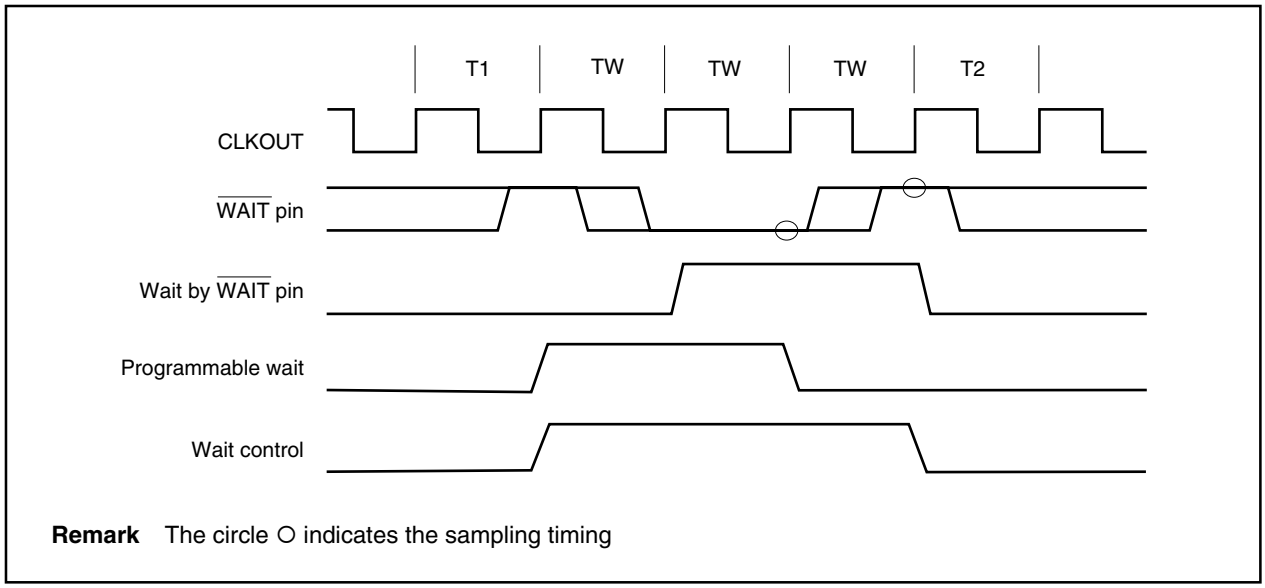
4.6.3 Relationship between programmable wait and external wait

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



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

Figure 4-5. Example of Wait Insertion



Remark The circle ○ indicates the sampling timing

4.6.4 Bus cycles in which wait function is valid

In the V850E/MA1, the number of waits can be specified according to the memory type specified for each memory block. The following shows the bus cycles in which the wait function is valid and the registers used for wait setting.

Table 4-1. Bus Cycles in Which Wait Function Is Valid

Bus Cycle			Type of Wait	Programmable Wait Setting			Wait from $\overline{\text{WAIT}}$ pin
				Register	Bit	Wait Count	
SRAM, external ROM, external I/O cycles			Address setup wait	ASC	ACn1, ACn0	0 to 3	– (invalid)
			Data access wait	DWC0, DWC1	DWn2 to DWn0	0 to 7	√ (valid)
Page ROM cycle			Address setup wait	ASC	ACn1, ACn0	0 to 3	– (invalid)
		Off-page	Data access wait	DWC0, DWC1	DWn2 to DWn0	0 to 7	√ (valid)
		On-page	Data access wait	PRC	PRW2 to PRW0	0 to 7	√ (valid)
EDO DRAM cycle	Read access	Off-page	RAS precharge	SCRm	RPC1m, RPC0m	1 to 3	– (invalid)
			Row address hold	SCRm	RHC1m, RHC0m	0 to 3	– (invalid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	– (invalid)
		On-page	CAS precharge	SCRm	CPC1m, CPC0m	0 to 3	– (invalid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	– (invalid)
	Write access	Off-page	RAS precharge	SCRm	RPC1m, RPC0m	1 to 3	– (invalid)
			Row address hold	SCRm	RHC1m, RHC0m	0 to 3	– (invalid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	– (invalid)
		On-page	CAS precharge	SCRm	CPC1m, CPC0m	1 to 3	– (invalid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	– (invalid)
	CBR refresh cycle		RAS precharge	RWC	RRW1, RRW0	0 to 3	– (invalid)
			RAS active width	RWC	RCW2 to RCW0	1 to 7	– (invalid)
	CBR self-refresh cycle		RAS precharge	RWC	RRW1, RRW0	0 to 3	– (invalid)
			RAS active width	RWC	RCW2 to RCW0	1 to 7	– (invalid)
			Self-refresh release width	RWC	SRW2 to SRW0	0 to 7	– (invalid)
★ SDRAM cycle			Row address precharge	SCRm	BCW1m, BCW0m	1 to 3	– (invalid)
DMA flyby transfer cycle	External I/O → SRAM		Data access wait	DWC0, DWC1	DWn2 to DWn0	0 to 7	√ (valid)
	DRAM → external I/O	Off-page	RAS precharge	SCRm	RPC1m, RPC0m	1 to 3	– (invalid)
			Row address hold	SCRm	RHC1m, RHC0m	0 to 3	– (invalid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	√ (valid)
		On-page	CAS precharge	SCRm	CPC1m, CPC0m	0 to 3	– (invalid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	√ (valid)
	External I/O → DRAM	Off-page	RAS precharge	SCRm	RPC1m, RPC0m	1 to 3	– (invalid)
			Row address hold	SCRm	RHC1m, RHC0m	0 to 3	√ (valid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	– (invalid)
		On-page	CAS precharge	SCRm	CPC1m, CPC0m	1 to 3	√ (valid)
			Data access wait	SCRm	DAC1m, DAC0m	0 to 3	– (invalid)

Remark n = 0 to 7, m = 1, 3, 4, 6

4.7 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state to meet the data output float delay time (t_{DF}) on memory read access for each CS space. The bus cycle following the T2 state starts after the idle state is inserted.

An idle state is inserted at the timing shown below.

- After read/write cycles for SRAM, external I/O, or external ROM
- After a read cycle for page ROM
- After a read cycle for EDO DRAM (no idle state is inserted when accessing the same CS space)
- After a read cycle for SDRAM

The idle state insertion setting can be specified by program using the bus cycle control register (BCC).

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

For the timing when an idle state is inserted, see the memory access timings in Chapter 5.

(1) Bus cycle control register (BCC)

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

Cautions 1. The internal ROM area, internal RAM area, and on-chip peripheral I/O area are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BCC register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCC	BC71	BC70	BC61	BC60	BC51	BC50	BC41	BC40	BC31	BC30	BC21	BC20	BC11	BC10	BC01	BC00
$\overline{\text{CSn}}$ signal	$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$	

Address: FFFFF488H

After reset: FFFFH

Bit position	Bit name	Function
15 to 0	BCn1, BCn0 (n = 0 to 7)	Data Cycle Specifies the insertion of an idle state in the CSn space.

BCn1	BCn0	Idle state in CSn space
0	0	Not inserted
0	1	1
1	0	2
1	1	3

4.8 Bus Hold Function

4.8.1 Function outline

If the PCM2 and PCM3 pins are specified in the control mode, the $\overline{\text{HLDAK}}$ and $\overline{\text{HLDRQ}}$ functions become valid.

If it is determined that the $\overline{\text{HLDRQ}}$ pin has become active (low level) as a bus mastership request from another bus master, the external address/data bus and each strobe pin are shifted to high impedance and then released (bus hold state). If the $\overline{\text{HLDRQ}}$ pin becomes inactive (high level) and the bus mastership request is canceled, driving of these pins begins again.

During the bus hold period, the internal operations of the V850E/MA1 continue until the external memory is accessed.

The bus hold state can be known by the $\overline{\text{HLDAK}}$ pin becoming active (low level). The period from when the $\overline{\text{HLDRQ}}$ pin becomes active (low level) to when the $\overline{\text{HLDAK}}$ pin becomes active (low level) is at least 2 clocks.

In a multiprocessor configuration, etc., a system with multiple bus masters can be configured.

State	Data Bus Width	Access Type	Timing at Which Bus Hold Request Cannot Be Acknowledged
CPU bus lock	16 bits	Word access for even address	Between first and second accesses
		Word access for odd address	Between first and second accesses
			Between second and third accesses
		Halfword access for odd address	Between first and second accesses
	8 bits	Word access	Between first and second accesses
			Between second and third accesses
			Between third and forth accesses
		Halfword access	Between first and second accesses
Read modify write access of bit manipulation instruction	–	–	Between read access and write access

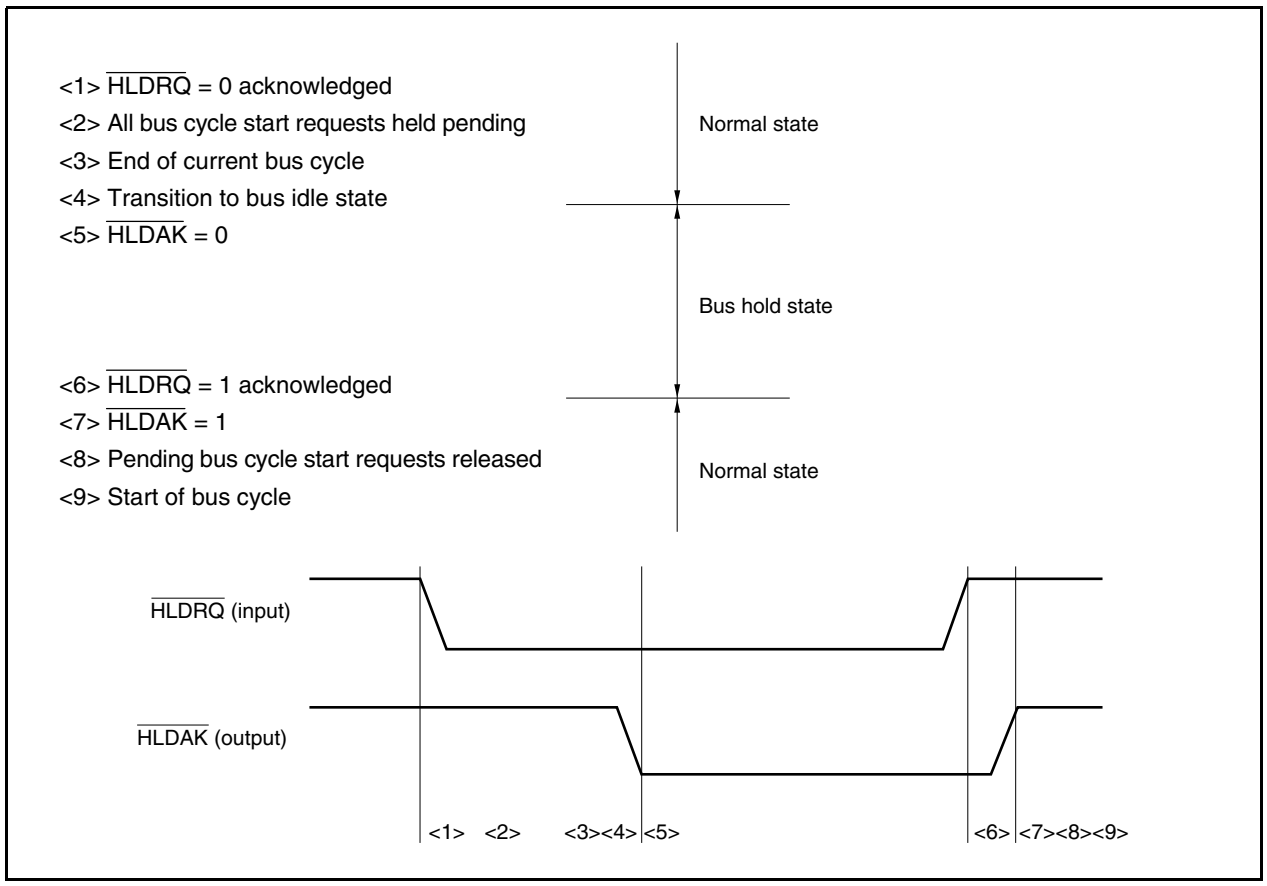
Cautions 1. When an external bus master accesses EDO DRAM during a bus hold state, make sure that the external bus master secures the RAS precharge time.

2. When an external bus master accesses SDRAM during a bus hold state, make sure that the external bus master executes the all bank precharge command.
The CPU always executes the all bank precharge command to release a bus hold state. In a bus hold state, do not allow an external bus master to change the SDRAM command register value.

3. The $\overline{\text{HLDRQ}}$ function is invalid during a reset period. The $\overline{\text{HLDAK}}$ pin becomes active either immediately after or after the insertion of a 1-clock address cycle from when the $\overline{\text{RESET}}$ pin is set to inactive following the simultaneous activation of the $\overline{\text{RESET}}$ and $\overline{\text{HLDRQ}}$ pins.
When a bus master other than the V850E/MA1 is externally connected, use the $\overline{\text{RESET}}$ signal for bus arbitration at power-on.

4.8.2 Bus hold procedure

The procedure of the bus hold function is illustrated below.



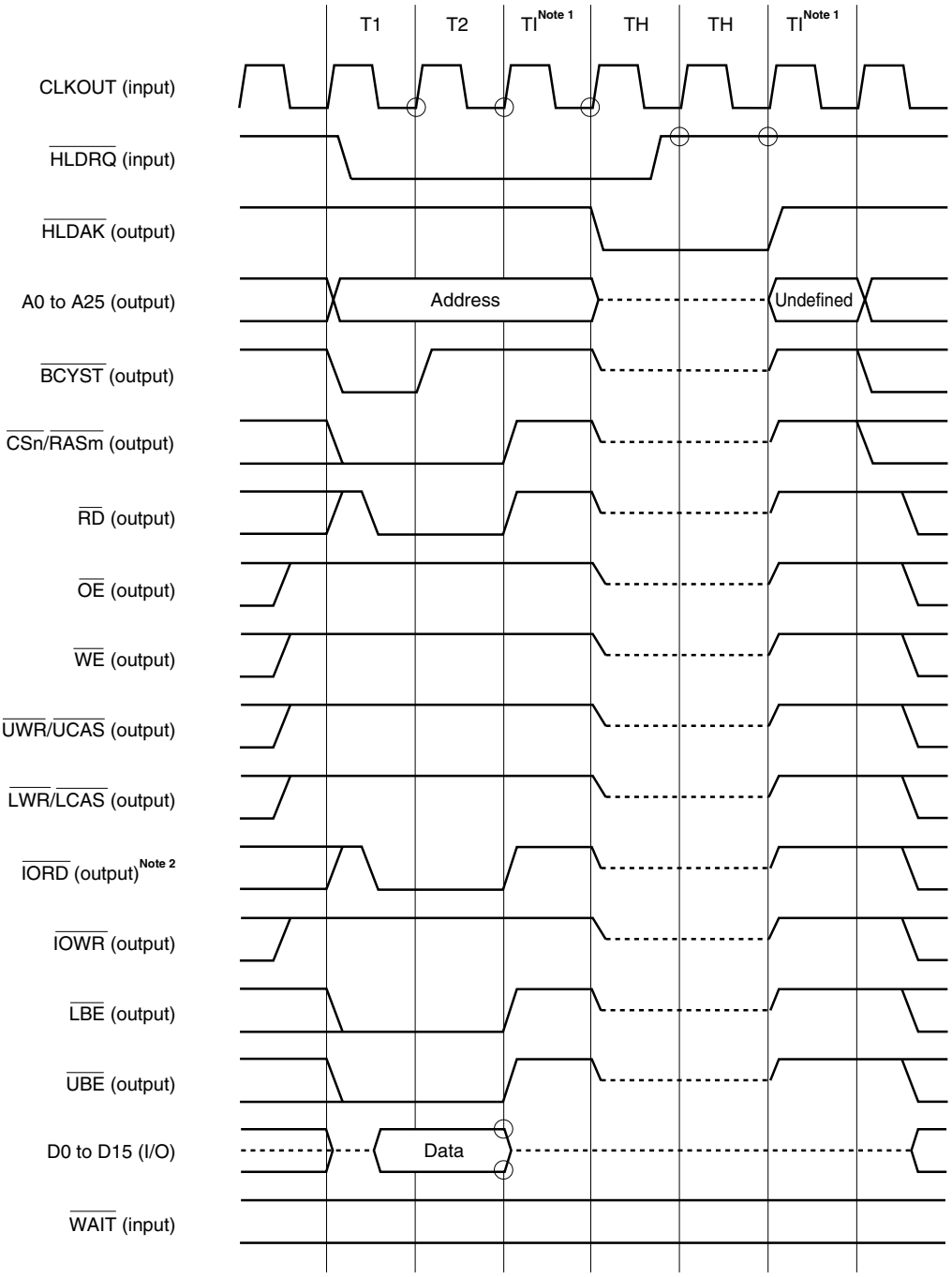
4.8.3 Operation in power-save mode

In the software STOP or IDLE mode, the internal system clock is stopped. Consequently, the bus hold state is not set since the $\overline{\text{HLDRQ}}$ pin cannot be acknowledged even if it becomes active.

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

4.8.4 Bus hold timing (SRAM)

(1) SRAM (when read, no idle states inserted)



Notes 1. This idle state (TI) is independent of the BCC register setting.

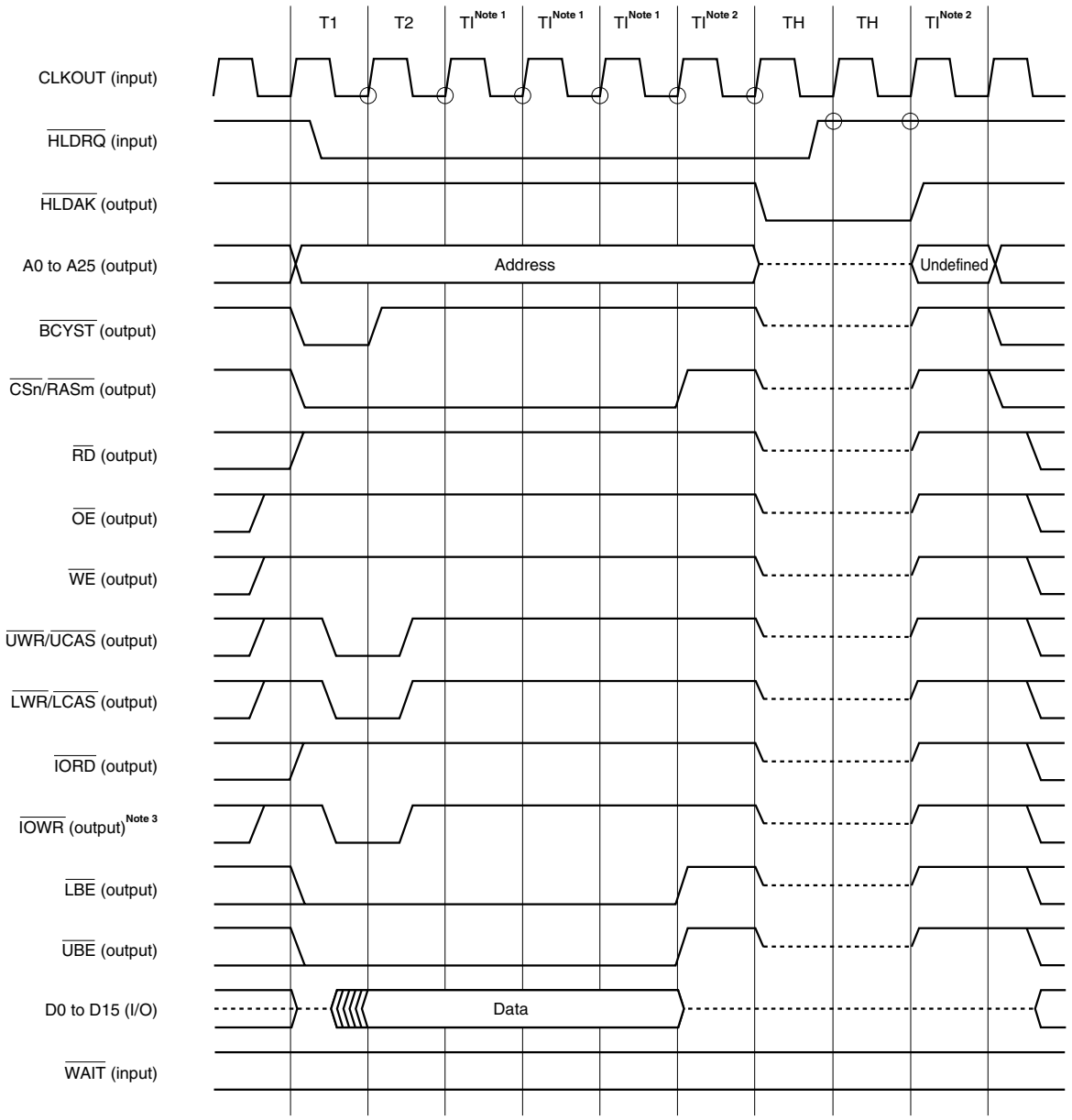
2. When the IOEN bit of the BCP register is set to 1.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

(2) SRAM (when written, three idle states inserted)



Notes 1. This idle state (TI) is inserted by means of a BCC register setting.

2. This idle state (TI) is independent of the BCC register setting.

3. When the IOEN bit of the BCP register is set to 1.

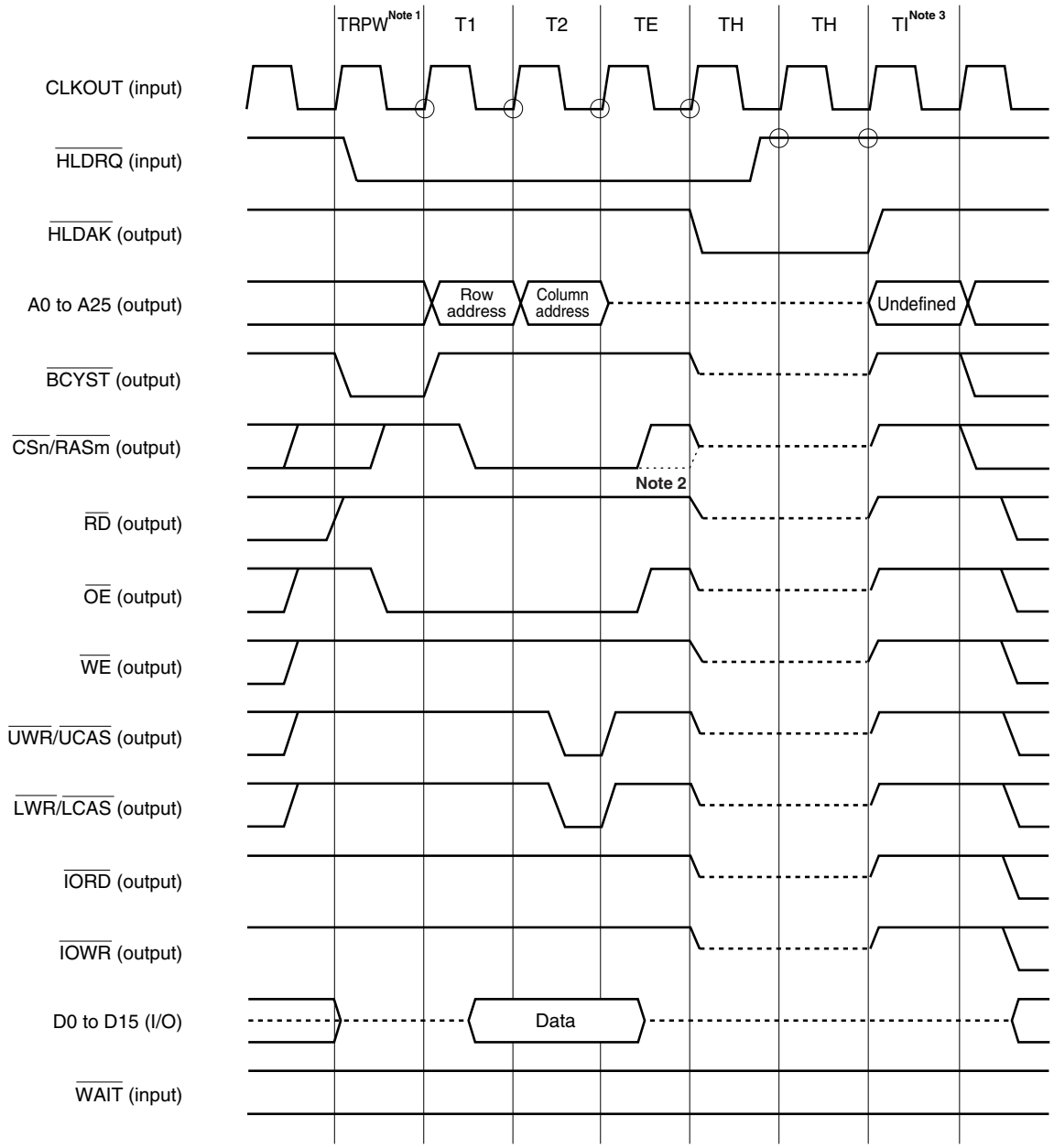
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

4.8.5 Bus hold timing (EDO DRAM)

(1) EDO DRAM (when read, no idle states inserted)



Notes 1. TRPW is always inserted for 1 or more cycles.

2. This timing applies when in the RAS hold mode.

3. This idle state (TI) is independent of the BCC register setting.

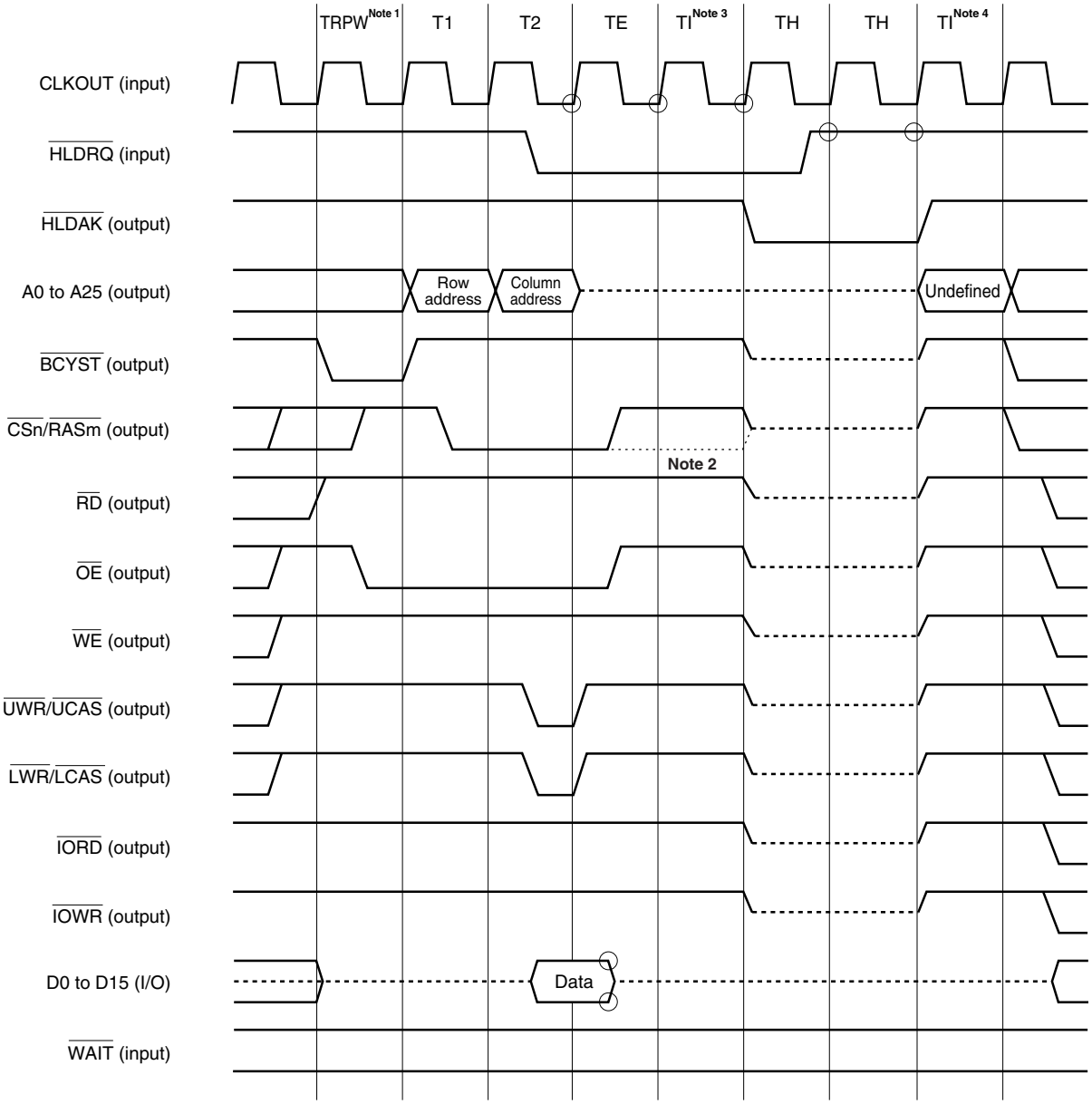
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

4. Timing from DRAM access to bus hold state.

(2) EDO DRAM (when read, three idle states inserted)



Notes 1. TRPW is always inserted for 1 or more cycles.

2. This timing applies when in the RAS hold mode.

3. This idle state (TI) is inserted by means of a BCC register setting. The number of idle states (TI) to be inserted depends on the timing of bus hold request acknowledgement.

4. This idle state (TI) is independent of the BCC register setting.

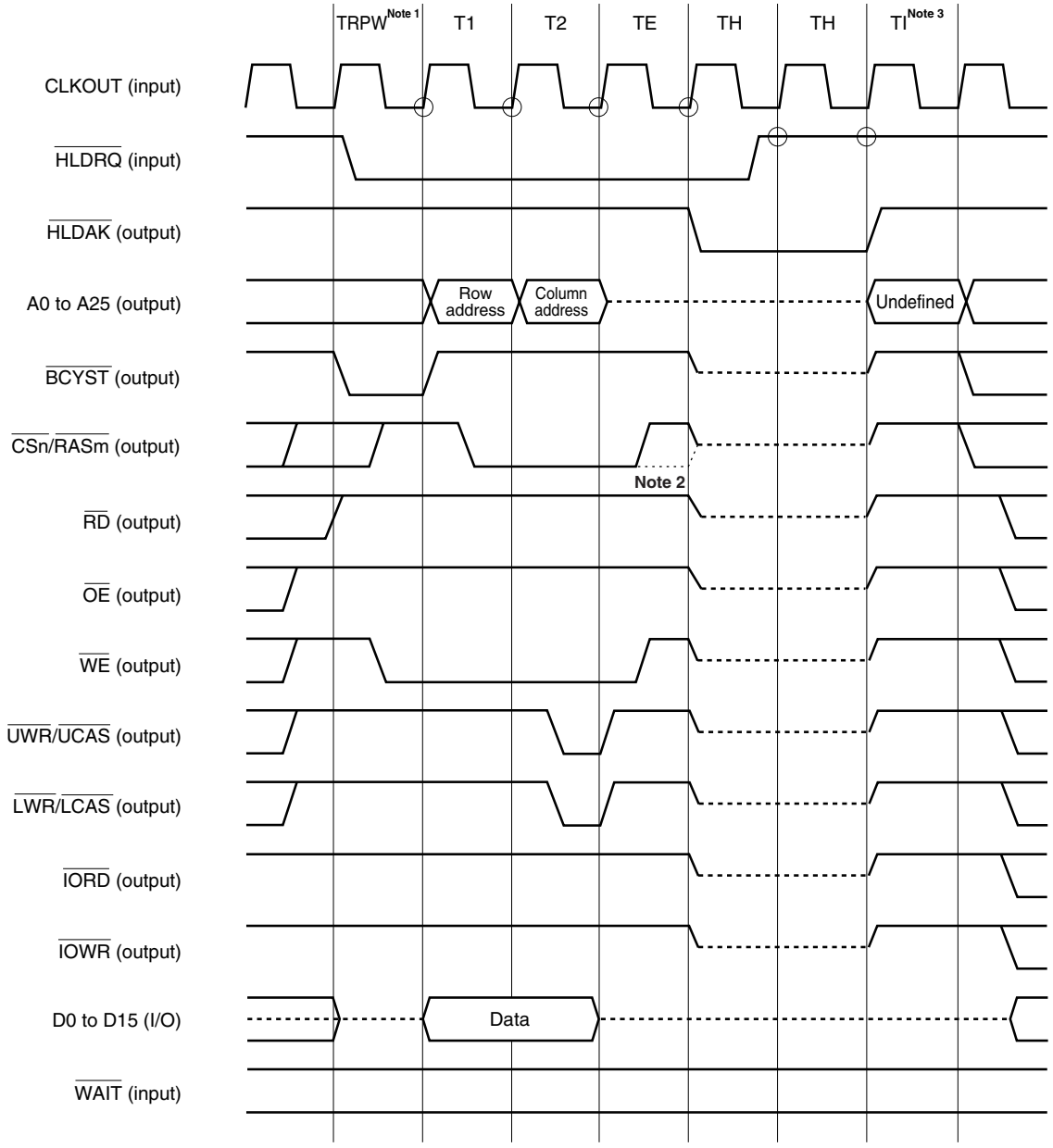
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

4. Timing from DRAM access to bus hold state.

(3) EDO DRAM (when written)



Notes 1. TRPW is always inserted for 1 or more cycles.

2. This timing applies when in the RAS hold mode.

3. This idle state (TI) is independent of the BCC register setting.

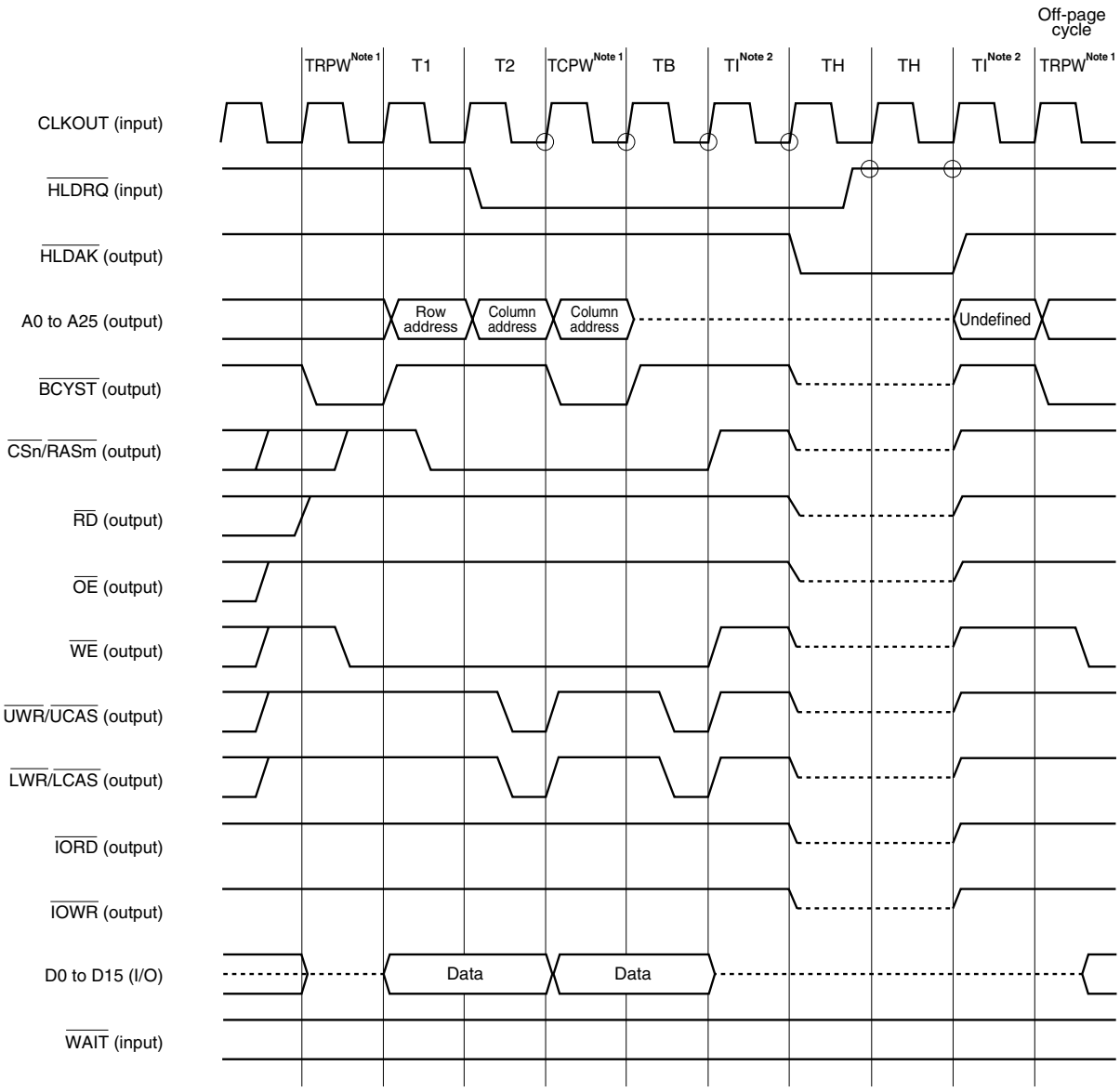
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

4. Timing from DRAM access to bus hold state.

(4) EDO DRAM (when written, when bus hold request acknowledged during on-page access)



Notes 1. TRPW and TCPW are always inserted for 1 or more cycles.

2. This idle state (TI) is independent of the BCC register setting.

Remarks 1. The circle ○ indicates the sampling timing.

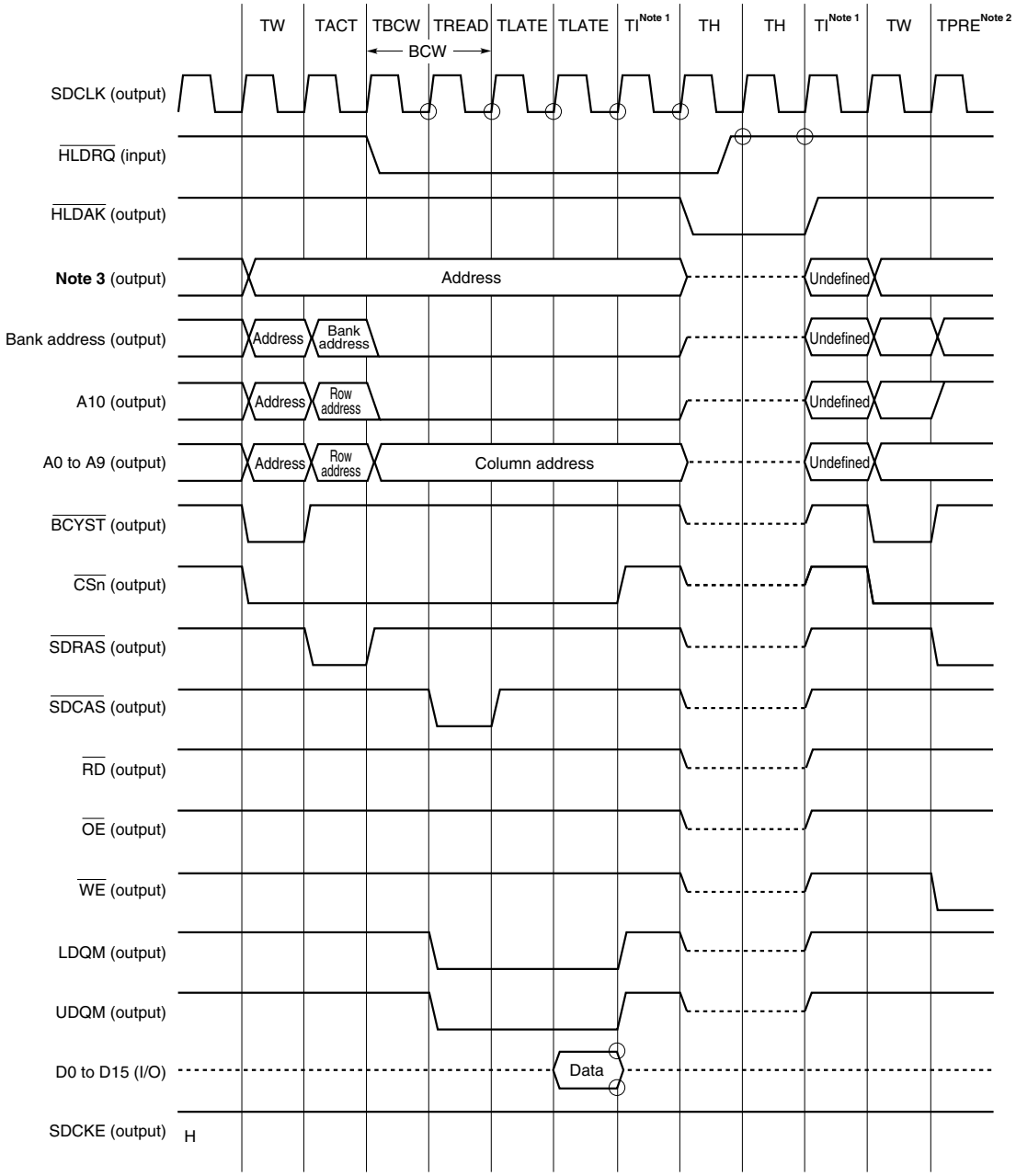
2. The broken lines indicate the high-impedance state.

3. n = 0 to 7, m = 1, 3, 4, 6

4. Timing from DRAM access to bus hold state.

4.8.6 Bus hold timing (SDRAM)

(1) SDRAM (when read, latency = 2, no idle states inserted)



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. The all bank precharge command is always executed.

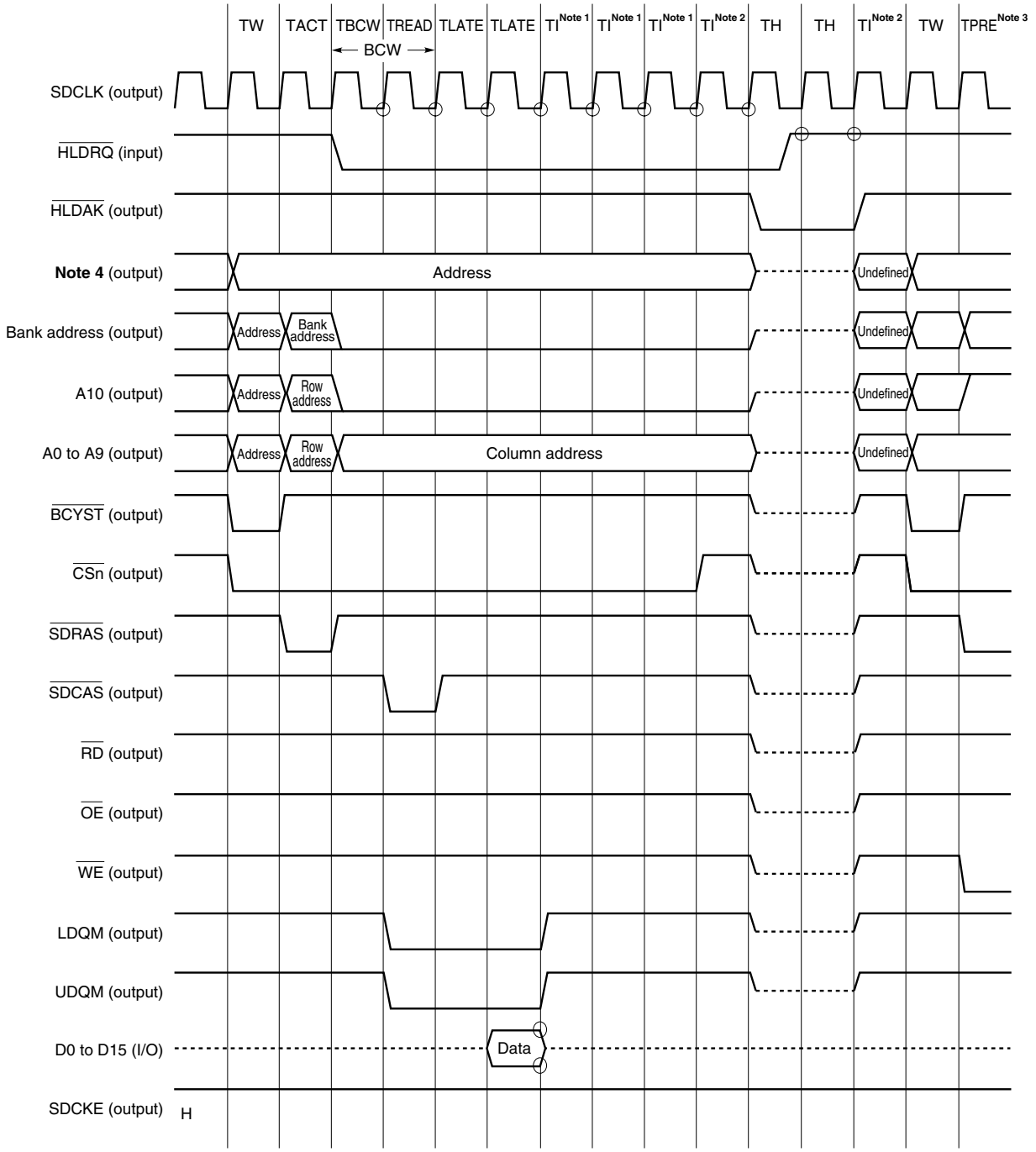
3. Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

(2) SDRAM (when read, latency = 2, three idle states inserted)



Notes 1. This idle state (TI) is inserted by means of a BCC register setting.

2. This idle state (TI) is independent of the BCC register setting.

3. The all bank precharge command is always executed.

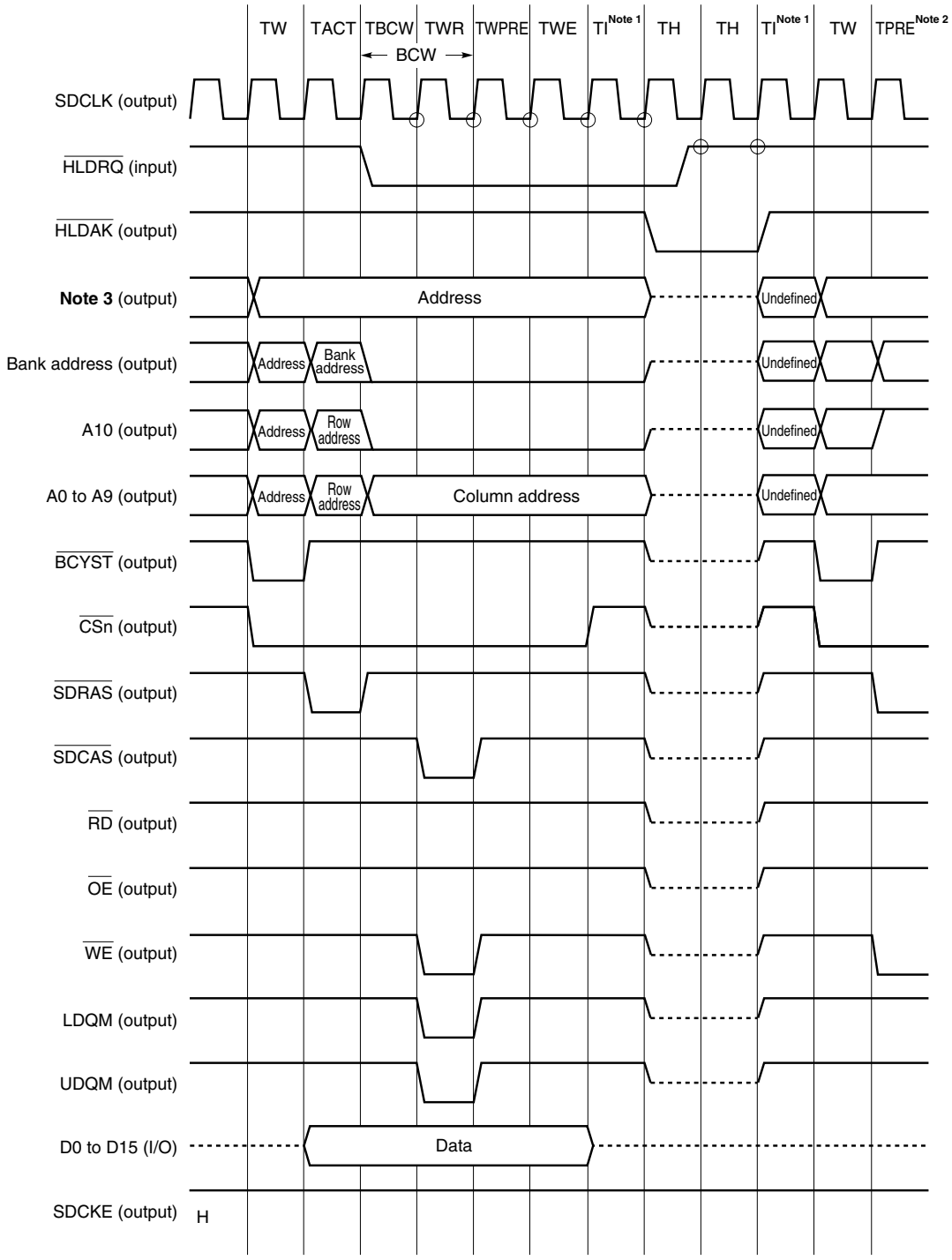
4. Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

(3) SDRAM (when written)



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. The all bank precharge command is always executed.

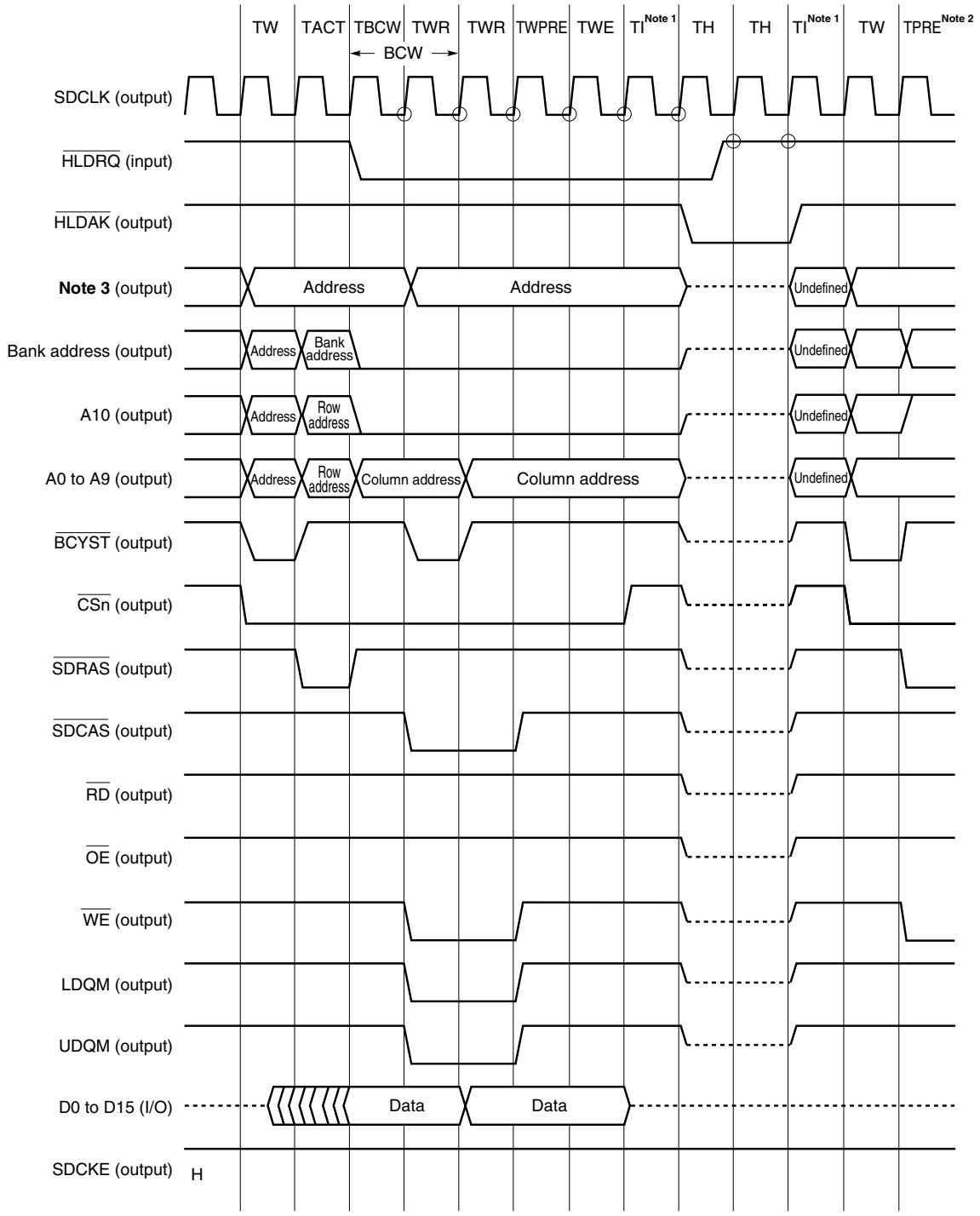
3. Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

(4) SDRAM (when written, when bus hold request acknowledged during on-page access)



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. The all bank precharge command is always executed.

3. Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

4.9 Bus Priority Order

There are five external bus cycles: bus hold, instruction fetch, operand data access, DMA cycle, and refresh cycle.

★ In order of priority, bus hold is the highest, followed by the refresh cycle, DMA cycle, operand data access, and instruction fetch, in that order.

An instruction fetch may be inserted between a read access and write access during a read modify write access.

Also, an instruction fetch may be inserted between bus accesses when the CPU bus clock is used.

Table 4-2. Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
High	Bus hold	External device
↕	Refresh cycle	DRAM controller
↕	DMA cycle	DMA controller
↕	Operand data access	CPU
Low	Instruction fetch	CPU

4.10 Boundary Operation Conditions

4.10.1 Program space

(1) Branching to the on-chip peripheral I/O area or successive fetches from the internal RAM area to the on-chip peripheral I/O area are prohibited. If the above is performed (branching or successive fetch), undefined data is fetched, and fetching from the external memory is not performed.

(2) If a branch instruction exists at the upper limit of the internal RAM area, a prefetch operation (invalid fetch) that straddles over the on-chip peripheral I/O area does not occur.

(3) If a burst fetch is performed for contiguous memory blocks, it is terminated at the upper limit of a block, and the startup cycle is started at the lower limit of the next block.

(4) Burst fetch is valid only in the external memory area. In memory block 7, it is terminated when the internal address count value has reached the upper limit of the external memory area.

4.10.2 Data space

The V850E/MA1 is provided with an address misalign function.

Through this function, regardless of the data format (word or halfword), data can be allocated to all addresses. However, in the case of word data and halfword data, if the data is not subject to boundary alignment, the bus cycle will be generated at least 2 times and bus efficiency will drop.

(1) In the case of halfword-length data access

When the address's LSB is 1, a byte-length bus cycle will be generated 2 times.

(2) In the case of word-length data access

(a) When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.

(b) When the address's lower 2 bits are 10, a halfword-length bus cycle will be generated 2 times.

CHAPTER 5 MEMORY ACCESS CONTROL FUNCTION

5.1 SRAM, External ROM, External I/O Interface

5.1.1 Features

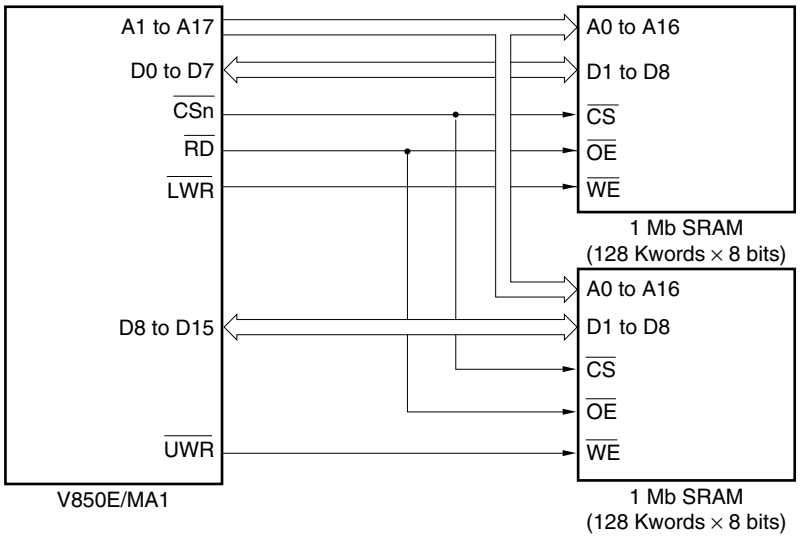
- SRAM is accessed in a minimum of 2 states.
- Up to 7 states of programmable data waits can be inserted by setting the DWC0 and DWC1 registers.
- Data wait can be controlled via $\overline{\text{WAIT}}$ pin input.
- Up to 3 idle states can be inserted after a read/write cycle by setting the BCC register.
- Up to 3 address setup wait states can be inserted by setting the ASC register.
- DMA flyby transfer can be activated (SRAM → external I/O, external I/O → SRAM)

5.1.2 SRAM connection

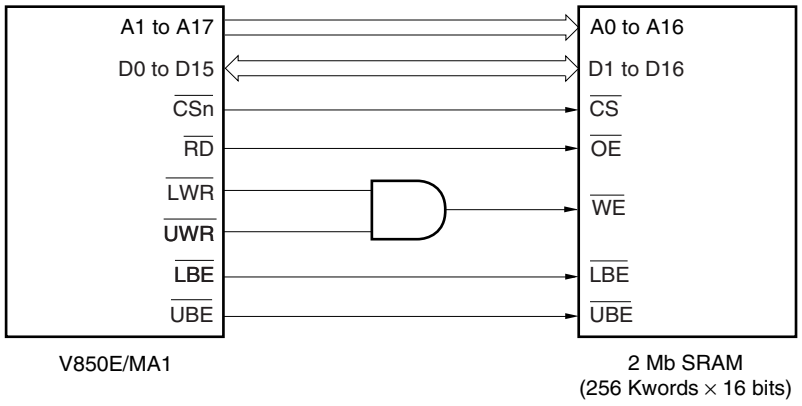
Examples of connection to SRAM are shown below.

Figure 5-1. Examples of Connection to SRAM (1/2)

(a) When data bus width is 8 bits



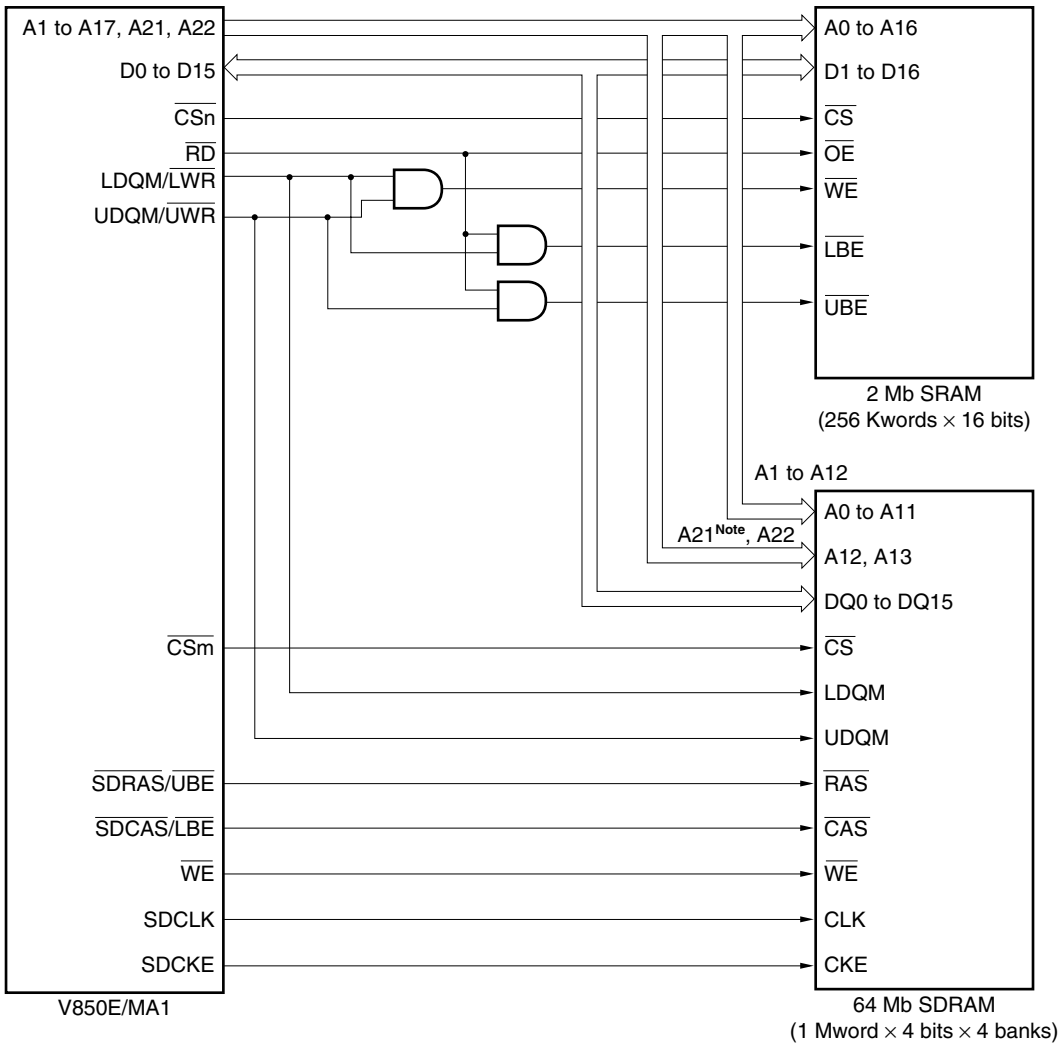
(b) When data bus width is 16 bits



Remark n = 0 to 7

Figure 5-1. Examples of Connection to SRAM (2/2)

(c) Mixture of SRAM (256 Kwords × 16 bits) and SDRAM (1 Mword × 16 bits)



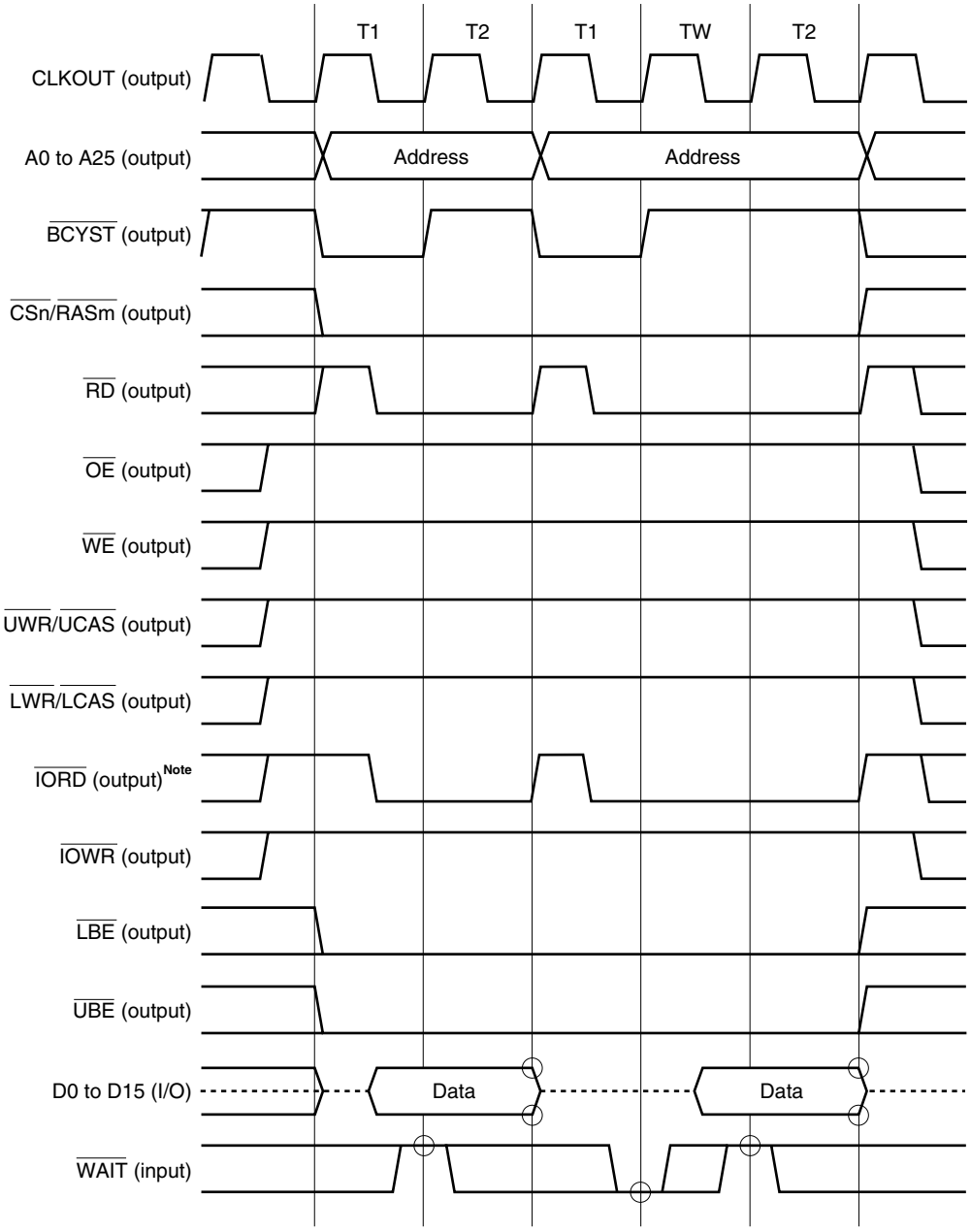
Note The address signals used depend on the SDRAM model.

Remark n = 0 to 7, m = 1, 3, 4, 6 (n ≠ m)

5.1.3 SRAM, external ROM, external I/O access

Figure 5-2. SRAM, External ROM, External I/O Access Timing (1/6)

(a) When read



Note When the IOEN bit of the BCP register is set to 1.

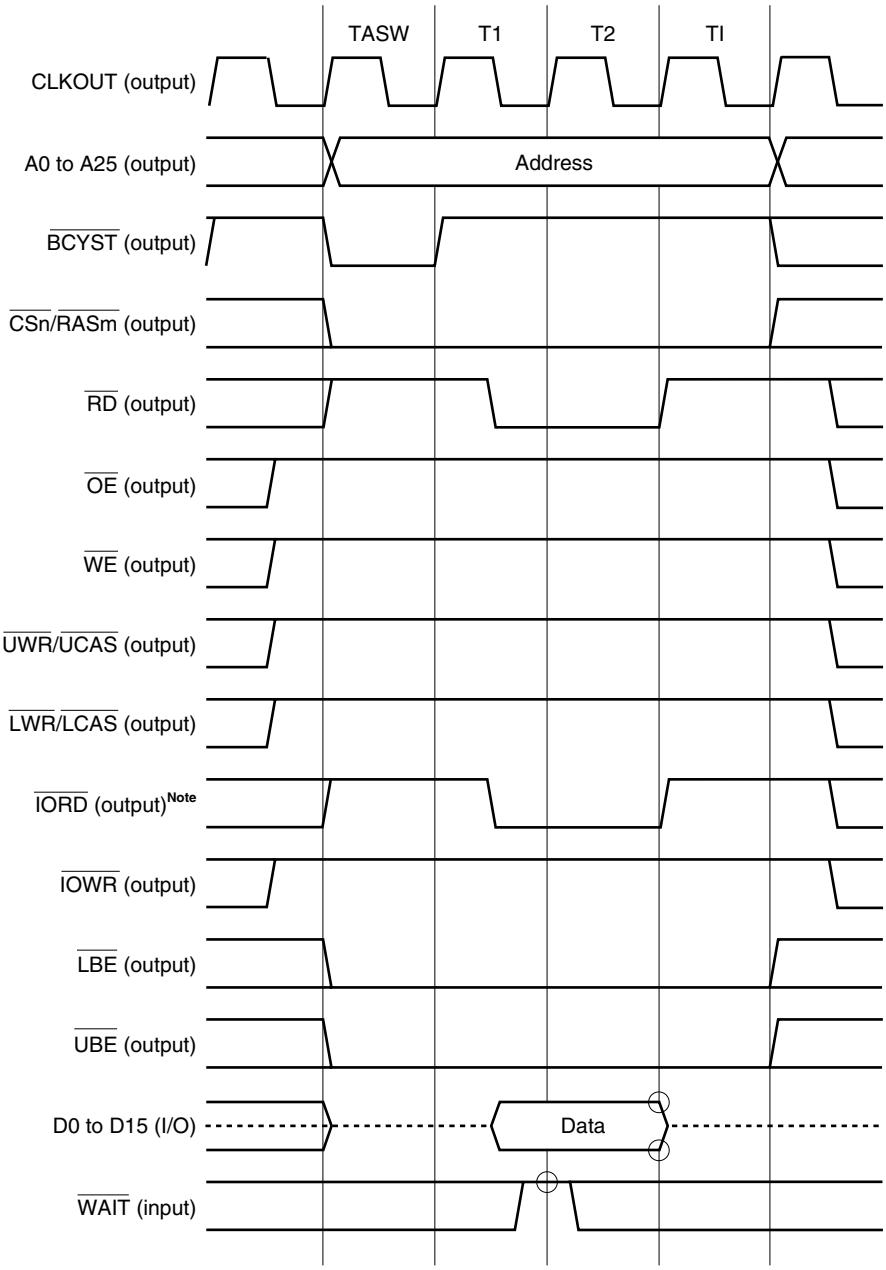
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-2. SRAM, External ROM, External I/O Access Timing (2/6)

(b) When read (address setup wait, idle state insertion)



Note When the IOEN bit of the BCP register is set to 1.

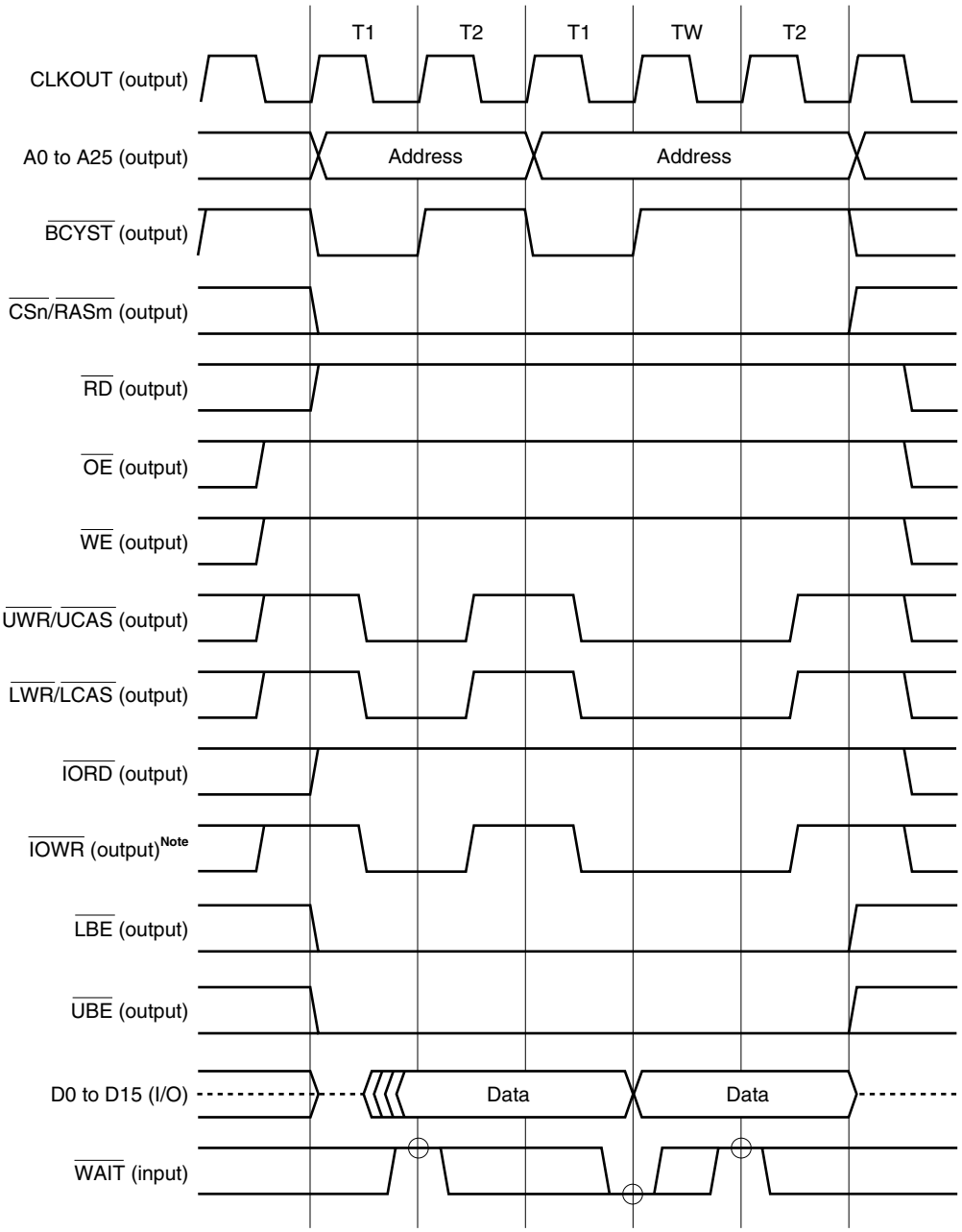
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-2. SRAM, External ROM, External I/O Access Timing (3/6)

(c) When written



Note When the IOEN bit of the BCP register is set to 1.

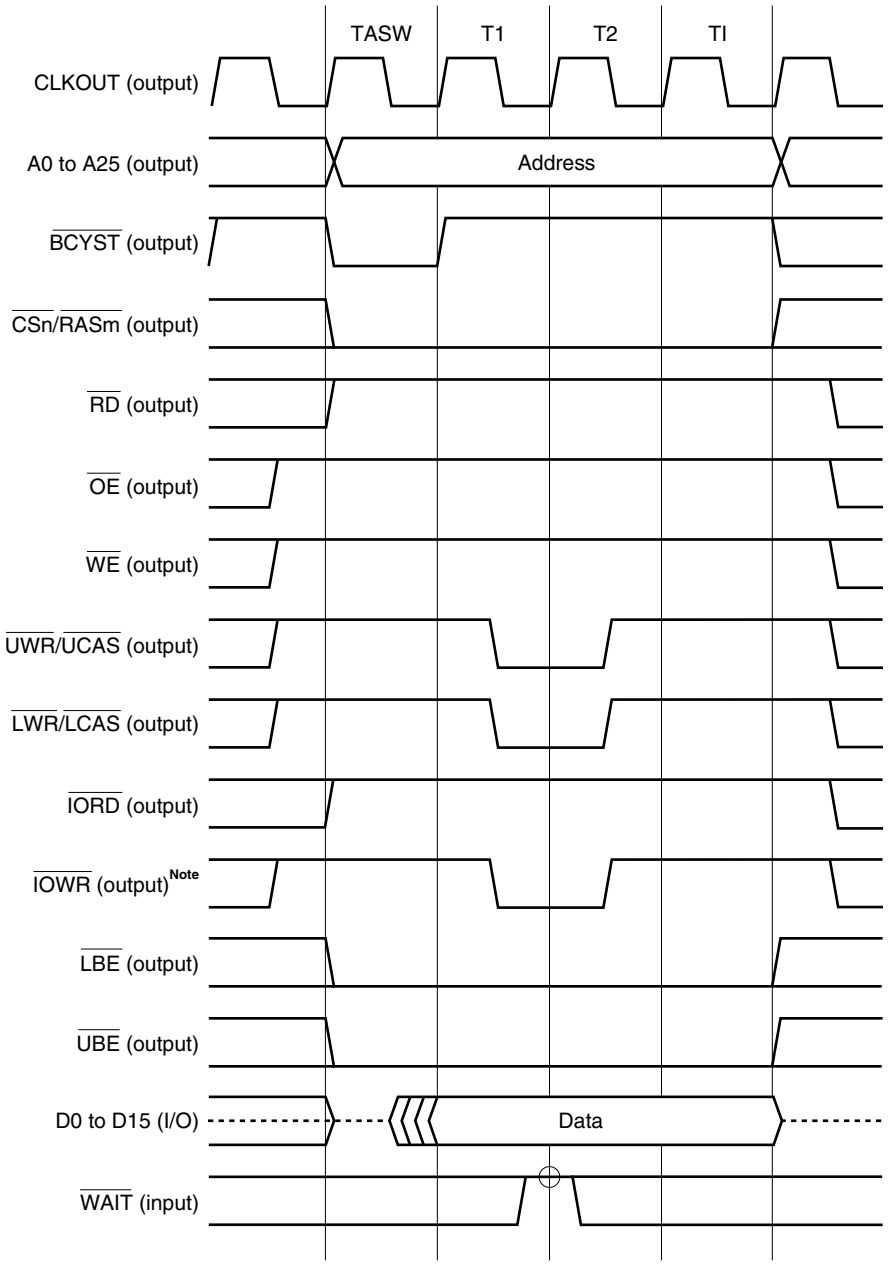
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-2. SRAM, External ROM, External I/O Access Timing (4/6)

(d) When written (address setup wait, idle state insertion)



Note When the IOEN bit of the BCP register is set to 1.

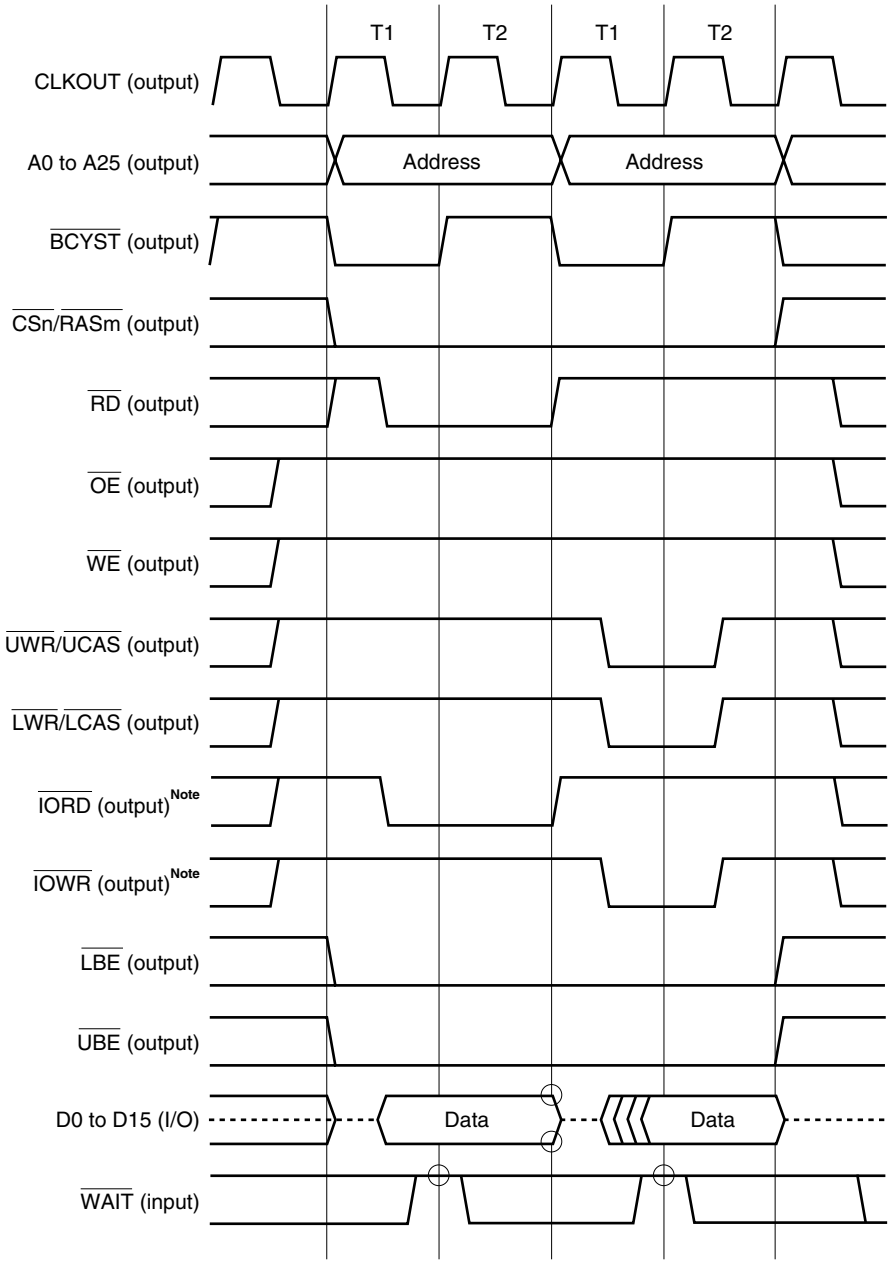
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-2. SRAM, External ROM, External I/O Access Timing (5/6)

(e) For read → write operation



Note When the IOEN bit of the BCP register is set to 1.

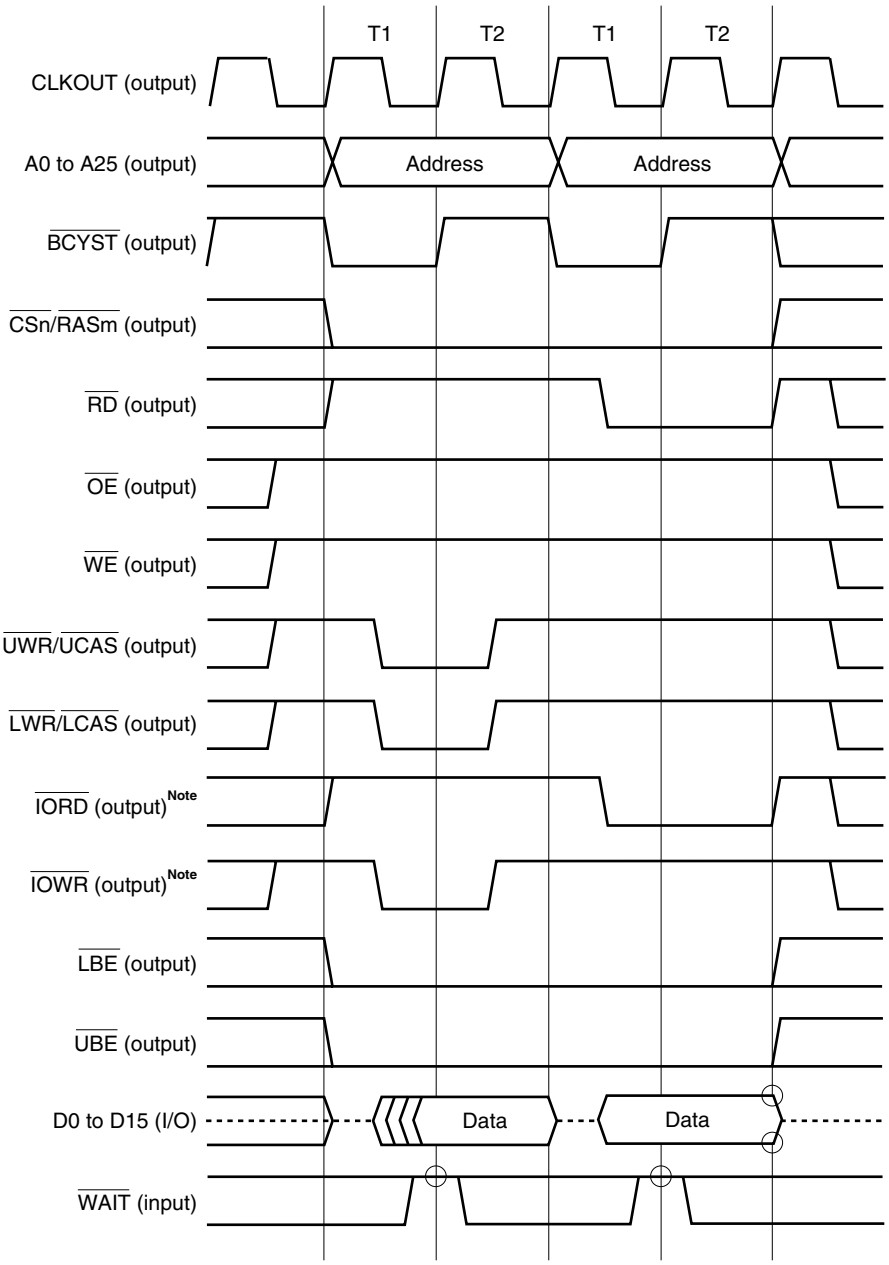
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-2. SRAM, External ROM, External I/O Access Timing (6/6)

(f) For write → read operation



Note When the IOEN bit of the BCP register is set to 1.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

5.2 Page ROM Controller (ROMC)

The page ROM controller (ROMC) is provided for accessing ROM (page ROM) with a page access function.

Addresses are compared with the immediately preceding bus cycle and wait control for normal access (off-page) and page access (on-page) is executed. This controller can handle page widths from 8 to 128 bytes.

5.2.1 Features

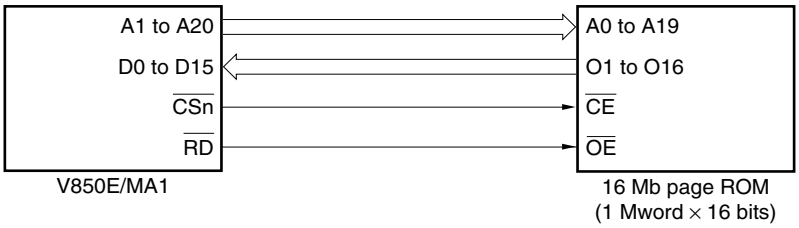
- Direct connection to 8-bit/16-bit page ROM supported
- For 16-bit bus width: 4/8/16/32/64-word page access supported
  For 8-bit bus width: 8/16/32/64/128-word page access supported
- Page ROM is accessed in a minimum of 2 states.
- On-page judgment function
- Addresses to be compared can be changed by setting the PRC register.
- Up to 7 states of programmable data waits can be inserted during an on-page cycle by setting the PRC register.
- Up to 7 states of programmable data waits can be inserted during an off-page cycle by setting the DWC0 and DWC1 registers.
- Waits can be controlled via $\overline{\text{WAIT}}$ pin input.
- DMA flyby cycle can be activated (page ROM → external I/O)

5.2.2 Page ROM connection

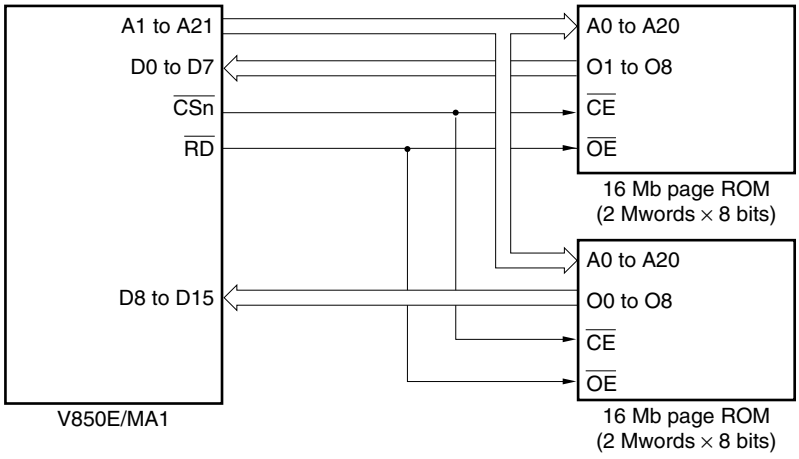
Examples of connection to page ROM are shown below.

Figure 5-3. Examples of Connection to Page ROM

(a) When data bus width is 16 bits



(b) When data bus width is 8 bits



Remark n = 0 to 7

5.2.3 On-page/off-page judgment

Whether a page ROM cycle is on-page or off-page is judged by latching the address of the previous cycle and comparing it with the address of the current cycle.

Through the page ROM configuration register (PRC), according to the configuration of the connected page ROM and the number of continuously readable bits, one of the addresses (A3 to A6) is set as the masking address (no comparison is made).

Figure 5-4. On-Page/Off-Page Judgment During Page ROM Connection (1/2)

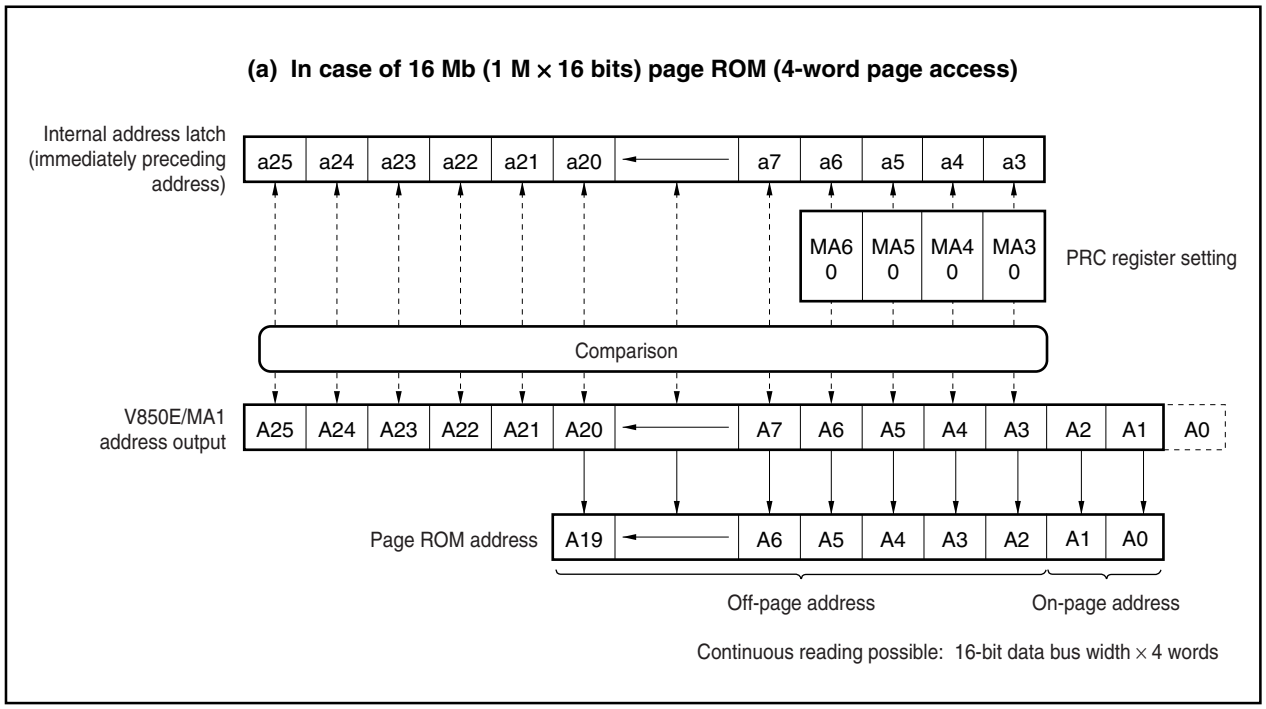
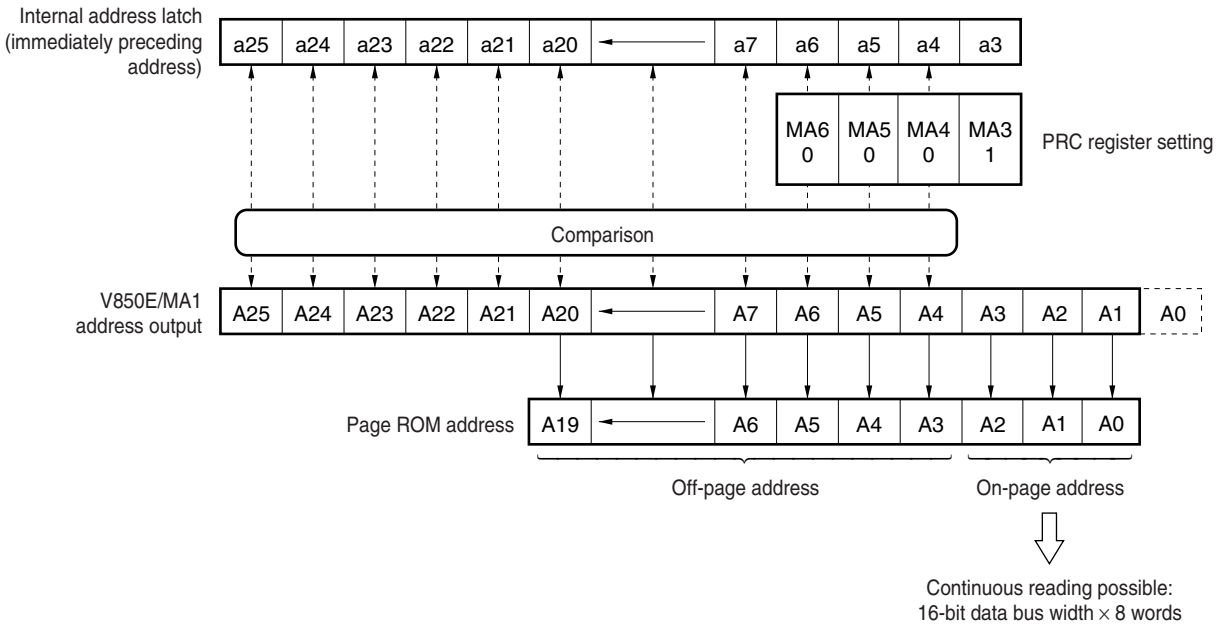
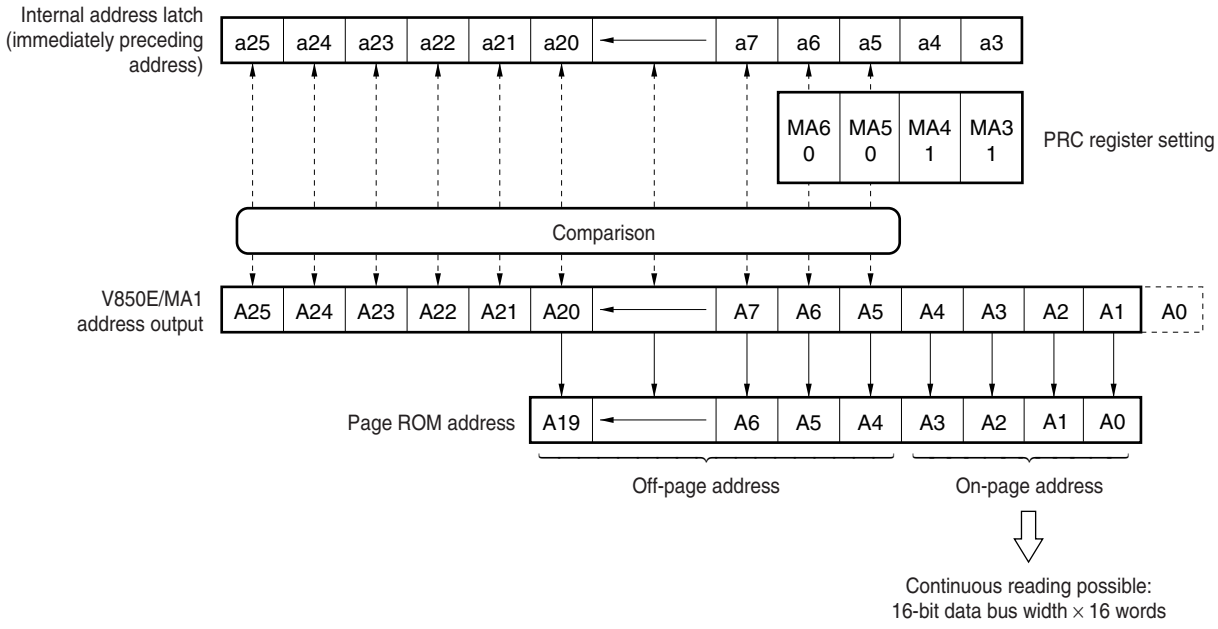


Figure 5-4. On-Page/Off-Page Judgment During Page ROM Connection (2/2)

(b) In case of 16 Mb (1 M × 16 bits) page ROM (8-word page access)



(c) In case of 32 Mb (2 M × 16 bits) page ROM (16-word page access)



5.2.4 Page ROM configuration register (PRC)

This register specifies whether page ROM cycle on-page access is enabled or disabled. If on-page access is enabled, the masking address (no comparison is made) out of the addresses (A3 to A6) corresponding to the configuration of the connected page ROM and the number of bits that can be read continuously, as well as the number of waits corresponding to the internal system clock, are set.

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

Caution Write to the PRC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the PRC register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
PRC	0	PRW2	PRW1	PRW0	0	0	0	0	0	0	0	0	MA6	MA5	MA4	MA3	FFFFF49AH	7000H

Bit position	Bit name	Function
14 to 12	PRW2 to PRW0	Page-ROM On-page Wait Control Sets the number of waits corresponding to the internal system clock. The number of waits set by these bits is inserted only for on-page access. For off-page access, the waits set by registers DWC0 and DWC1 are inserted.
3 to 0	MA6 to MA3	Mask Address Each respective address (A6 to A3) corresponding to MA6 to MA3 is masked (by 1). The masked address is not subject to comparison during on/off-page judgment, and is set according to the number of continuously readable bits.

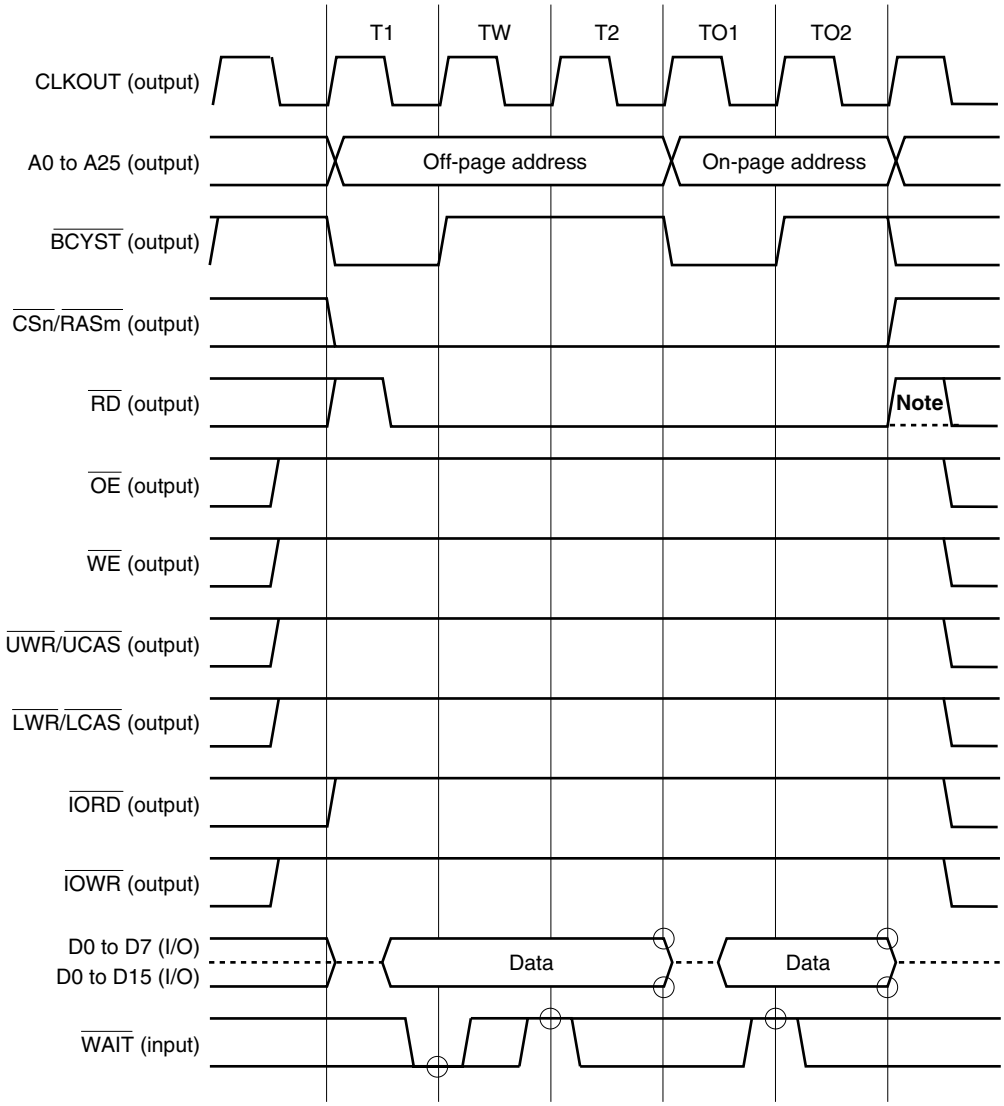
PRW2	PRW1	PRW0	Number of inserted wait cycles
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

MA6	MA5	MA4	MA3	Number of continuously readable bits
0	0	0	0	4 words × 16 bits (8 words × 8 bits)
0	0	0	1	8 words × 16 bits (16 words × 8 bits)
0	0	1	1	16 words × 16 bits (32 words × 8 bits)
0	1	1	1	32 words × 16 bits (64 words × 8 bits)
1	1	1	1	64 words × 16 bits (128 words × 8 bits)
Other than above				Setting prohibited

5.2.5 Page ROM access

Figure 5-5. Page ROM Access Timing (1/4)

(a) When read (halfword/word access with 8-bit bus width or word access with 16-bit bus width)



★ **Note** When accessing a word boundary with 8-bit bus width.

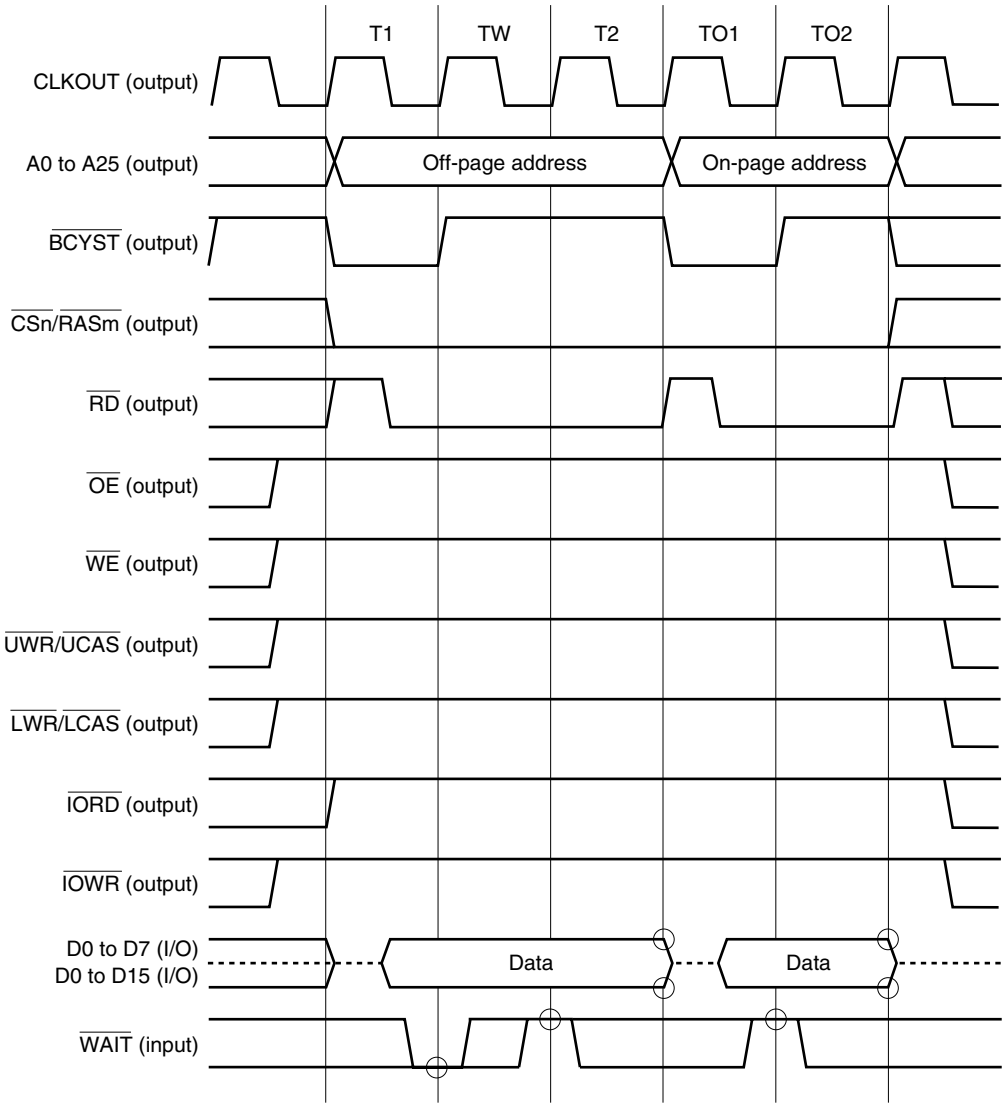
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-5. Page ROM Access Timing (2/4)

(b) When read (byte access with 8-bit bus width or byte/half-word access with 16-bit bus width)



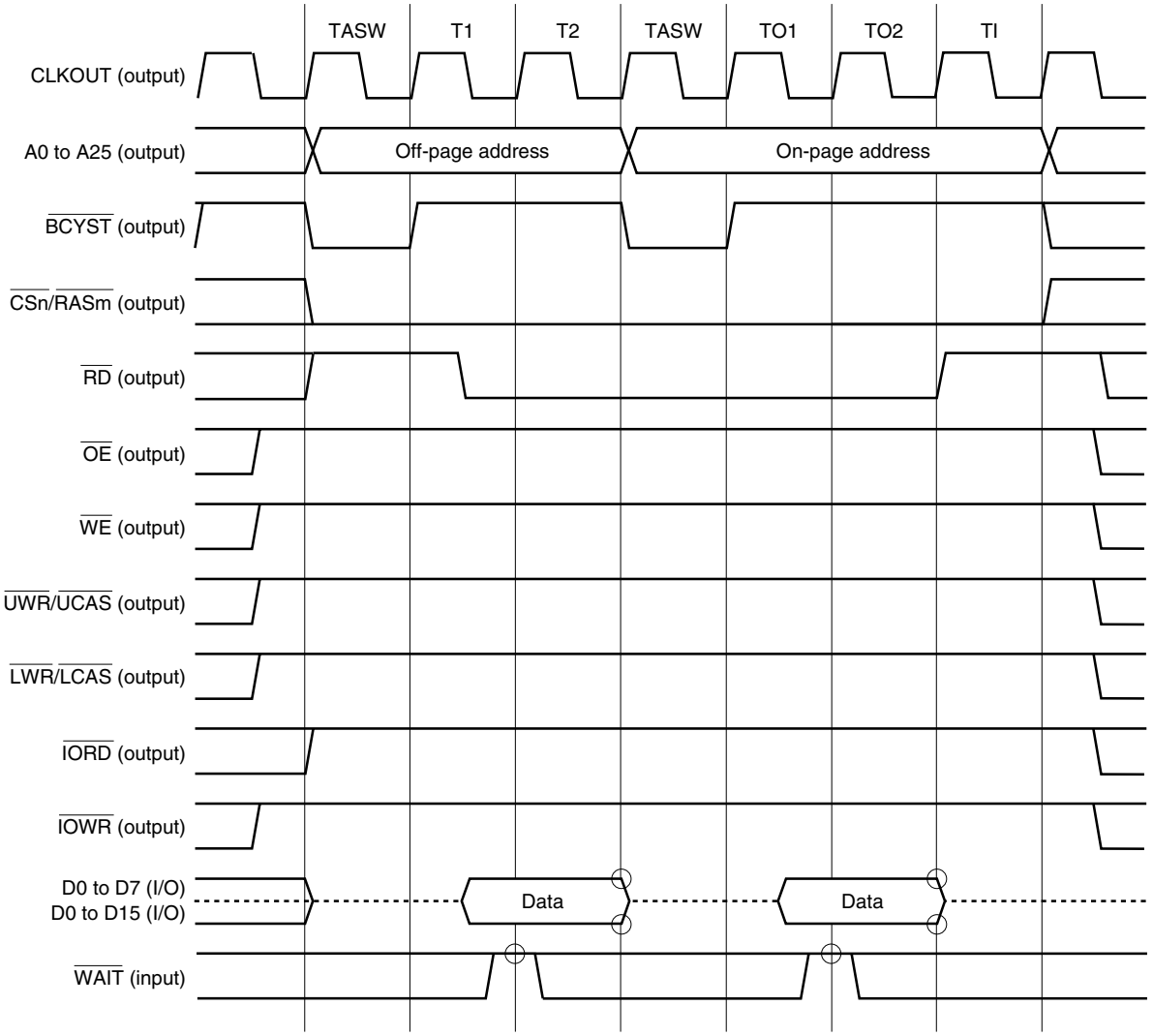
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-5. Page ROM Access Timing (3/4)

(c) When read (address setup wait, idle state insertion) (halfword/word access with 8-bit bus width or word access with 16-bit bus width)



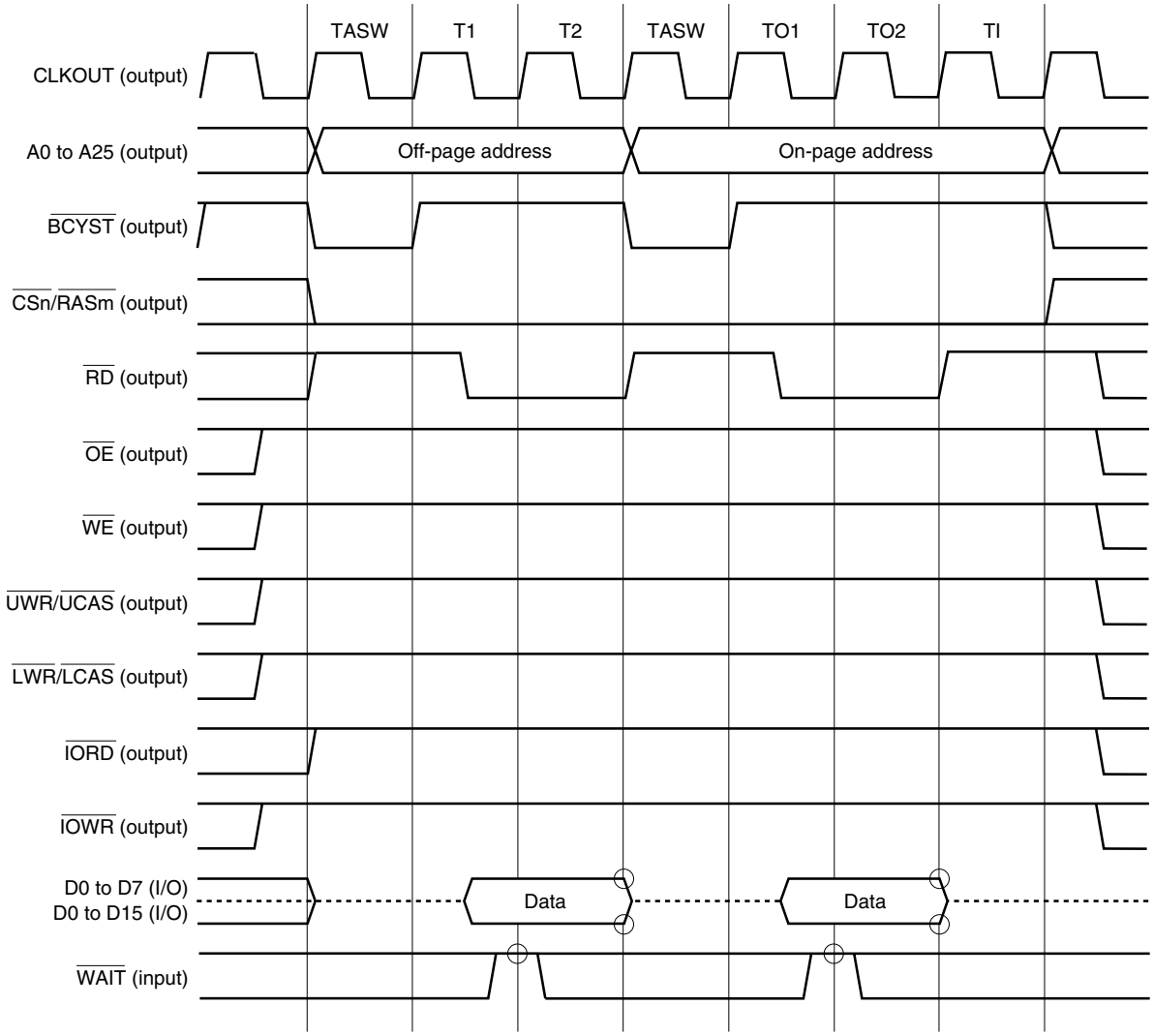
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-5. Page ROM Access Timing (4/4)

(d) When read (address setup wait, idle state insertion) (byte access with 8-bit bus width or byte/halfword access with 16-bit bus width)



Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

5.3 DRAM Controller (EDO DRAM)

5.3.1 Features

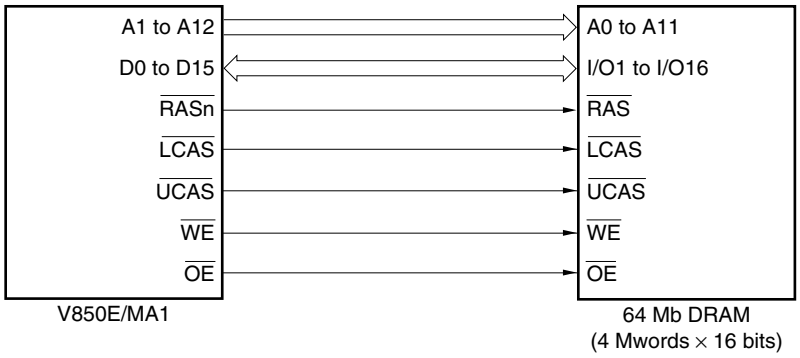
- Generates the $\overline{RAS}$, $\overline{LCAS}$, and $\overline{UCAS}$ signals
- Can be connected directly to EDO DRAM.
- Supports the RAS hold mode.
- 4 types of DRAM can be assigned to 4 memory block spaces.
- Supports 2CAS type DRAM.
- Row and column address multiplex widths can be changed.
- Waits (0 to 3 waits) can be inserted at the following timings:
  - Row address precharge wait
  - Row address hold wait
  - Data access wait
  - Column address precharge wait
- Supports CBR refresh and CBR self-refresh.

5.3.2 DRAM connection

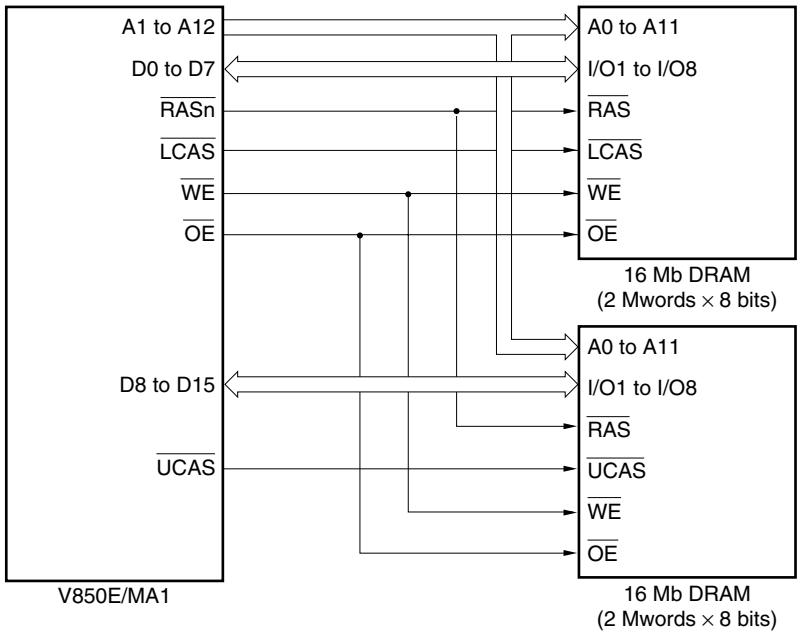
Examples of connection to DRAM are shown below.

Figure 5-6. Examples of Connection to DRAM

(a) When DRAM is 64 Mb (4 M × 16 bits)



(b) When DRAM is 16 Mb (2 M × 8 bits)



Remark n = 1, 3, 4, 6

5.3.3 Address multiplex function

Depending on the value of the DAW0n and DAW1n bits in DRAM configuration register n (SCRn), the row address and column address outputs in the DRAM cycle are multiplexed as shown in Figure 5-7 (n = 1, 3, 4, 6). In Figure 5-7, a0 to a25 show the addresses output from the CPU and A0 to A25 show the address pins of the V850E/MA1.

For example, when DAW1n and DAW0n = 11, it indicates that a12 to a22 are output as row addresses and a1 to a11 are output as column addresses from the address pins (A1 to A11).

Figure 5-7. Row Address/Column Address Output

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address (DAW1n, DAW0n = 11)	a25 to a18	a17	a16	a15	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11
Row address (DAW1n, DAW0n = 10)	a25 to a18	a17	a16	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
Row address (DAW1n, DAW0n = 01)	a25 to a18	a17	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
Row address (DAW1n, DAW0n = 00)	a25 to a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8
Column address	a25 to a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

Remark n = 1, 3, 4, 6

Table 5-1 shows the relationship between the DRAM that can be connected and the address multiplex width. The DRAM space differs according to the DRAM that is connected, as shown in Table 5-1.

Table 5-1. Example of DRAM and Address Multiplex Width

Address Multiplex Width	DRAM Capacity (Bits) and Configuration					DRAM Space[Note] (Bytes)
	256 K	1 M	4 M	16 M	64 M	
8 bits (DAW1n, DAW0n = 00)	64 K × 4	–	–	–	–	128 K
9 bits (DAW1n, DAW0n = 01)	–	256 K × 4	256 K × 16	–	–	512 K
	–	–	512 K × 8	–	–	1 M
	–	–	–	–	4 M × 16	8 M
10 bits (DAW1n, DAW0n = 10)	–	–	1 M × 4	1 M × 16	–	2 M
	–	–	–	2 M × 8	–	4 M
	–	–	–	–	4 M × 16	8 M
11 bits (DAW1n, DAW0n = 11)	–	–	–	4 M × 4	–	8 M

Note When the data bus width is 16 bits

Remark n = 1, 3, 4, 6

5.3.4 DRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6)

These registers are used to set the type of DRAM to be connected. SCRn corresponds to $\overline{CSn}$ (n = 1, 3, 4, 6). For example, to connect DRAM to $\overline{CS1}$, set SCR1. These registers can be read/written in 16-bit units.

Cautions 1. If the object of access is a DRAM area, the wait set by registers DWC0 and DWC1 becomes invalid. In this case, waits are controlled by registers SCR1, SCR3, SCR4, and SCR6.

2. If bit 14 is set to 1, the operation is not guaranteed.

3. Write to the SCR1, SCR3, SCR4, and SCR6 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the SCR1, SCR3, SCR4, and SCR6 registers are complete. However, it is possible to access external memory areas whose initialization settings are complete.

(1/3)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SCR1	PAE11	0	RPC11	RPC01	RHC11	RHC01	DAC11	DAC01	CPC11	CPC01	0	RHD1	ASO11	ASO01	DAW11	DAW01	FFFFF4A4H	3FC1H
SCR3	PAE13	0	RPC13	RPC03	RHC13	RHC03	DAC13	DAC03	CPC13	CPC03	0	RHD3	ASO13	ASO03	DAW13	DAW03	FFFFF4ACH	3FC1H
SCR4	PAE14	0	RPC14	RPC04	RHC14	RHC04	DAC14	DAC04	CPC14	CPC04	0	RHD4	ASO14	ASO04	DAW14	DAW04	FFFFF4B0H	3FC1H
SCR6	PAE16	0	RPC16	RPC06	RHC16	RHC06	DAC16	DAC06	CPC16	CPC06	0	RHD6	ASO16	ASO06	DAW16	DAW06	FFFFF4B8H	3FC1H

Bit position	Bit name	Function
15	PAE1n (n = 1, 3, 4, 6)	DRAM On-page Access Mode Control Sets the on-page access cycle. PAE1n: 0 — On-page access disabled PAE1n: 1 — On-page access enabled
13, 12	RPC1n, RPC0n (n = 1, 3, 4, 6)	Row Address Pre-charge Control Specifies the number of wait states inserted as row address precharge time. RPC1n = 0, RPC0n = 0: 1 (at least 1 wait is always inserted.) RPC1n = 0, RPC0n = 1: 1 RPC1n = 1, RPC0n = 0: 2 RPC1n = 1, RPC0n = 1: 3

(2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11, 10</td><td>RHC1n,
RHC0n
(n = 1, 3,
4, 6)</td><td>Row Address Hold Wait Control
Specifies the number of wait states inserted as row address hold time.
<table><tr><th>RHC1n</th><th>RHC0n</th><th>Number of wait states inserted</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>2</td></tr><tr><td>1</td><td>1</td><td>3</td></tr></table></td></tr>
<tr><td>9, 8</td><td>DAC1n,
DAC0n
(n = 1, 3,
4, 6)</td><td>Data Access Programmable Wait Control
Specifies the number of wait states inserted as data access time during DRAM access.
<table><tr><th>DAC1n</th><th>DAC0n</th><th>Number of wait states inserted</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>2</td></tr><tr><td>1</td><td>1</td><td>3</td></tr></table></td></tr>
<tr><td>7, 6</td><td>CPC1n,
CPC0n
(n = 1, 3,
4, 6)</td><td>Column Address Pre-charge Control
Specifies the number of wait states inserted as column address precharge time.
<table><tr><th>CPC1n</th><th>CPC0n</th><th>Number of wait states inserted</th></tr><tr><td>0</td><td>0</td><td>0 (at least 1 wait is always inserted during on-page write access)</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>2</td></tr><tr><td>1</td><td>1</td><td>3</td></tr></table></td></tr>
<tr><td>4</td><td>RHDn
(n = 1, 3,
4, 6)</td><td>RAS Hold Disable
Sets the RAS hold mode.
If access to DRAM during on-page operation is not continuous and another space is accessed midway, the $\overline{\text{RAS}}$n signal is maintained in the active state (low level) during the time the other space is being accessed in the RAS hold mode. In this way, if access continues in the same DRAM row address following access of the other space, on-page operation can be continued.
0: RAS hold mode enabled
1: RAS hold mode disabled</td></tr>
</table>

(3/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>ASO1n,
ASO0n
(n = 1, 3,
4, 6</td><td>Address Shift Width On-page Control
This sets the address shift width during on-page judgment.
When the external data bus width is 8 bits: Set ASO1n, ASO0n = 00B
When the external data bus width is 16 bits: Set ASO1n, ASO0n = 01B

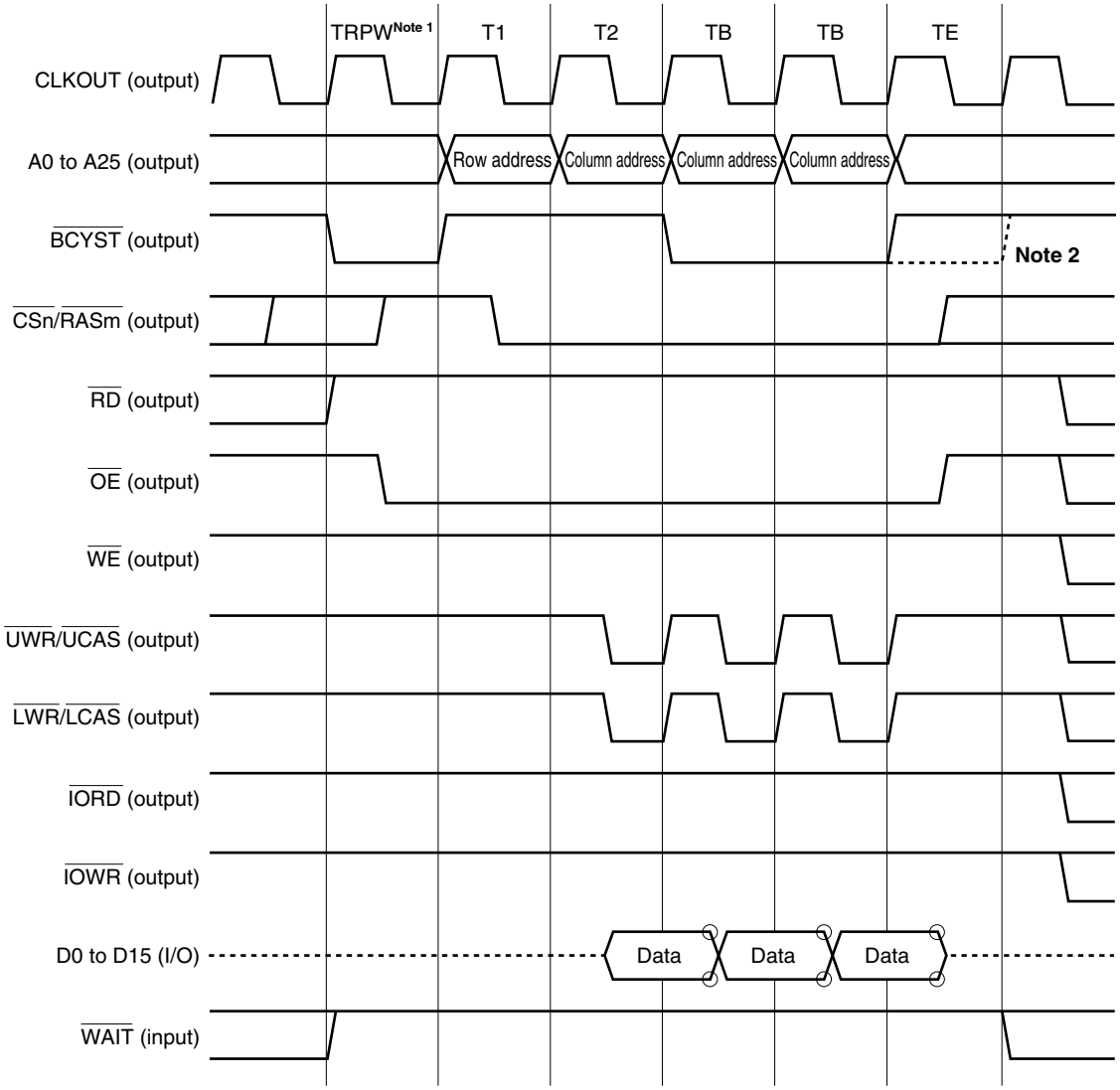
<table><tr><th>ASO1n</th><th>ASO0n</th><th>Address shift width</th></tr><tr><td>0</td><td>0</td><td>0 (data bus width: 8 bits)</td></tr><tr><td>0</td><td>1</td><td>1 (data bus width: 16 bits)</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited</td></tr></table></td></tr>
<tr><td>1, 0</td><td>DAW1n,
DAW0n
(n = 1, 3,
4, 6)</td><td>DRAM Address Multiplex Width Control
This sets the address multiplex width (refer to 5.3.3 Address multiplex function).

<table><tr><th>DAW1n</th><th>DAW0n</th><th>Address multiplex width</th></tr><tr><td>0</td><td>0</td><td>8 bits</td></tr><tr><td>0</td><td>1</td><td>9 bits</td></tr><tr><td>1</td><td>0</td><td>10 bits</td></tr><tr><td>1</td><td>1</td><td>11 bits</td></tr></table></td></tr>
</table>

5.3.5 DRAM access

Figure 5-8. EDO DRAM Access Timing (1/5)

(a) Read timing (when no waits are inserted)



Notes 1. TRPW is always inserted for 1 or more cycles.

2. When a bus cycle accessing another CS space or a write cycle accessing the same CS space follows this read cycle.

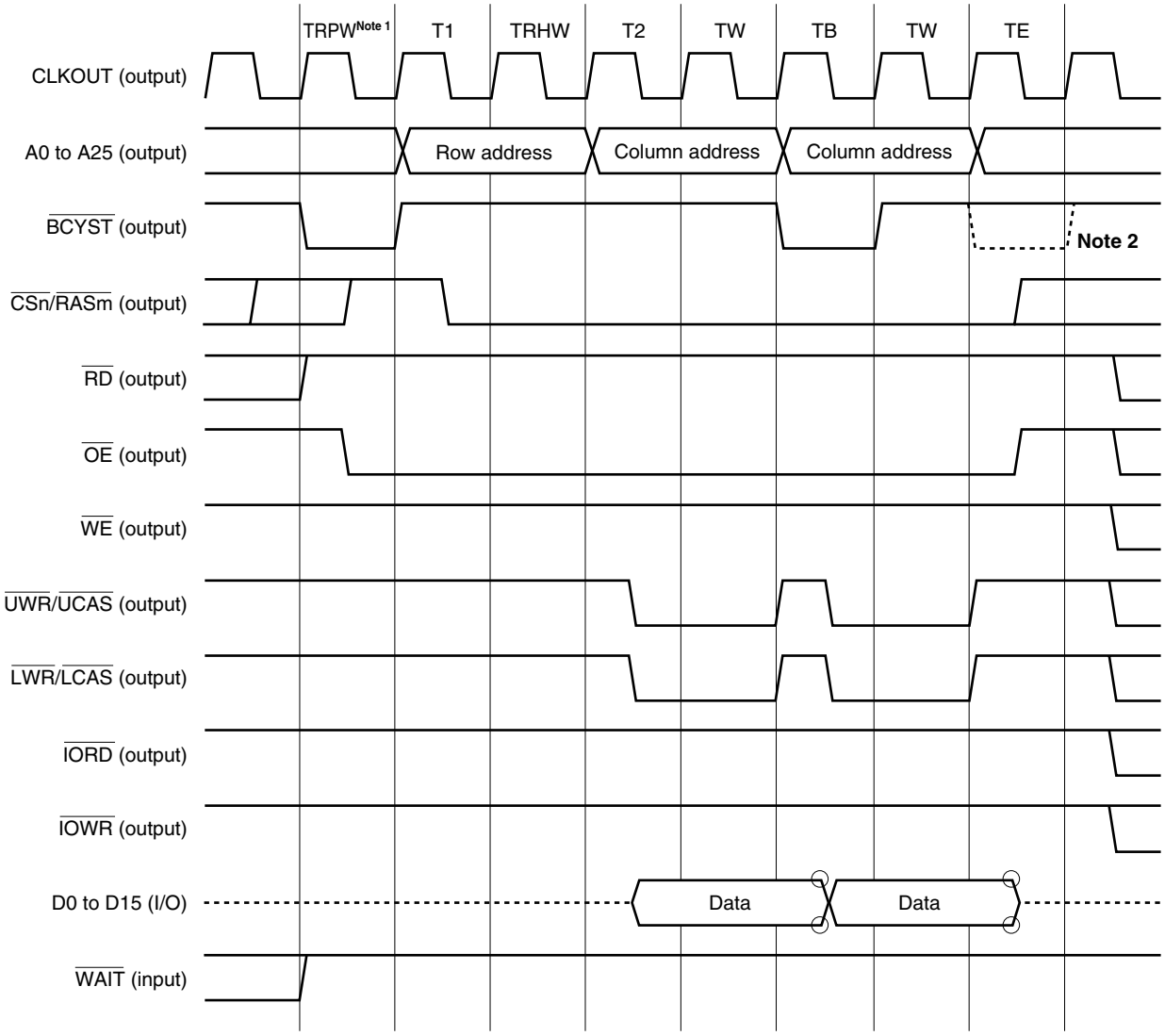
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-8. EDO DRAM Access Timing (2/5)

(b) Read timing (when TRHW and TW are inserted)



Notes 1. TRPW is always inserted for 1 or more cycles.

2. When a bus cycle accessing another CS space or a write cycle accessing the same CS space follows this read cycle.

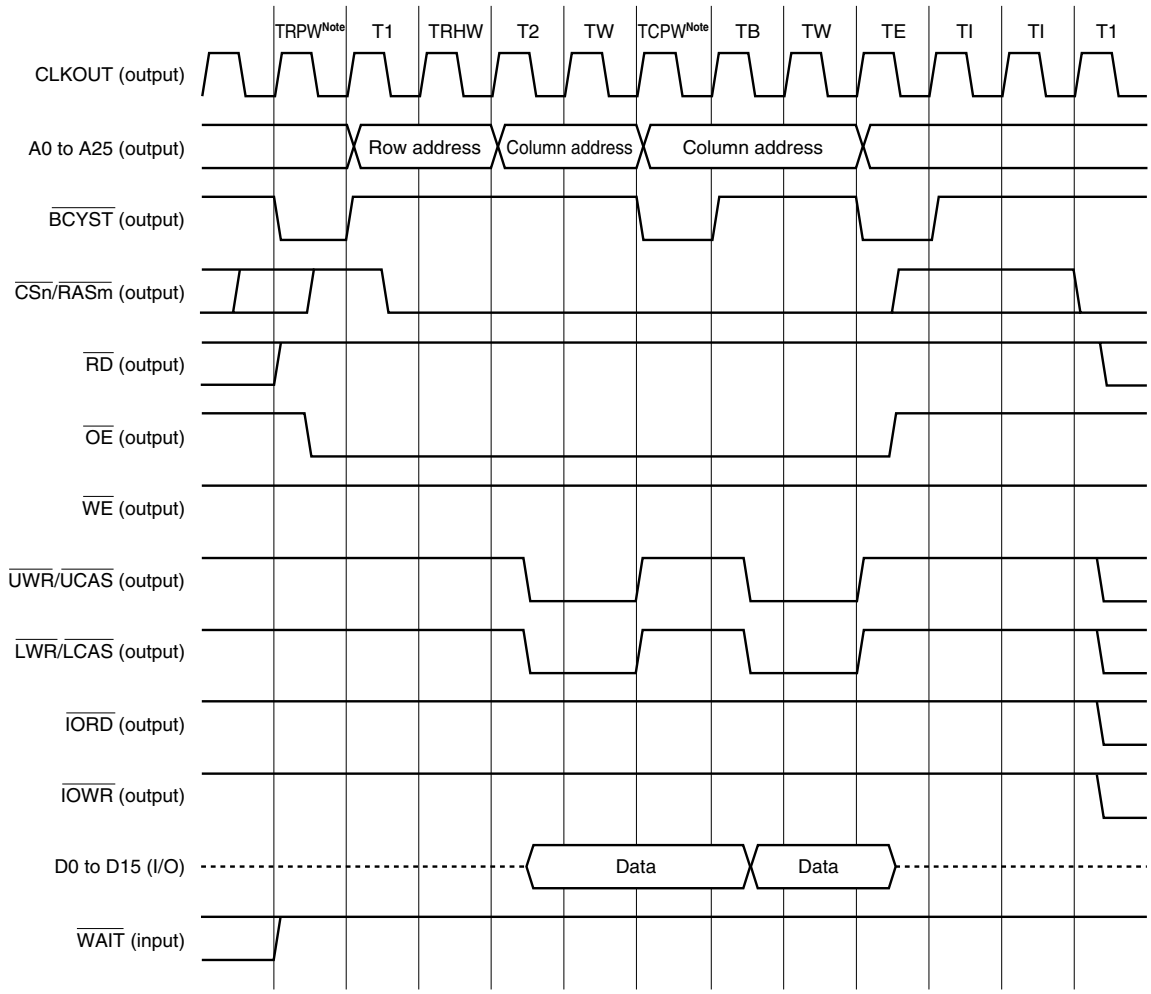
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-8. EDO DRAM Access Timing (3/5)

(c) Read timing (when two idle states are inserted)



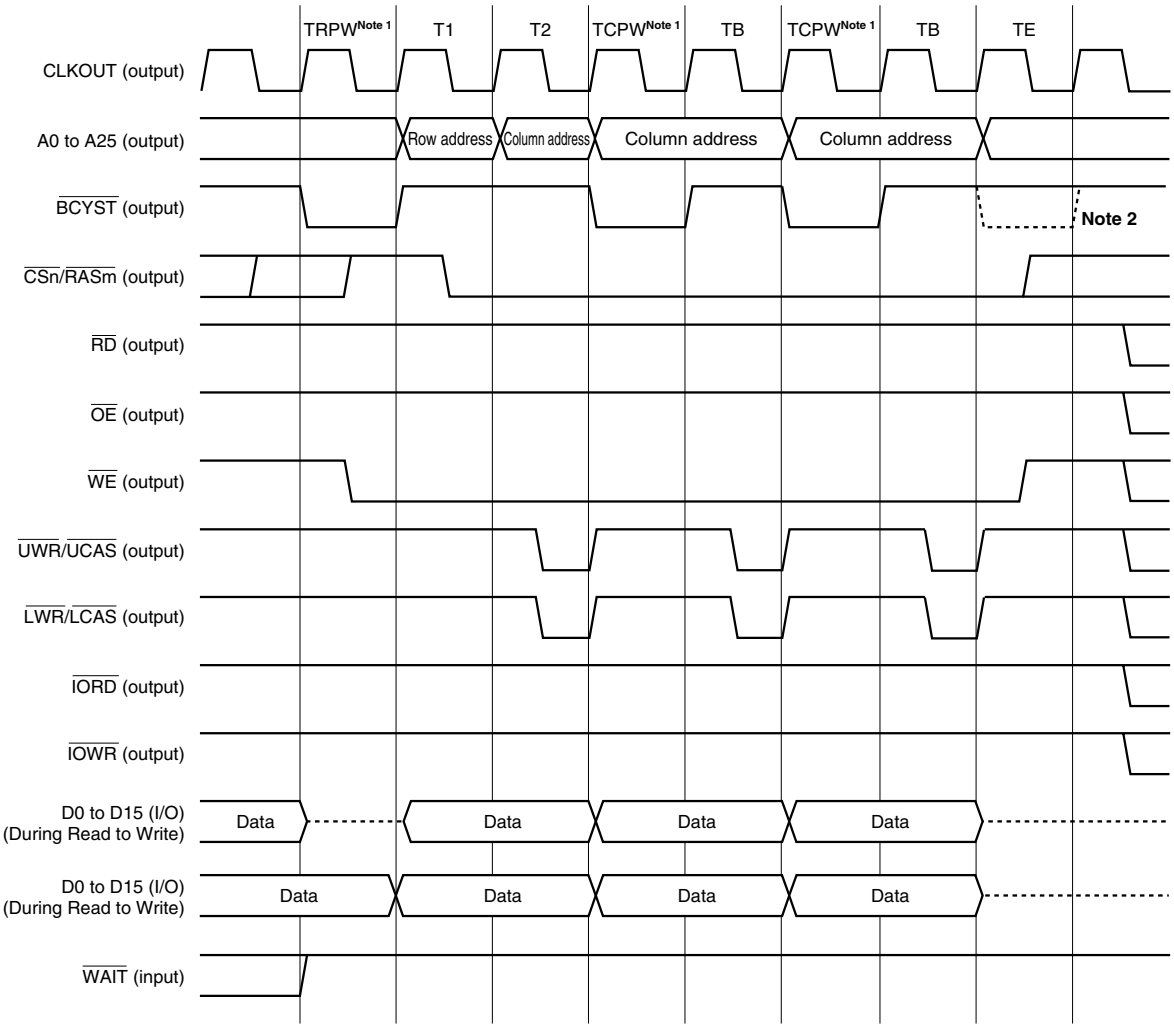
Note TRPW and TCPW are always inserted for 1 or more cycles.

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-8. EDO DRAM Access Timing (4/5)

(d) Write timing (when no waits are inserted)



Notes 1. TRPW and TCPW are always inserted for 1 or more cycles.

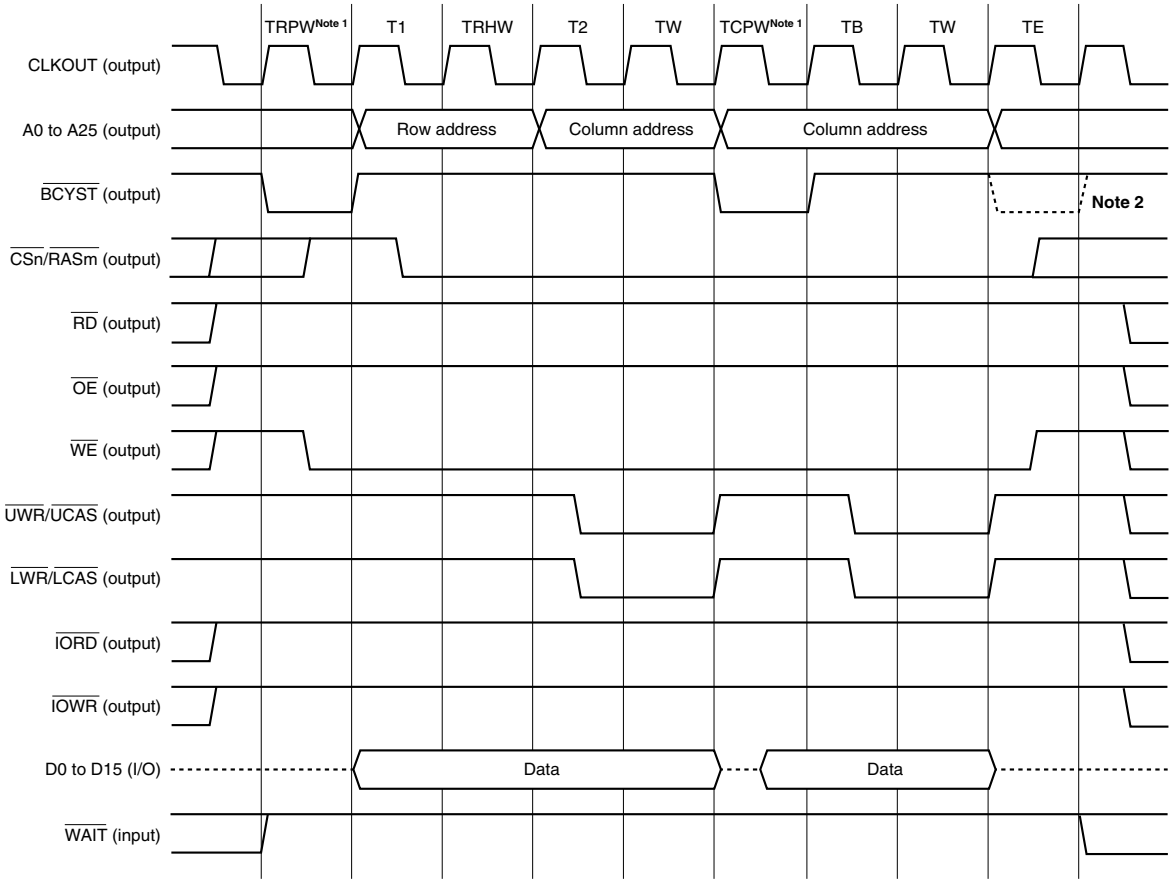
2. When a bus cycle accessing another CS space or a read cycle accessing the same CS space follows this write cycle.

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 1, 3, 4, 6

Figure 5-8. EDO DRAM Access Timing (5/5)

(e) Write timing (when TRHW and TW are inserted)



Notes 1. TRPW and TCPW are always inserted for 1 or more cycles.

2. When a bus cycle accessing another CS space or a read cycle accessing the same CS space follows this write cycle.

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 1, 3, 4, 6

5.3.6 Refresh control function

The V850E/MA1 can generate the CBR (CAS-before-RAS) refresh cycle. The refresh cycle is set with refresh control registers 1, 3, 4, and 6 (RFS1, RFS3, RFS4, RFS6). RFSn corresponds to $\overline{CSn}$ (n = 1, 3, 4, 6). For example, to connect DRAM to $\overline{CS1}$, set RFS1.

When another bus master occupies the external bus, the DRAM controller cannot occupy the external bus. In this case, the DRAM controller issues a refresh request to the bus master by changing the $\overline{REFRQ}$ signal to active (low level).

During a refresh operation, the address bus retains the state it was in just before the refresh cycle.

(1) Refresh control registers 1, 3, 4, 6 (RFS1, RFS3, RFS4, RFS6)

These registers are used to enable or disable a refresh and set the refresh interval. The refresh interval is determined by the following calculation formula.

Refresh interval (μs) = Refresh count clock (T_{RCY}) $\times$ Interval factor

The refresh count clock and interval factor are determined by the RENn bit and RIN5n to RIN0n bits, respectively, of the RFSn register.

Note that n corresponds to the register number (1, 3, 4, 6) of DRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6).

These registers can be read/written in 16-bit units.

Caution Write to the RFS1, RFS3, RFS4, and RFS6 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the RFS1, RFS3, RFS4, and RFS6 registers are complete. However, it is possible to access external memory areas whose initialization settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
RFS1	REN1	0	0	0	0	0	RCC11	RCC01	0	0	RIN51	RIN41	RIN31	RIN21	RIN11	RIN01	FFFFF4A6H	0000H
RFS3	REN3	0	0	0	0	0	RCC13	RCC03	0	0	RIN53	RIN43	RIN33	RIN23	RIN13	RIN03	FFFFF4AEH	0000H
RFS4	REN4	0	0	0	0	0	RCC14	RCC04	0	0	RIN54	RIN44	RIN34	RIN24	RIN14	RIN04	FFFFF4B2H	0000H
RFS6	REN6	0	0	0	0	0	RCC16	RCC06	0	0	RIN56	RIN46	RIN36	RIN26	RIN16	RIN06	FFFFF4BAH	0000H

Bit position	Bit name	Function
15	RENn (n = 1, 3, 4, 6)	Refresh Enable Specifies whether CBR refresh is enabled or disabled. 0: Refresh disabled 1: Refresh enabled
9, 8	RCC1n, RCC0n (n = 1, 3, 4, 6)	Refresh Count Clock Specifies the refresh count clock (T_{RCY})
5 to 0	RIN5n to RIN0n (n = 1, 3, 4, 6)	Refresh Interval Sets the interval factor of the interval timer for the generation of the refresh timing.

RCC1n	RCC0n	Refresh count clock (T_{RCY})
0	0	32/ϕ
0	1	128/ϕ
1	0	256/ϕ
1	1	Setting prohibited

RIN5n	RIN4n	RIN3n	RIN2n	RIN1n	RIN0n	Interval factor
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
1	1	1	1	1	1	64

Remark ϕ: Internal system clock frequency

Table 5-2. Interval Factor Setting Examples

Specified Refresh Interval Value (μs)	Refresh Count Clock (T_{RCY})	Interval Factor Value[Notes 1, 2]		
		ϕ = 20 MHz	ϕ = 33 MHz	ϕ = 50 MHz
7.8	32/ϕ	4 (6.4)	8 (7.8)	12 (7.7)
	128/ϕ	1 (6.4)	2 (7.8)	5 (7.7)
	256/ϕ	–	1 (7.8)	1 (5.1)
15.6	32/ϕ	9 (14.4)	16 (15.5)	24 (15.4)
	128/ϕ	2 (12.8)	4 (15.5)	6 (15.4)
	256/ϕ	1 (12.8)	2 (15.5)	3 (15.4)
31.2	32/ϕ	19 (30.4)	32 (31.0)	48 (30.7)
	128/ϕ	4 (25.6)	8 (31.0)	12 (30.7)
	256/ϕ	2 (25.6)	4 (31.0)	6 (30.7)
62.5	32/ϕ	39 (62.4)	64 (62.1)	–
	128/ϕ	9 (57.6)	16 (62.1)	24 (61.4)
	256/ϕ	4 (51.2)	8 (62.1)	12 (61.4)
125	128/ϕ	19 (121.6)	32 (124.1)	48 (122.9)
	256/ϕ	9 (115.2)	16 (124.1)	24 (122.9)
250	128/ϕ	39 (249.6)	64 (248.2)	–
	256/ϕ	19 (243.2)	32 (248.2)	48 (245.8)

Notes 1. The interval factor is set by bits RIN0n to RIN5n of the RFSn register (n = 1, 3, 4, 6).

2. The values in parentheses are the calculated values for the refresh interval (μs).
Refresh interval (μs) = Refresh count clock (T_{RCY}) $\times$ Interval factor

Remark ϕ: Internal system clock frequency

(2) Refresh wait control register (RWC)

This register specifies the number of wait states inserted during the refresh cycle.
This register can be read/written in 8-bit units.

Caution Write to the RWC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the RWC register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

RWC

7	6	5	4	3	2	1	0
RRW1	RRW0	RCW2	RCW1	RCW0	SRW2	SRW1	SRW0

Address: FFFFF49EH
After reset: 00H

Bit position	Bit name	Function
7, 6	RRW1, RRW0	Refresh RAS Wait Control Specifies the number of wait states inserted as hold time for the $\overline{\text{RAS}}$m signal's high level width during CBR refresh (m = 1, 3, 4, 6).
5 to 3	RCW2 to RCW0	Refresh Cycle Wait Control Specifies the number of wait states inserted as hold time for the $\overline{\text{RAS}}$m signal's low level width during CBR refresh (m = 1, 3, 4, 6).
2 to 0	SRW2 to SRW0	Self-refresh Release Wait Control Specifies the number of wait states inserted as CBR self-refresh release time.

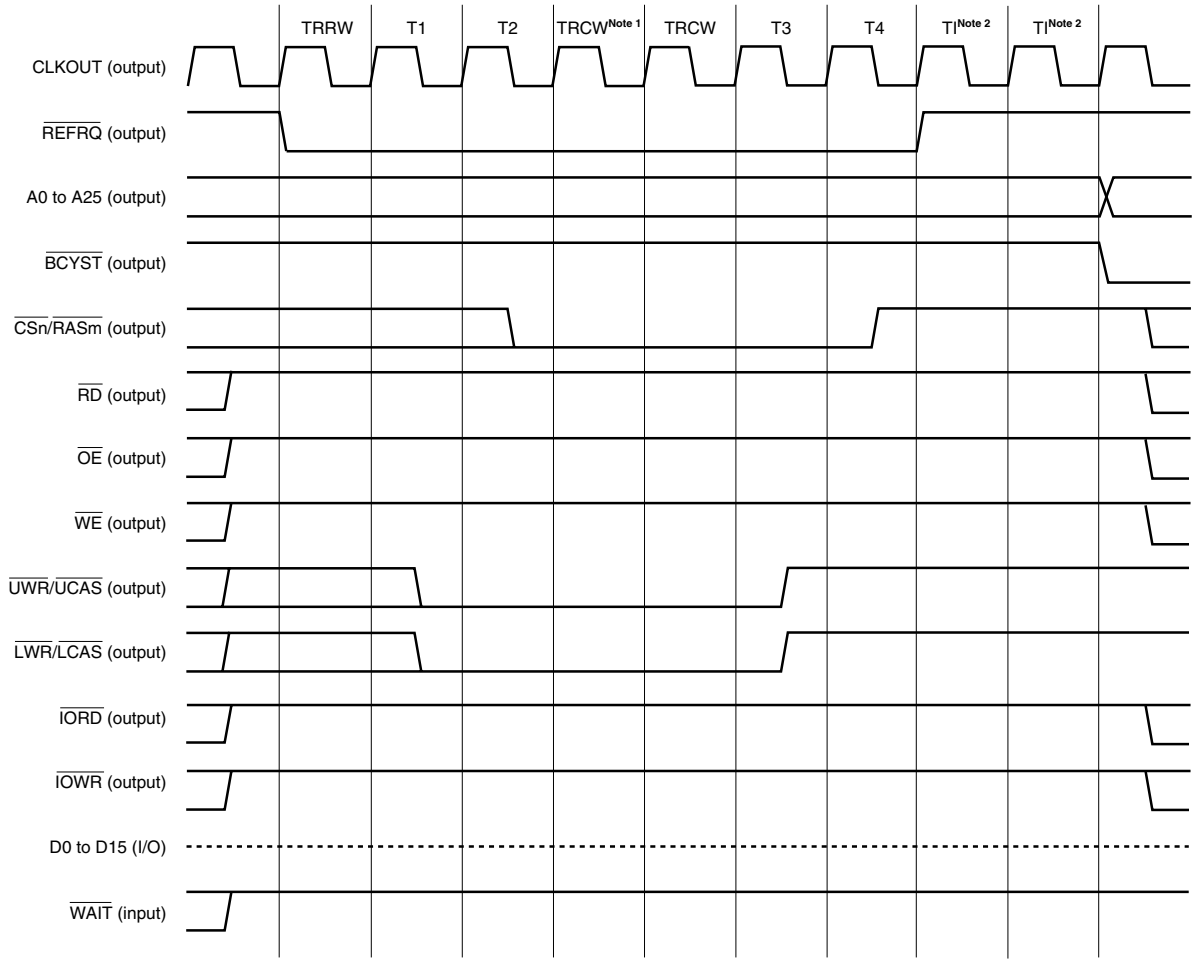
RRW1	RRW0	Number of inserted wait states
0	0	0
0	1	1
1	0	2
1	1	3

RCW2	RCW1	RCW0	Number of inserted wait states
0	0	0	1 (at least 1 wait is always inserted)
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

SRW2	SRW1	SRW0	Number of inserted wait states
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

(3) Refresh timing

Figure 5-9. CBR Refresh Timing



Notes 1. The TRCW cycle is always inserted for one or more clocks, irrespective of the setting of bits RCW2 to RCW0 of the RWC register.

2. This idle state (TI) is independent of the BCC register setting.

Remark n = 0 to 7, m = 1, 3, 4, 6

5.3.7 Self-refresh control function

When transferring to the IDLE or software STOP mode, or if the SELFREF signal becomes active, the DRAM controller generates the CBR self-refresh cycle.

Note that the $\overline{\text{RASn}}$ pulse width of DRAM must meet the specifications for DRAM to enable the self-refresh operation (n = 1, 3, 4, 6).

Cautions 1. When the transition to the self-refresh cycle is caused by SELFREF signal input, releasing the self-refresh cycle is only possible by inputting an inactive level to the SELFREF pin.

2. The internal ROM and internal RAM can be accessed even in the self-refresh cycle. However, access to a peripheral I/O register or external device is held pending until the self-refresh cycle is cleared.

To release the self-refresh cycle, use one of the three methods below.

(1) Release by NMI input

(a) In the case of self-refresh cycle in IDLE mode

To release the self-refresh cycle, make the $\overline{\text{RASn}}$, $\overline{\text{LCAS}}$, and $\overline{\text{UCAS}}$ signals inactive (high level) immediately.

(b) In the case of self-refresh cycle in software STOP mode

To release the self-refresh cycle, make the $\overline{\text{RASn}}$, $\overline{\text{LCAS}}$, and $\overline{\text{UCAS}}$ signals inactive (high level) after stabilizing oscillation.

(2) Release by INTP0n0 and INTP0n1 inputs (n = 0 to 3)

(a) In the case of self-refresh cycle in IDLE mode

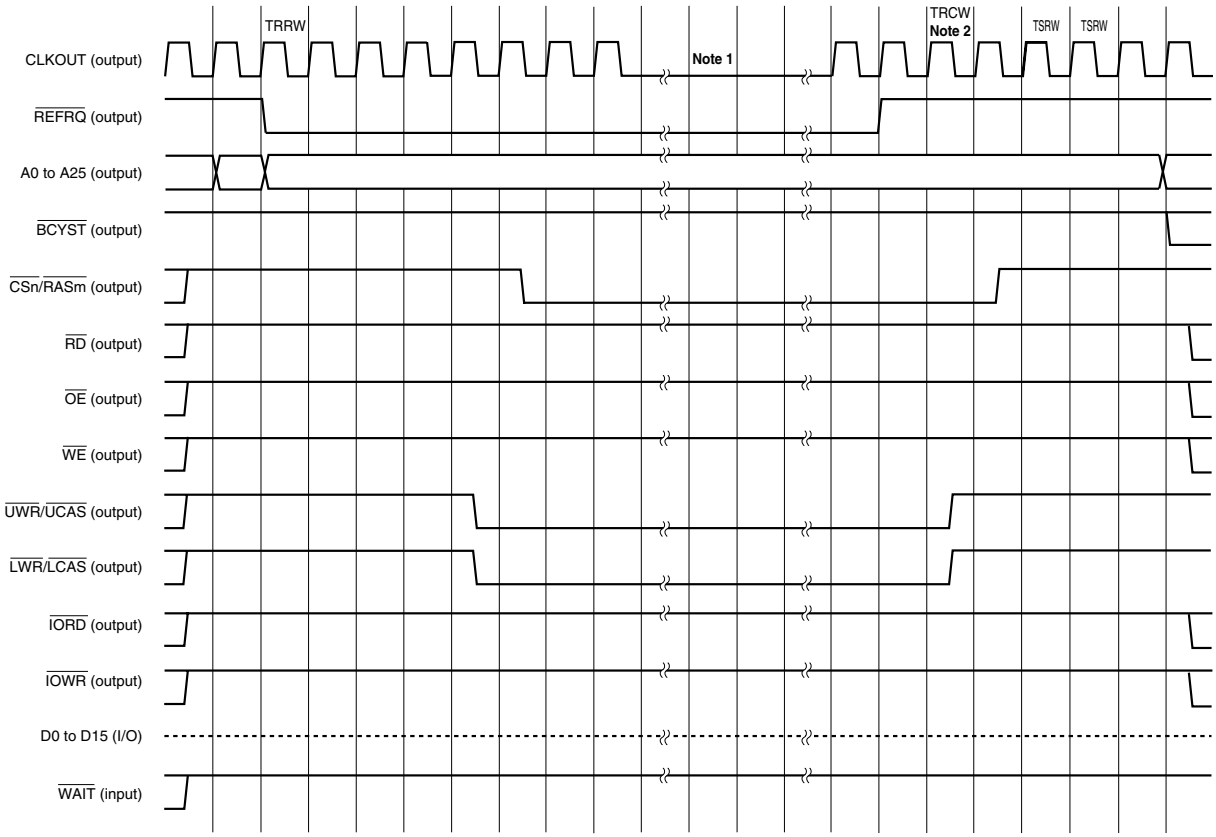
To release the self-refresh cycle, make the $\overline{\text{RASn}}$, $\overline{\text{LCAS}}$, and $\overline{\text{UCAS}}$ signals inactive (high level) immediately.

(b) In the case of self-refresh cycle in software STOP mode

To release the self-refresh cycle, make the $\overline{\text{RASn}}$, $\overline{\text{LCAS}}$, and $\overline{\text{UCAS}}$ signals inactive (high level) after stabilizing oscillation.

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

Figure 5-10. Self-Refresh Timing (DRAM)



Notes 1. Shown above is the case when the self-refresh cycle is started in the IDLE or software STOP mode. If the self-refresh cycle is started by inputting the active level of the SELFREF signal, CLKOUT is output without going low.

2. The TRCW cycle is always inserted for one or more clocks, irrespective of the setting of bits RCW2 to RCW0 of the RWC register.

Remarks 1. This timing is obtained when the bits of the RWC register have the following settings.
RRW1, RRW0 = 01B: 1 wait (TRRW)
RCW2 to RCW0 = 001B: 1 wait (TRCW)
SRW2 to SRW0 = 001B: 1 wait (TSRW) (double the number of wait states than the set value will be inserted)

2. n = 0 to 7, m = 1, 3, 4, 6

5.4 DRAM Controller (SDRAM)

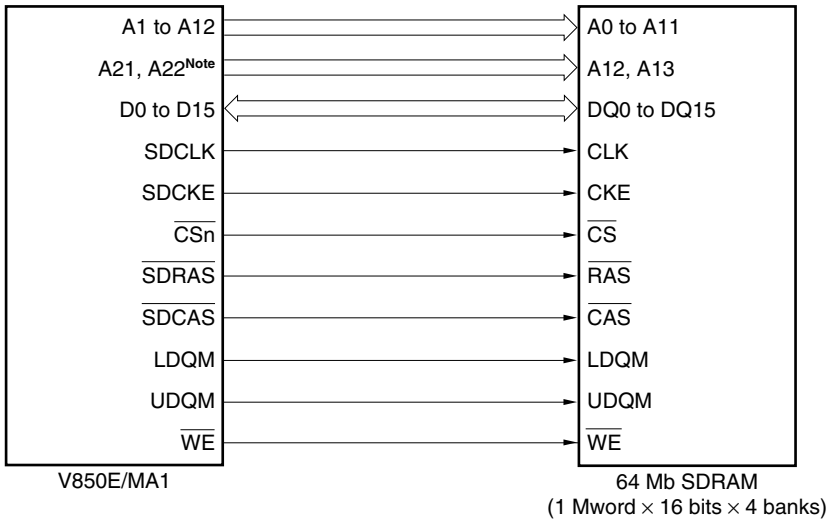
5.4.1 Features

- Burst length: 1
- Wrap type: Sequential
- CAS latency: 2 and 3 supported
- 4 types of SDRAM can be assigned to 4 memory blocks.
- Row and column address multiplex widths can be changed.
- Waits (0 to 3 waits) can be inserted between the bank active command and the read/write command.
- Supports CBR refresh and CBR self-refresh.

5.4.2 SDRAM connection

An example of connection to SDRAM is shown below.

Figure 5-11. Example of Connection to SDRAM



Note The address signals to be used differ depending on the SDRAM product.

Remark n = 1, 3, 4, 6

5.4.3 Address multiplex function

Depending on the value of the SAW0n and SAW1n bits in SDRAM configuration register n (SCRn), the row address output in the SDRAM cycle is multiplexed as shown in Figure 5-12 (a) (n = 1, 3, 4, 6). Depending on the value of the SSO0n and SSO1n bits, the column address output in the SDRAM cycle is multiplexed as shown in Figure 5-12 (b) (n = 1, 3, 4, 6). In Figures 5-12 (a) and (b), a0 to a25 indicate the addresses output from the CPU, and A0 to A25 indicate the address pins of the V850E/MA1.

Figure 5-12. Row Address/Column Address Output (1/2)

(a) Row address output

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address (SAW1n, SAW0n = 10)	a25 to a18	a17	a16	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
Row address (SAW1n, SAW0n = 01)	a25 to a18	a17	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
Row address (SAW1n, SAW0n = 00)	a25 to a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8

Remark n = 1, 3, 4, 6

(b) Column address output (using all bank precharge command)

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Column address (SSO1n, SSO0n = 00)	a25 to a18	a17	a16	a15	a14	a13	a12	a11	1	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
Column address (SSO1n, SSO0n = 01)	a25 to a18	a17	a16	a15	a14	a13	a12	1	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

Remark n = 1, 3, 4, 6

(c) Column address output (using register write command)

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Column address (SSO1n, SSO0n = 00)	0	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0
Column address (SSO1n, SSO0n = 01)	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0	0

Remark n = 1, 3, 4, 6

Figure 5-12. Row Address/Column Address Output (2/2)

(d) Column address output (using read/write command)

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Column address (SSO1n, SSO0n = 00)	a25 to a18	a17	a16	a15	a14	a13	a12	a11	0	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
Column address (SSO1n, SSO0n = 01)	a25 to a18	a17	a16	a15	a14	a13	a12	0	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

Remark n = 1, 3, 4, 6

5.4.4 SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6)

These registers specify the number of waits and the address multiplex width. SCRn corresponds to $\overline{CSn}$ (n = 1, 3, 4, 6). For example, to connect SDRAM to $\overline{CS1}$, set SCR1.

These registers can be read/written in 16-bit units.

Cautions 1. The SDRAM read/write cycle is not generated prior to executing the power-on cycle. Access SDRAM after waiting 20 clocks following a program write to the SCR register. To write to the SCR register again following access to SDRAM, clear the MEn bit of the BCT0 and BCT1 registers to 0, and then set it to 1 again before performing access (n = 0 to 7).

2. Do not execute continuous instructions to write to the SCR register. Be sure to insert another instruction between commands to write to the SCR register.

(1/2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
SCR1	0	LTM21	LTM11	LTM01	0	0	0	0	BCW11	BCW01	SSO11	SSO01	RAW11	RAW01	SAW11	SAW01	FFFFF4A4H	0000H
SCR3	0	LTM23	LTM13	LTM03	0	0	0	0	BCW13	BCW03	SSO13	SSO03	RAW13	RAW03	SAW13	SAW03	FFFFF4ACH	0000H
SCR4	0	LTM24	LTM14	LTM04	0	0	0	0	BCW14	BCW04	SSO14	SSO04	RAW14	RAW04	SAW14	SAW04	FFFFF4B0H	0000H
SCR6	0	LTM26	LTM16	LTM06	0	0	0	0	BCW16	BCW06	SSO16	SSO06	RAW16	RAW06	SAW16	SAW06	FFFFF4B8H	0000H

Bit position	Bit name	Function
14 to 12	LTM2n to LTM0n (n = 1, 3, 4, 6)	Latency Sets the CAS latency value for reading.

LTM2n	LTM1n	LTM0n	Latency
0	0	×	2
0	1	0	2
0	1	1	3
1	×	×	Setting prohibited

Bit position	Bit name	Function
7, 6	BCW1n, BCW0n (n = 1, 3, 4, 6)	Bank Active Command Wait Control Specifies the number of wait states inserted from the bank active command to a read/write command, or from the precharge command to the bank active command.

BCW1n	BCW0n	Number of wait states inserted
0	0	1 (at least 1 wait is always inserted)
0	1	1
1	0	2
1	1	3

Remark ×: don't care

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>SSO1n,
SSO0n
(n = 1 3,
4, 6)</td><td>SDRAM Shift Width On-Page Control
Specifies the address shift width during on-page judgment.
When the external data bus width is 8 bits: Set SSO1n, SSO0n = 00B
When the external data bus width is 16 bits: Set SSO1n, SSO0n = 01B

<table><tr><th>SSO1n</th><th>SSO0n</th><th>Address shift width</th></tr><tr><td>0</td><td>0</td><td>8 bits</td></tr><tr><td>0</td><td>1</td><td>16 bits</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited</td></tr></table></td></tr>
<tr><td>3, 2</td><td>RAW1n,
RAW0n
(n = 1, 3,
4, 6)</td><td>Row Address Width Control
Specifies the row address width.

<table><tr><th>RAW1n</th><th>RAW0n</th><th>Row address width</th></tr><tr><td>0</td><td>0</td><td>11</td></tr><tr><td>0</td><td>1</td><td>12</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited</td></tr></table></td></tr>
<tr><td>1, 0</td><td>SAW1n,
SAW0n
(n = 1, 3,
4, 6)</td><td>Row Address Multiplex Width Control
Specifies the address multiplex width during SDRAM access.

<table><tr><th>SAW1n</th><th>SAW0n</th><th>Address multiplex width</th></tr><tr><td>0</td><td>0</td><td>8</td></tr><tr><td>0</td><td>1</td><td>9</td></tr><tr><td>1</td><td>0</td><td>10</td></tr><tr><td>1</td><td>1</td><td>Setting prohibited</td></tr></table></td></tr>
</table>

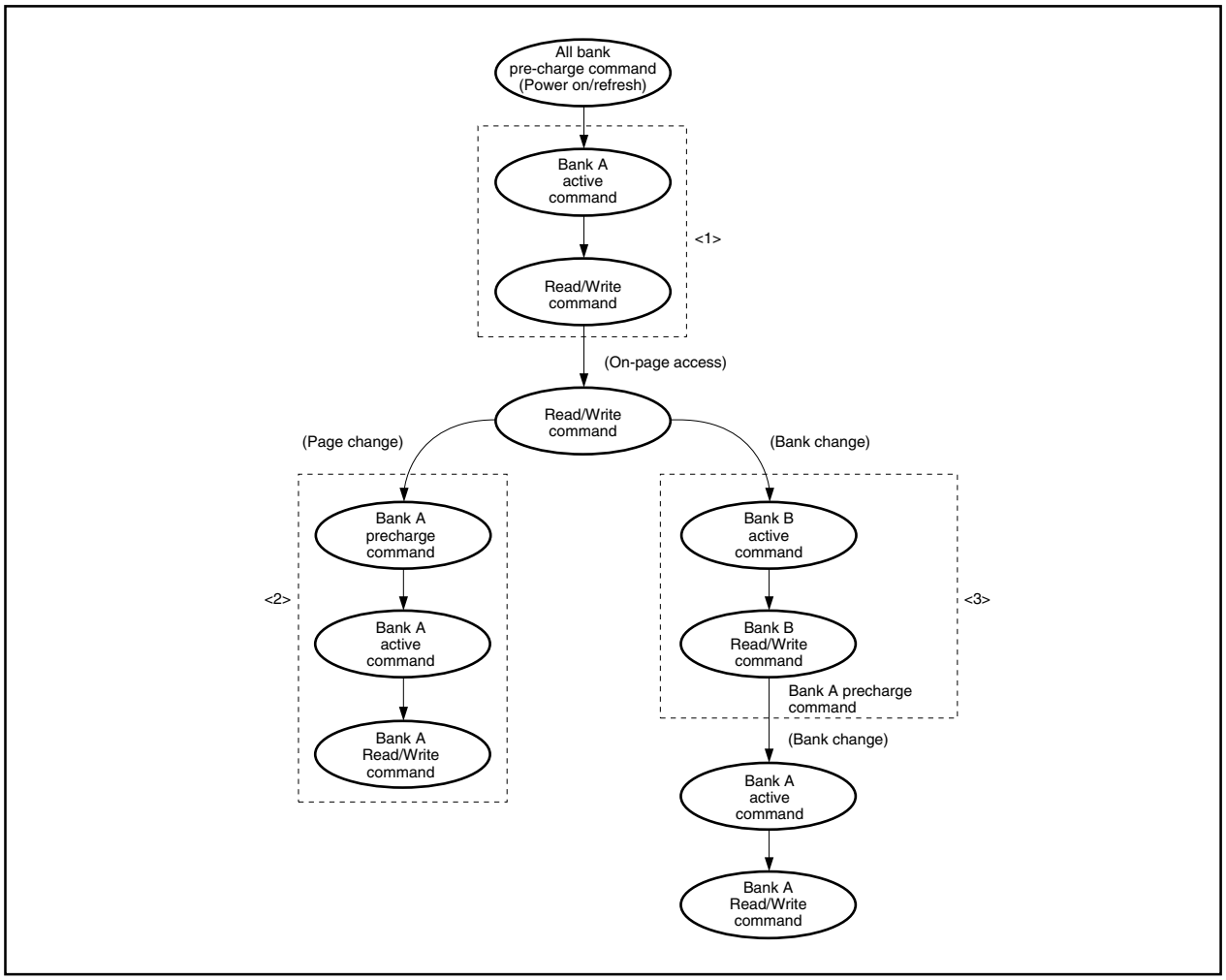
5.4.5 SDRAM access

During power-on or a refresh operation, the all bank precharge command is always issued for SDRAM. When accessing SDRAM after that, therefore, the active command and read/write command are issued in that order (see <1> in Figure 5-13).

If a page change occurs following this, the precharge command, active command, and read/write command are issued in that order (see <2> in Figure 5-13).

If a bank change occurs, the active command and read/write command for the bank to be accessed next are issued in that order. Following this read/write command, the precharge command for the bank that was accessed before the bank currently being accessed will be issued (see <3> in Figure 5-13).

Figure 5-13. State Transition of SDRAM Access



(1) SDRAM single read cycle

The SDRAM single read cycle is a cycle for reading from SDRAM by executing a load instruction (LD) for the SDRAM area, by fetching an instruction, or by 2-cycle DMA transfer.

In the SDRAM single read cycle, the active command (ACT) and read command (RD) are issued for SDRAM in that order. During on-page access, however, only the read command is issued and the precharge command and active command are not issued. When a page change occurs in the same bank, the precharge command (PR) is issued before the active command.

The timing to sample data is synchronized with rising of the UDQM and LDQM signals.

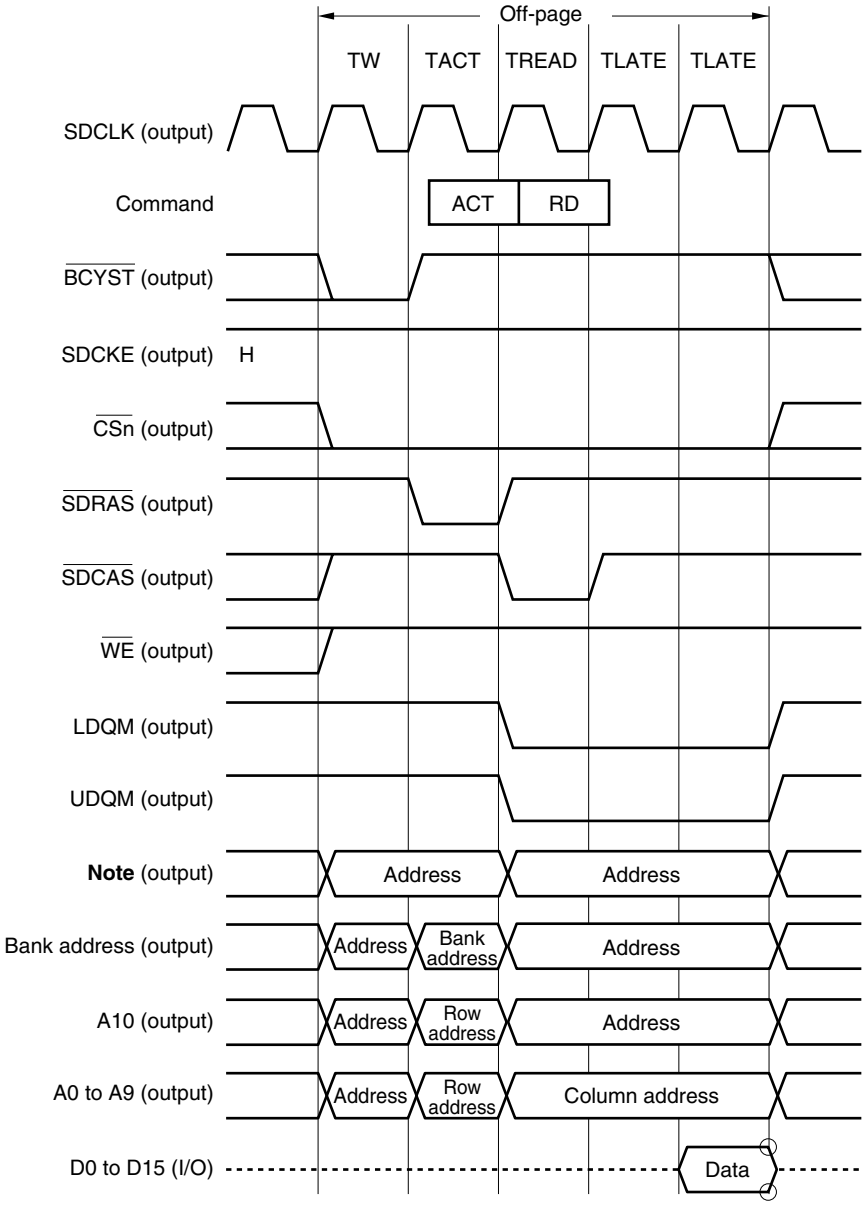
A one-state TW cycle is always inserted immediately before every read command, which is activated by the CPU.

The number of idle states set by the bus cycle control register (BCC) are inserted before the read cycle (no idle states are inserted, however, if BCn1 and BCn0 are 00) (n = 1, 3, 4, 6). The timing charts of the SDRAM single read cycle are shown below.

Caution When executing a write access to SRAM or external I/O after read accessing SDRAM, data conflict may occur depending on the SDRAM data output float delay time. In such a case, avoid data conflict by inserting an idle state in the SDRAM space via a setting in the BCC register.

Figure 5-14. SDRAM Single Read Cycle (1/3)

(a) During off-page access (when latency = 2)



Note Addresses other than the bank address, A10, and A0 to A9.

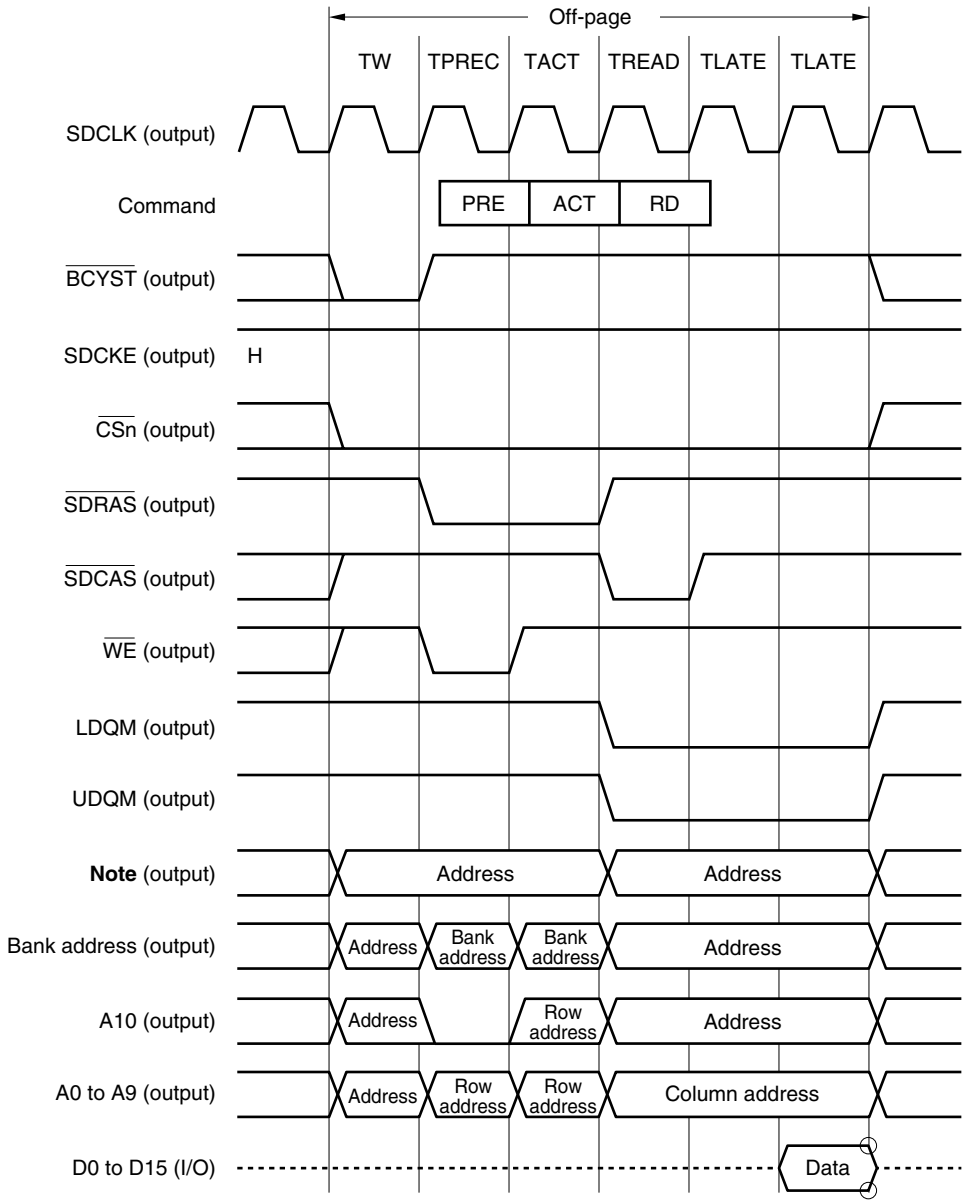
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

Figure 5-14. SDRAM Single Read Cycle (2/3)

(b) During off-page access (when latency = 2, page change)



Note Addresses other than the bank address, A10, and A0 to A9.

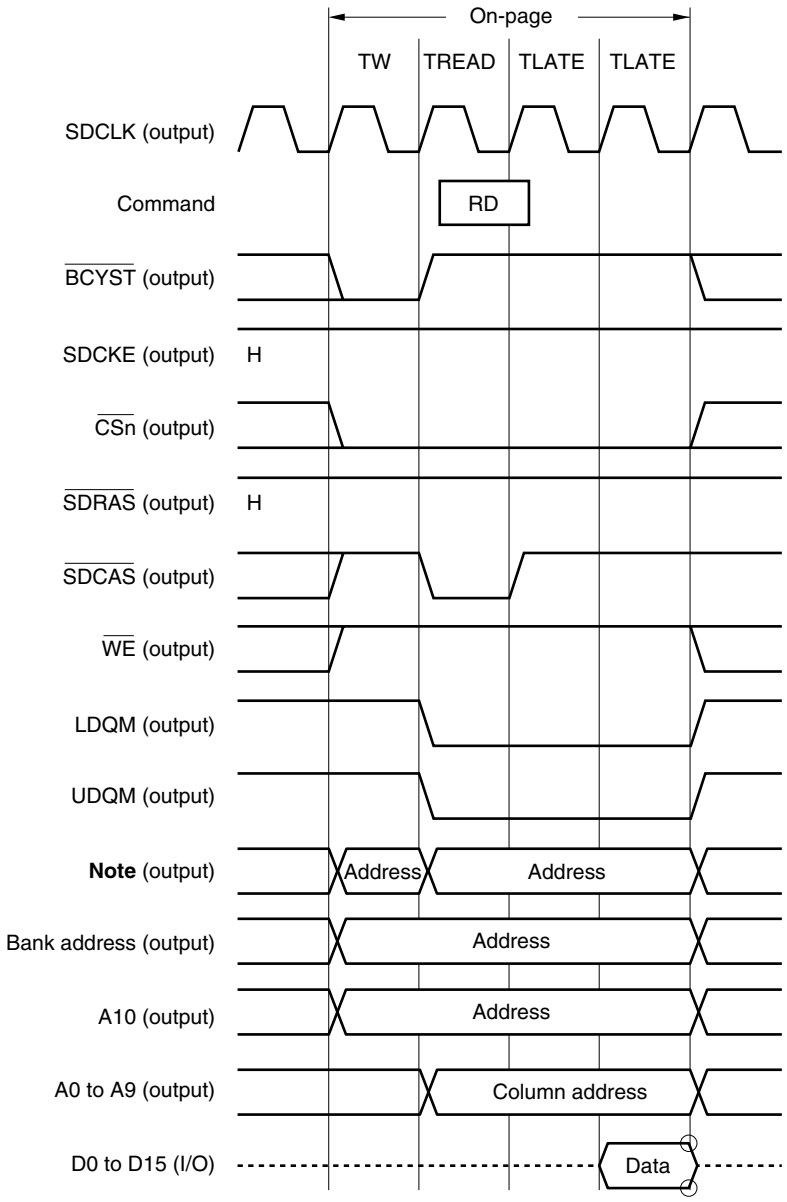
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

Figure 5-14. SDRAM Single Read Cycle (3/3)

(c) During on-page access (when latency = 2)



Note Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

4. The timing chart shown here is the timing when the previous cycle accessed another CS space or when the bus was in an idle state. If access to the same CS space continues, a TW state is not inserted (the $\overline{\text{BCYST}}$ signal becomes active in the TREAD state).

(2) SDRAM single write cycle

The SDRAM single write cycle is a cycle for writing to SDRAM by executing a write instruction (ST) for the SDRAM area or by 2-cycle DMA transfer.

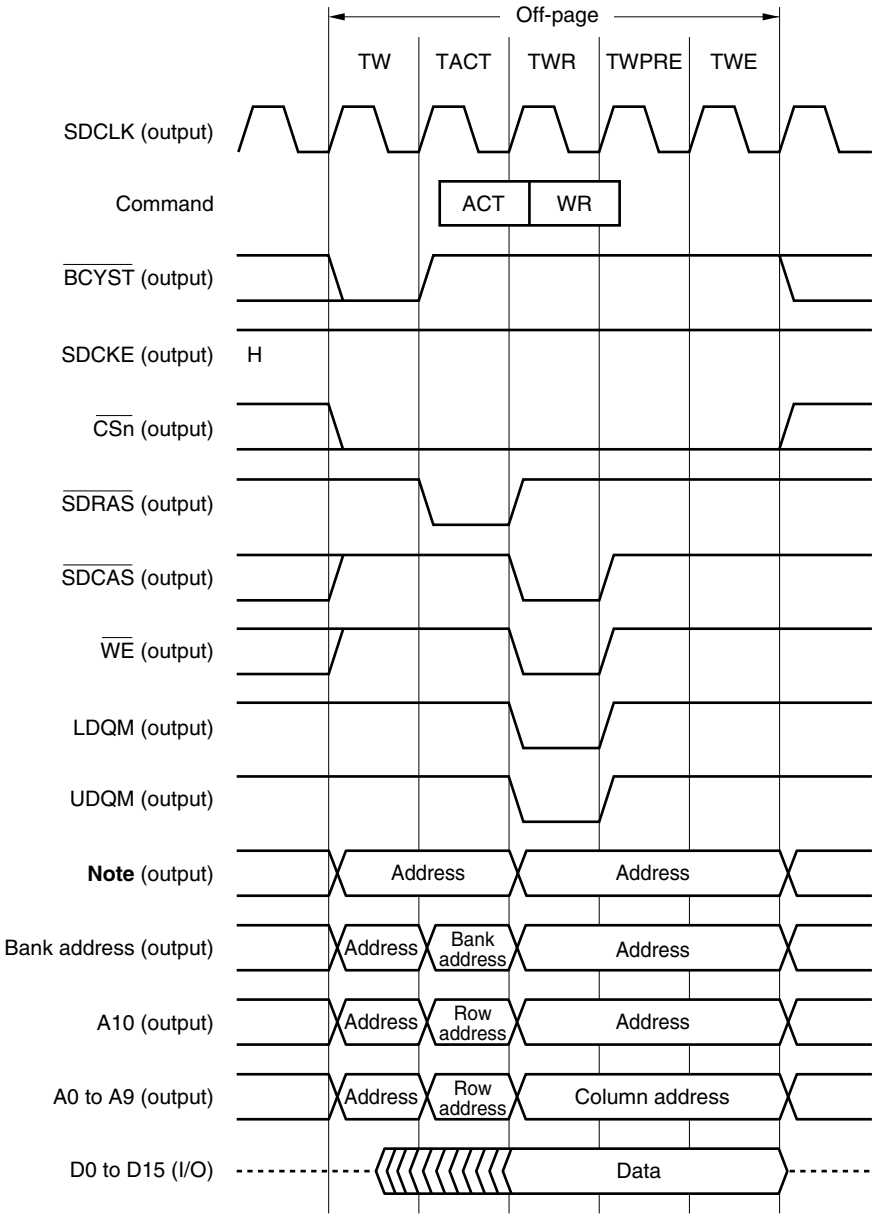
In the SDRAM single write cycle, the active command (ACT) and write command (WR) are issued for SDRAM in that order. During on-page access, however, only the write command is issued and the precharge command and active command are not issued. When a page change occurs in the same bank, the precharge command (PR) is issued before the active command.

A one-state TW cycle is always inserted immediately before every write command, which is activated by the CPU.

The timing charts of the SDRAM single write cycle are shown below.

Figure 5-15. SDRAM Single Write Cycle (1/3)

(a) During off-page access



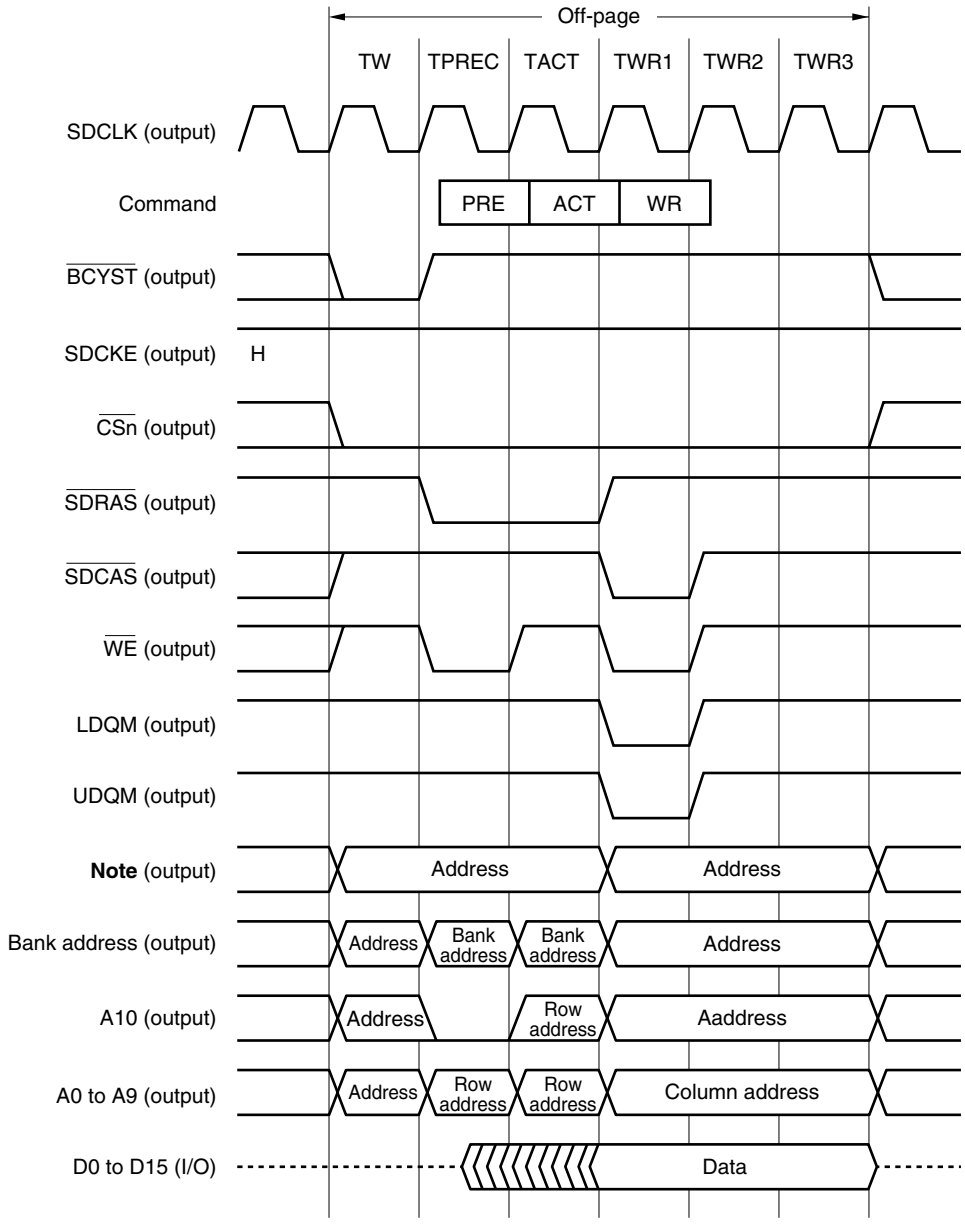
Note Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6

Figure 5-15. SDRAM Single Write Cycle (2/3)

(b) During off-page access (page change)



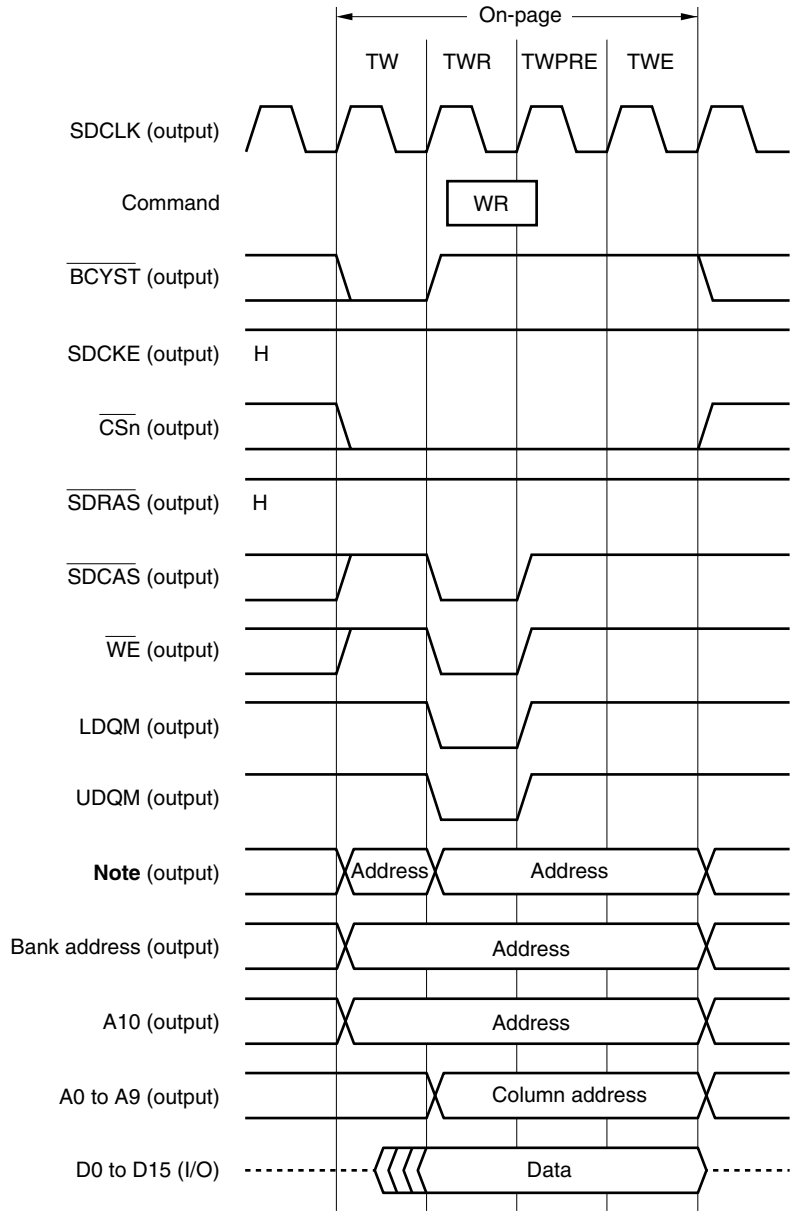
Note Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6

Figure 5-15. SDRAM Single Write Cycle (3/3)

(c) During on-page access



Note Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6

3. The timing chart shown here is the timing when the previous cycle accessed another CS space or when the bus is an idle state. If access to the same CS space continues, a TW state is not inserted (the BCYST signal becomes active in the TWR1 state).

(3) SDRAM access timing control

The SDRAM access timing can be controlled by SDRAM configuration register n (SCRn) (n = 1, 3, 4, 6). For details, see **5.4.4 SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6)**.

Caution Wait control by the $\overline{\text{WAIT}}$ pin is not available during SDRAM access.

(a) Number of waits from bank active command to read/write command

The number of wait states from bank active command issue to read/write command issue can be set by setting the BCW1n and BCW0n bits of the SCRn register.

BCW1n, BCW0n = 01B: 1 wait
BCW1n, BCW0n = 10B: 2 waits
BCW1n, BCW0n = 11B: 3 waits

(b) Number of waits from precharge command to bank active command

The number of wait states from precharge command issue to bank active command issue can be set by setting the BCW1n and BCW0n bits of the SCRn register.

BCW1n, BCW0n = 01B: 1 wait
BCW1n, BCW0n = 10B: 2 waits
BCW1n, BCW0n = 11B: 3 waits

(c) CAS latency setting when read

The CAS latency during a read operation can be set by setting the LTM2n to LTM0n bits of the SCRn register.

LTM2n to LTM0n = 010B: Latency = 2
LTM2n to LTM0n = 011B: Latency = 3

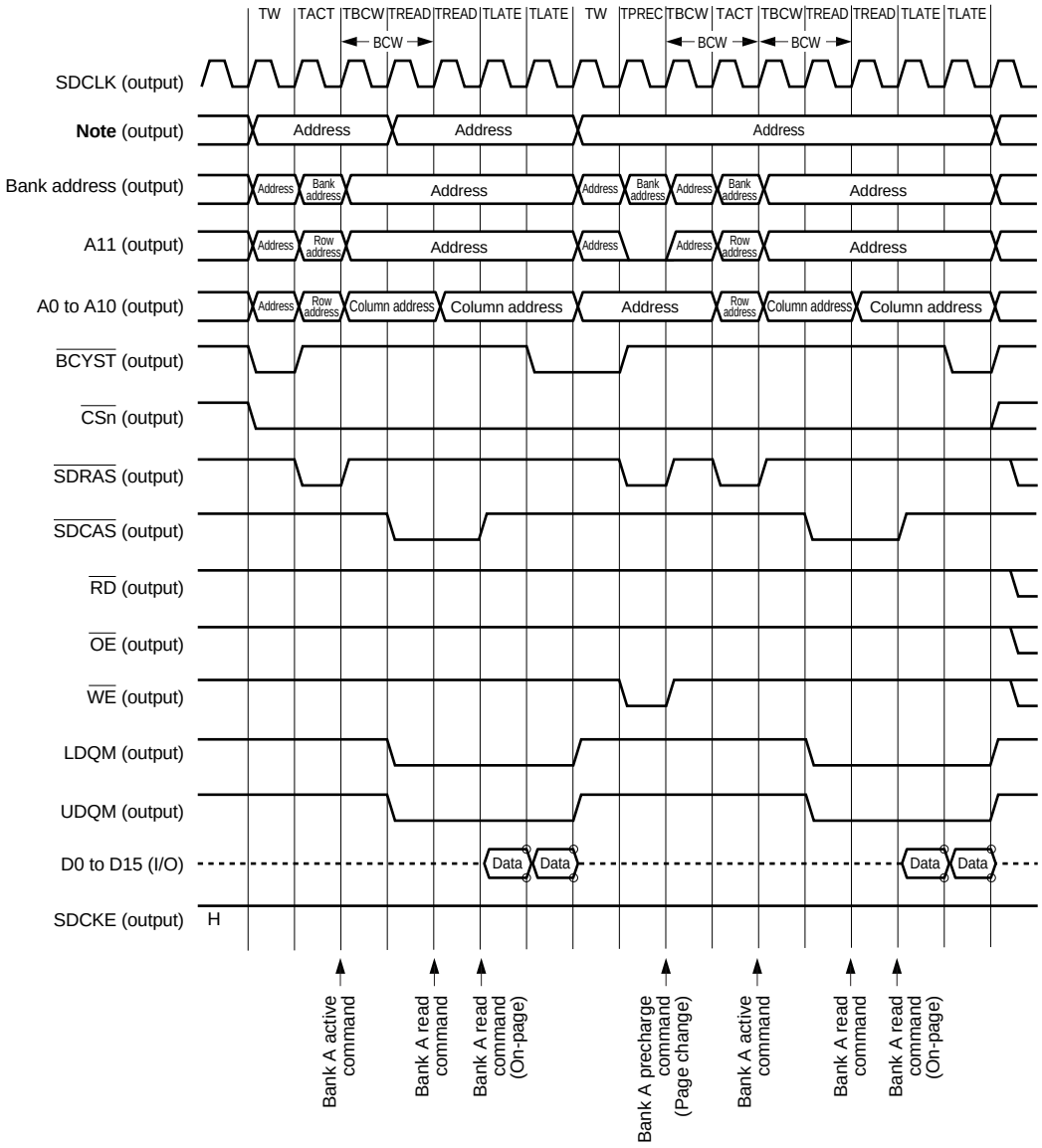
(d) Number of waits from refresh command to next command

The number of wait states from refresh command issue to next command issue can be set by setting the BCW1n and BCW0n bits of the SCRn register. The number of wait states becomes four times the value set by BCW1n and BCW0n.

BCW1n, BCW0n = 01B: 4 waits
BCW1n, BCW0n = 10B: 8 waits
BCW1n, BCW0n = 11B: 12 waits

Figure 5-16. SDRAM Access Timing (1/4)

(a) Read timing (16-bit bus width word access, page change, BCW = 2, latency = 2)



Note Addresses other than the bank address, A11, and A0 to A10.

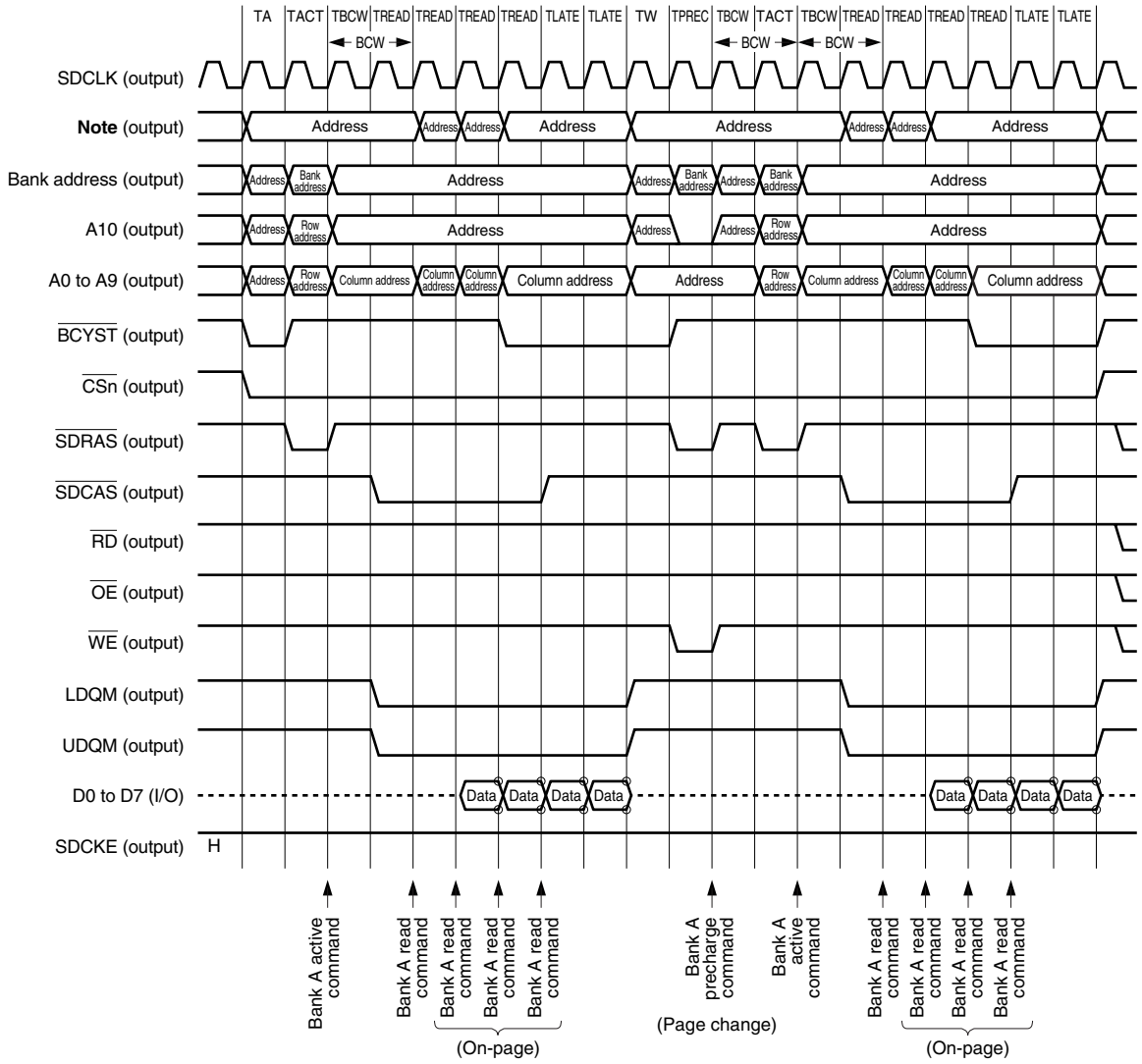
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

Figure 5-16. SDRAM Access Timing (2/4)

(b) Read timing (8-bit bus width word access, page change, BCW = 2, latency = 2)



Note Addresses other than the bank address, A10, and A0 to A9.

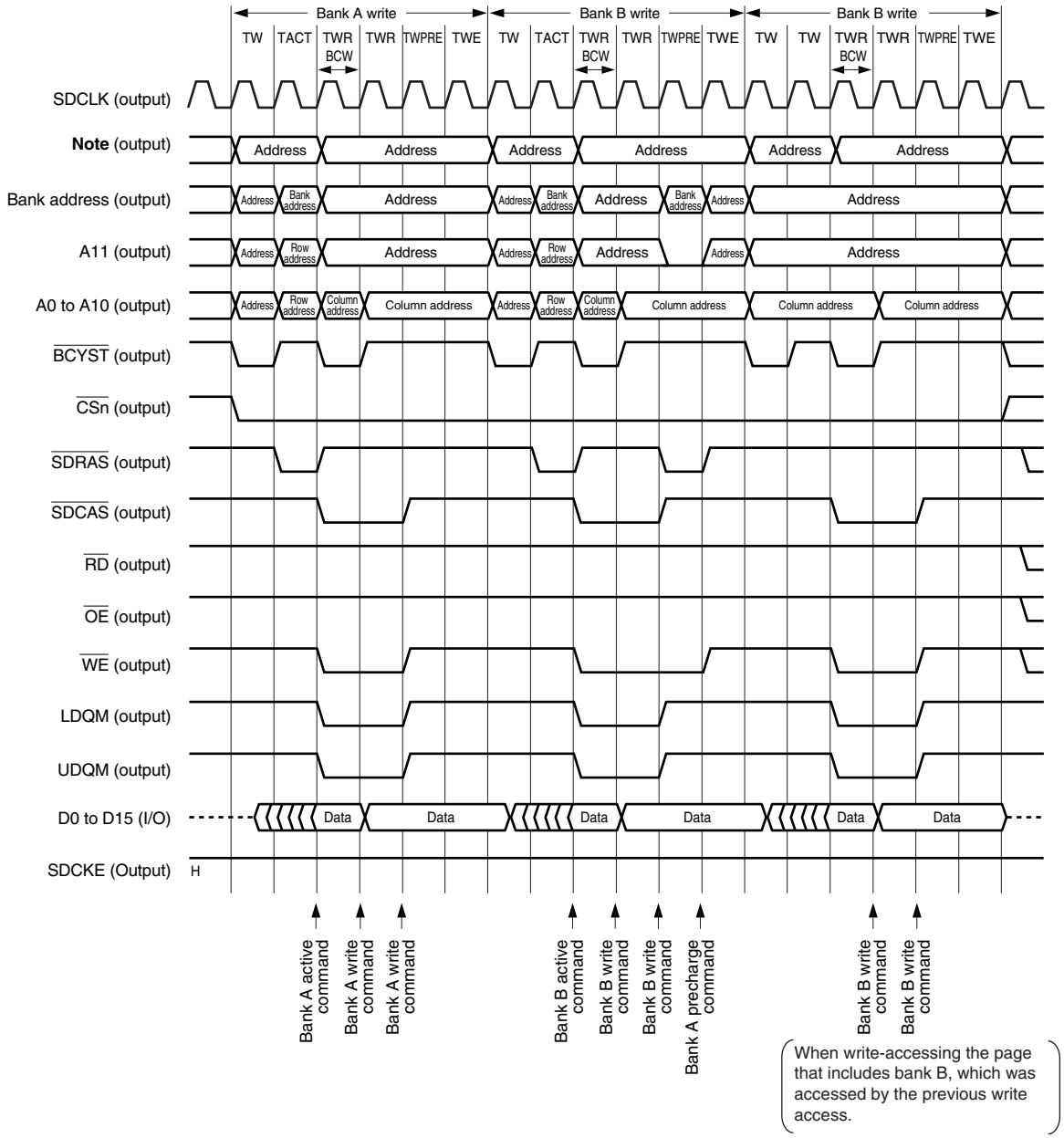
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

Figure 5-16. SDRAM Access Timing (3/4)

(c) Write timing (16-bit bus width word access, bank change, BCW = 1, latency = 2)



Note Addresses other than the bank address, A11, and A0 to A10.

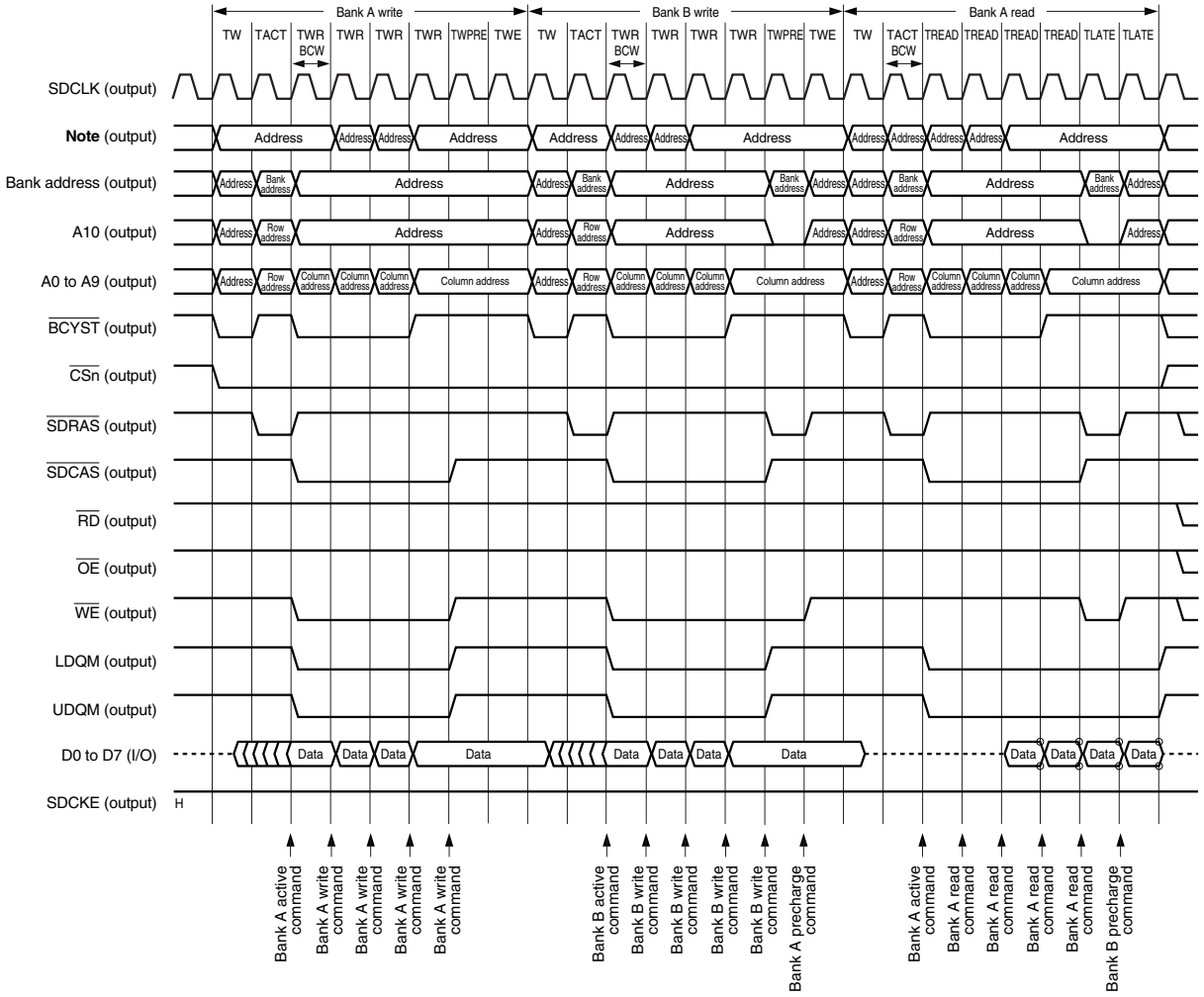
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

Figure 5-16. SDRAM Access Timing (4/4)

(d) Write timing (8-bit bus width word access, bank change, BCW = 1, latency = 2)



Note Addresses other than the bank address, A10, and A0 to A9.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 1, 3, 4, 6

5.4.6 Refresh control function

The V850E/MA1 can generate a refresh cycle. The refresh cycle is set with SDRAM refresh control registers 1, 3, 4, and 6 (RFS1, RFS3, RFS4, RFS6). RFSn corresponds to $\overline{CSn}$ (n = 1, 3, 4, 6). For example, to connect SDRAM to $\overline{CS1}$, set RFS1.

When another bus master occupies the external bus, the DRAM controller cannot occupy the external bus. In this case, the DRAM controller issues a refresh request to the bus master by changing the $\overline{REFRQ}$ signal to active (low level).

During a refresh operation, the address bus retains the state it was in just before the refresh cycle.

(1) SDRAM refresh control registers 1, 3, 4, 6 (RFS1, RFS3, RFS4, RFS6)

These registers are used to enable or disable a refresh and set the refresh interval. The refresh interval is determined by the following calculation formula.

Refresh interval (μs) = Refresh count clock (T_{RCY}) $\times$ Interval factor

The refresh count clock and interval factor are determined by the RENn bit and RIN5n to RIN0n bits, respectively, of the RFSn register.

Note that n corresponds to the register number (1, 3, 4, 6) of SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6).

These registers can be read/written in 16-bit units.

Caution Write to the RFS1, RFS3, RFS4, and RFS6 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the RFS1, RFS3, RFS4, and RFS6 registers are complete. However, it is possible to access external memory areas whose initialization settings are complete.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
RFS1	REN1	0	0	0	0	0	RCC11	RCC01	0	0	RIN51	RIN41	RIN31	RIN21	RIN11	RIN01	FFFFF4A6H	0000H
RFS3	REN3	0	0	0	0	0	RCC13	RCC03	0	0	RIN53	RIN43	RIN33	RIN23	RIN13	RIN03	FFFFF4AEH	0000H
RFS4	REN4	0	0	0	0	0	RCC14	RCC04	0	0	RIN54	RIN44	RIN34	RIN24	RIN14	RIN04	FFFFF4B2H	0000H
RFS6	REN6	0	0	0	0	0	RCC16	RCC06	0	0	RIN56	RIN46	RIN36	RIN26	RIN16	RIN06	FFFFF4BAH	0000H

Bit position	Bit name	Function
15	RENn (n = 1, 3, 4, 6)	Refresh Enable Specifies whether CBR refresh is enabled or disabled. 0: Refresh disabled 1: Refresh enabled
9, 8	RCC1n, RCC0n (n = 1, 3, 4, 6)	Refresh Count Clock Specifies the refresh count clock (T_{RCY}).
5 to 0	RIN5n to RIN0n (n = 1, 3, 4, 6)	Refresh Interval Sets the interval factor of the interval timer for the generation of the refresh timing.

RCC1n	RCC0n	Refresh count clock (T_{RCY})
0	0	32/ϕ
0	1	128/ϕ
1	0	256/ϕ
1	1	Setting prohibited

RIN5n	RIN4n	RIN3n	RIN2n	RIN1n	RIN0n	Interval factor
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
1	1	1	1	1	1	64

Remark ϕ: Internal system clock frequency

Table 5-3. Example of Interval Factor Settings

Specified Refresh Interval Value (μs)	Refresh Count Clock (T_{RCY})	Interval Factor Value[Notes 1, 2]		
		ϕ = 20 MHz	ϕ = 33 MHz	ϕ = 50 MHz
15.6	32/ϕ	9 (14.4)	16 (15.5)	24 (15.4)
	128/ϕ	2 (12.8)	4 (15.5)	6 (15.4)
	256/ϕ	1 (12.8)	2 (15.5)	3 (15.4)

Notes 1. The interval factor is set by bits RIN0n to RIN5n of the RFSn register (n = 1, 3, 4, 6).

2. The values in parentheses are the calculated values for the refresh interval (μs).
Refresh interval (μs) = Refresh count clock (T_{RCY}) $\times$ Interval factor

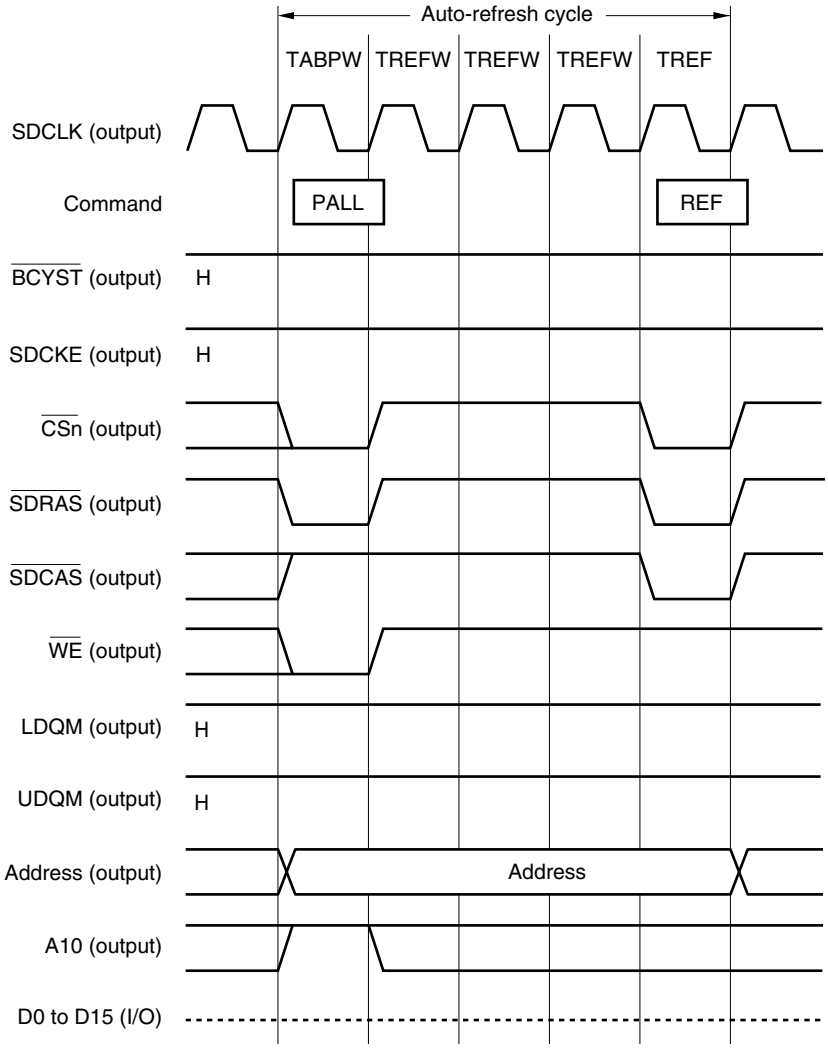
Remark ϕ: Internal system clock frequency

The V850E/MA1 can automatically generate an auto-refresh cycle and a self-refresh cycle.

(2) Auto-refresh cycle

In the auto-refresh cycle, the auto-refresh command (REF) is issued four clocks after the precharge command for all banks (PALL) is issued.

Figure 5-17. Auto-Refresh Cycle

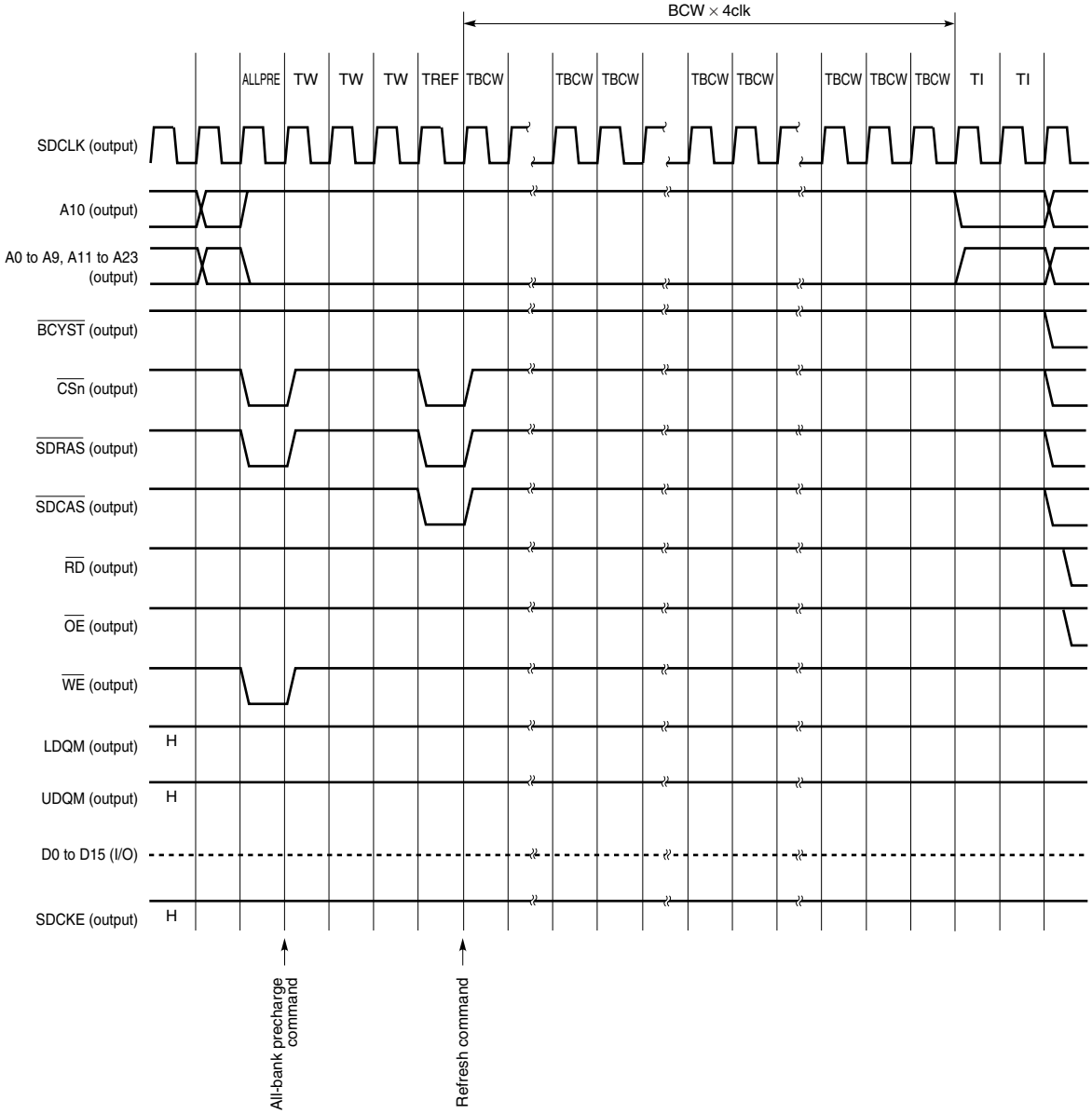


Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6

(3) Refresh timing

★ **Figure 5-18. CBR Refresh Timing (SDRAM)**



Remarks 1. The number of wait states set by the BCW1n and BCW0n bits of the SCRn register × 4 clocks will be inserted in the BCW × 4 clk period.

2. n = 1, 3, 4, 6

5.4.7 Self-refresh control function

In the case of transition to the IDLE or software STOP mode, or if the SELFREF signal becomes active, the DRAM controller generates the CBR self-refresh cycle (the system enters a state in which not only SDRAM, but also all DRAM is self-refreshed).

Note that the $\overline{\text{SDRAS}}$ pulse width of SDRAM must meet the specifications for SDRAM to enter the self-refresh operation.

Cautions 1. When the transition to the self-refresh cycle is caused by SELFREF signal input, releasing the self-refresh cycle is only possible by inputting an inactive level to the SELFREF pin.

2. The internal ROM and internal RAM can be accessed even in the self-refresh cycle. However, access to an on-chip peripheral I/O register or external device is held pending until the self-refresh cycle is cleared.

To release the self-refresh cycle, use one of the three methods below.

(1) Release by NMI input

(a) In the case of self-refresh cycle in IDLE mode

To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, LDQM, and UDQM signals inactive immediately.

(b) In the case of self-refresh cycle in software STOP mode

To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, LDQM, and UDQM signals inactive after stabilizing oscillation.

(2) Release by INTP0n0 and INTP0n1 inputs (n = 0 to 3)

(a) In the case of self-refresh cycle in IDLE mode

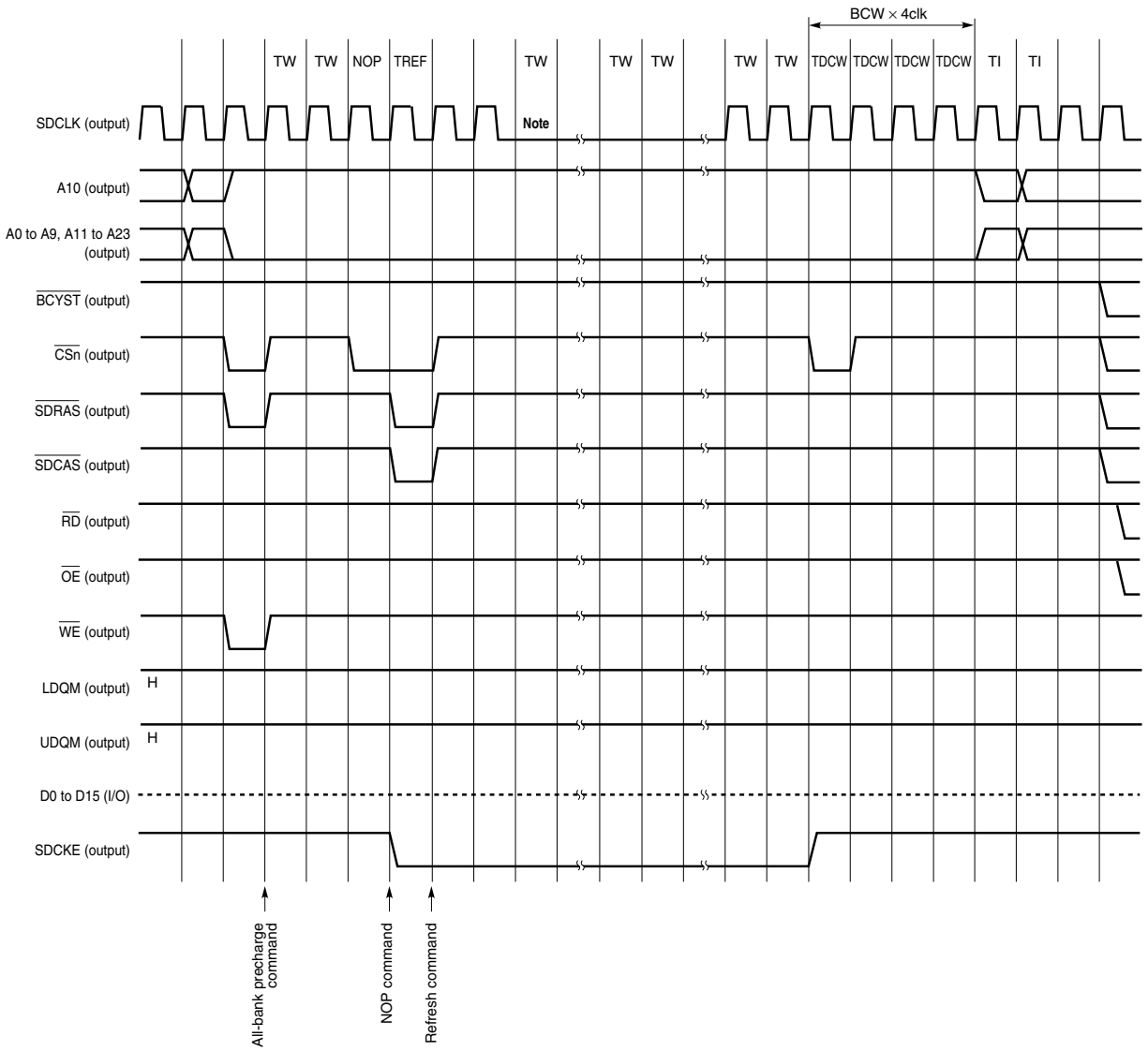
To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, LDQM, and UDQM signals inactive immediately.

(b) In the case of self-refresh cycle in software STOP mode

To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, LDQM, and UDQM signals inactive after stabilizing oscillation.

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

★ **Figure 5-19. Self-Refresh Timing (SDRAM)**



Note Shown above is the case when the self-refresh cycle is started in the IDLE or software STOP mode. If the self-refresh cycle is started by inputting the active level of the SELFREF signal, SDCLK is output without going low.

Remarks 1. The number of wait states set by the BCW1n and BCW0n bits of the SCRn register × 4 clocks will be inserted in the BCW × 4 clk period.

2. n = 1, 3, 4, 6

5.4.8 SDRAM initialization sequence

Be sure to initialize SDRAM when applying power.

(1) Set the registers of SDRAM (other than SDRAM configuration register n (SCRn))

- Bus cycle type configuration registers 0 and 1 (BCT0 and BCT1)
- Bus cycle control register (BCC)
- SDRAM refresh control registers 1, 3, 4, 6 (RFS1, RFS3, RFS4, RFS6)

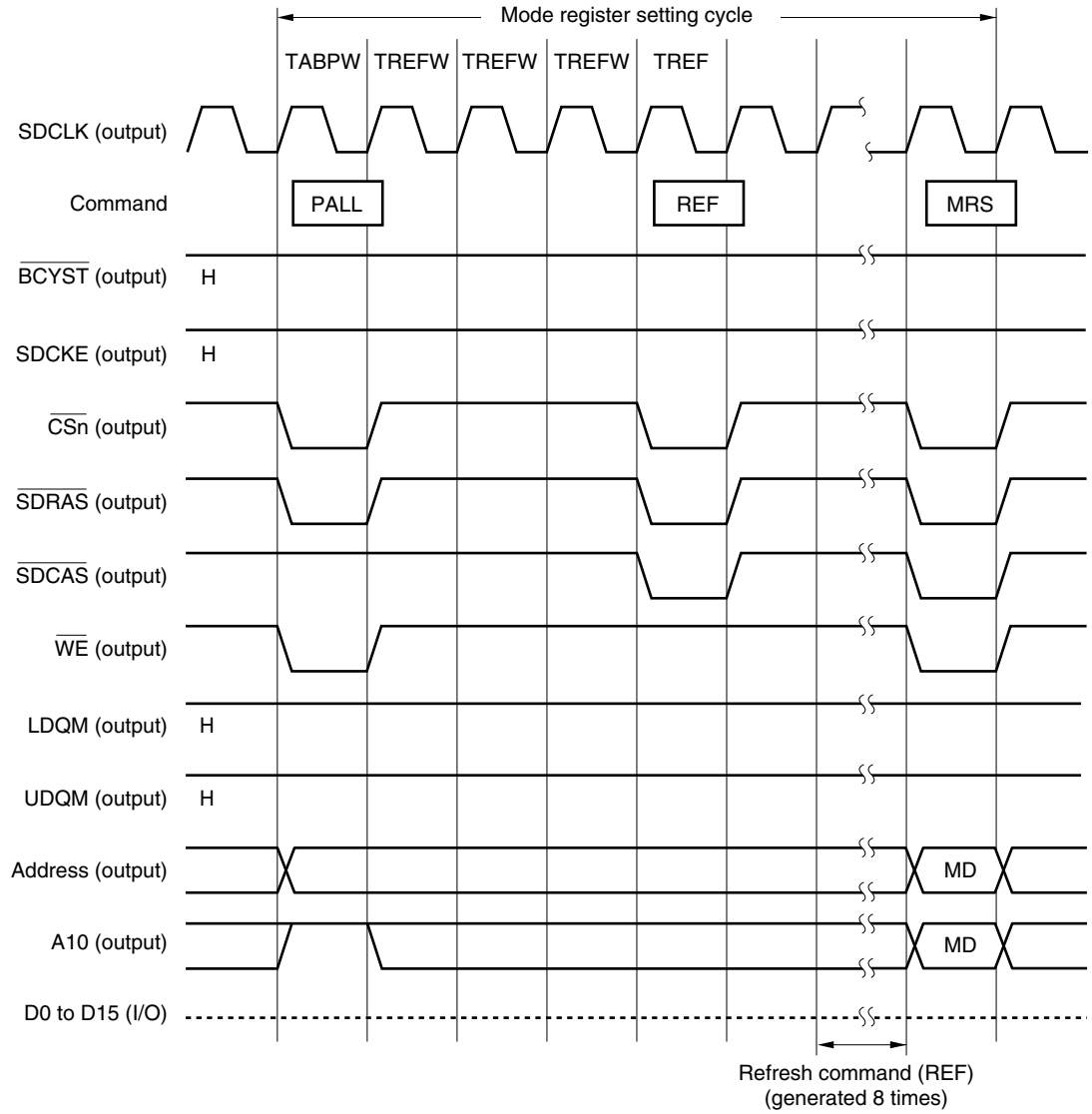
(2) Set SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6). When writing data to these registers, the following commands are issued for SDRAM in the order shown below.

- All bank precharge command
- Refresh command (8 times)
- Command that is used to set a mode register

Figures 5-20 and 5-21 show examples of the SDRAM mode register setting timing.

Caution When using the SDCLK and SDCKE signals, it is necessary to set the SDCLK output mode and the SDCKE output mode for these signals by setting the PMCCD register. In this case, however, these settings must not be executed at the same time.
Be sure to set the SDCKE output mode after setting the SDCLK output mode (refer to 14.3.14 (2) (b) Port CD mode control register (PMCCD)).

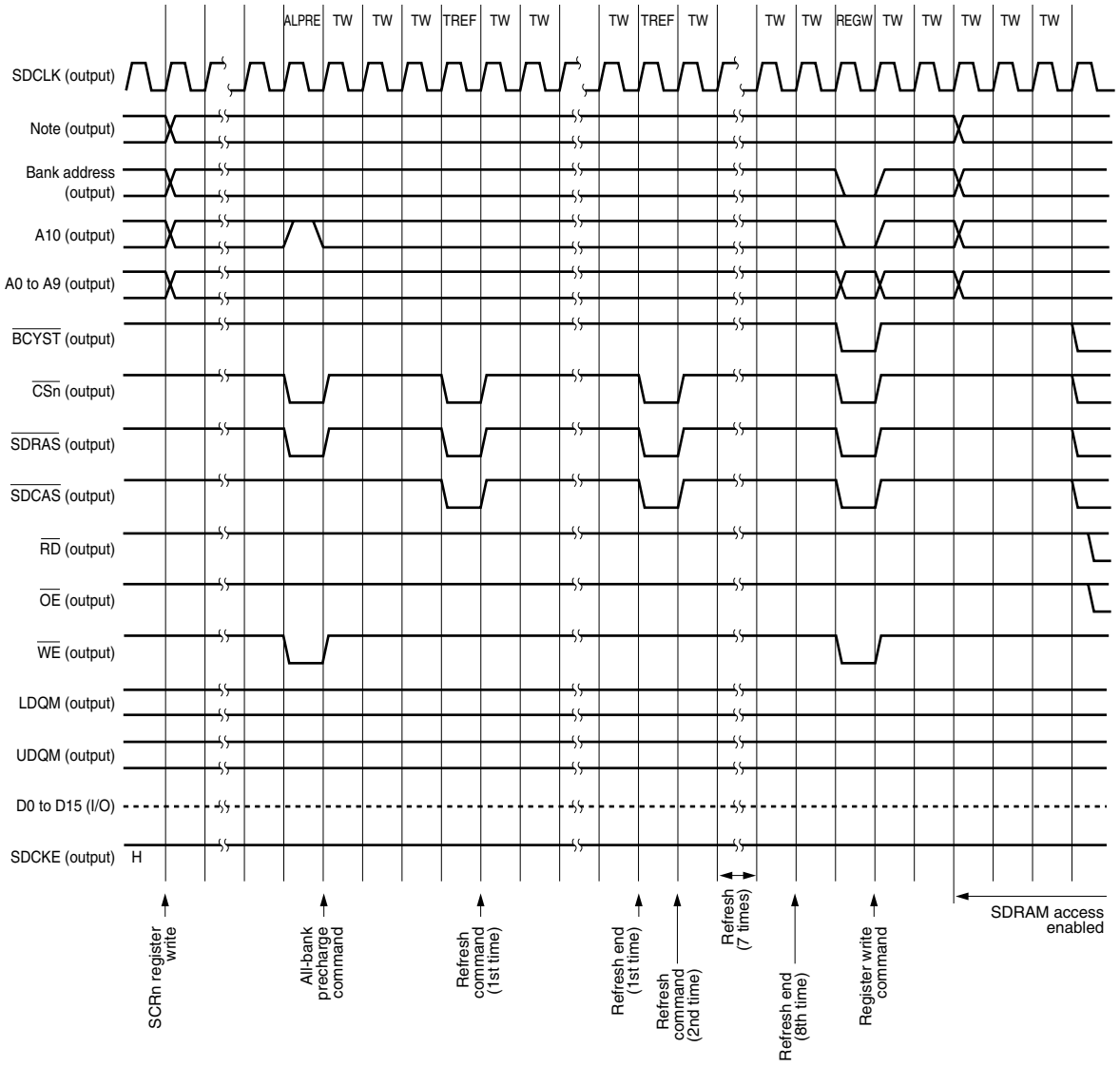
Figure 5-20. SDRAM Mode Register Setting Cycle



Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6

Figure 5-21. SDRAM Register Write Operation Timing



Note Addresses other than the bank address, A10, and A0 to A9.

Remark n = 1, 3, 4, 6

CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)

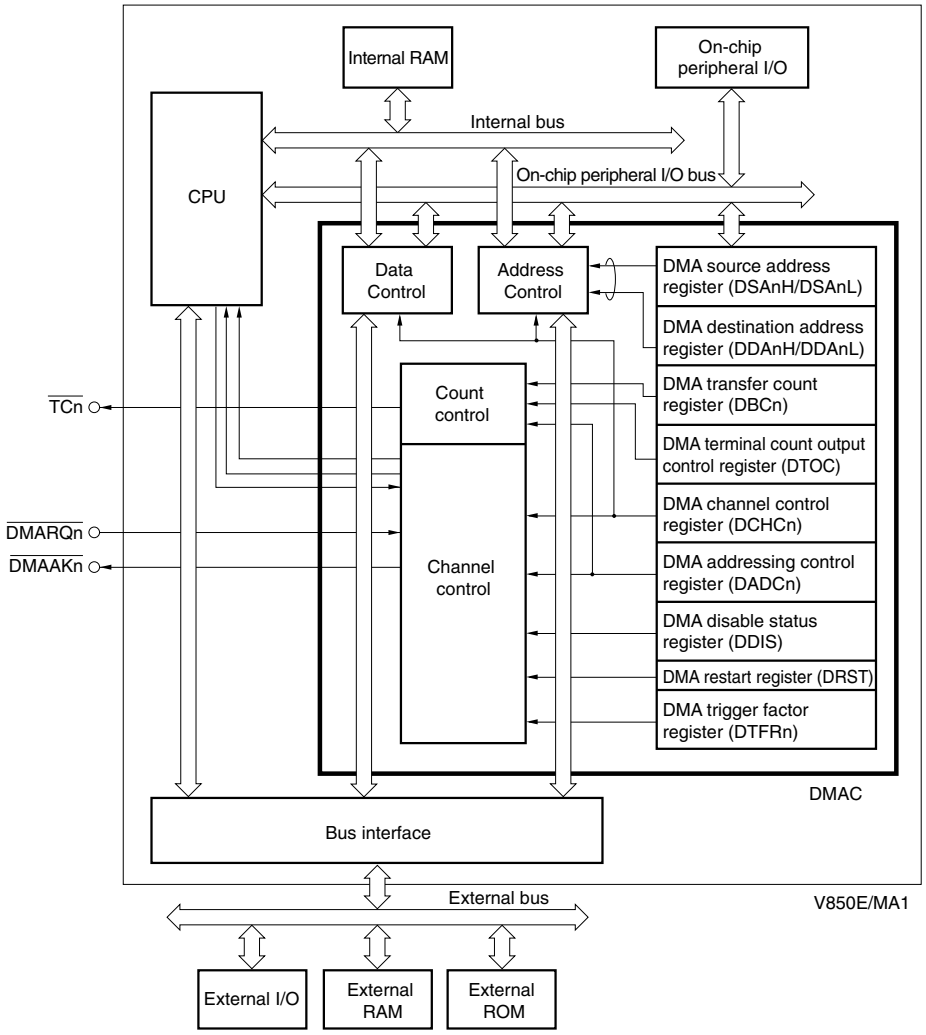
The V850E/MA1 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between memory and I/O, or among memories, based on DMA requests issued by the on-chip peripheral I/O (serial interface, real-time pulse unit, and A/D converter), $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ pins, or software triggers (memory refers to internal RAM or external memory).

6.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Two types of transfer
  - Flyby (1-cycle) transfer
  - 2-cycle transfer
- Three transfer modes
  - Single transfer mode
  - Single-step transfer mode
  - Block transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, real-time pulse unit, A/D converter)
  - Requests via $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ pin input
  - Requests by software trigger
- Transfer objects
  - Memory ↔ I/O
  - Memory ↔ memory
- DMA transfer end output signals ($\overline{\text{TC0}}$ to $\overline{\text{TC3}}$)
- Next address setting function

6.2 Configuration



Remark n = 0 to 3

6.3 Control Registers

6.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)

These registers are used to set the DMA source address (28 bits) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DSAnH and DSAnL.

Since these registers are configured as 2-stage FIFO buffer registers, a new source address for DMA transfer can be specified during DMA transfer. (Refer to **6.9 Next Address Setting Function**.)

When flyby transfer is specified with the TTYP bit of DMA addressing control register n (DADCn), the external memory addresses are set by the DSAn register. At this time, the setting of DMA destination address register n (DDAn) is ignored (n = 0 to 3).

(1) DMA source address registers 0H to 3H (DSA0H to DSA3H)

These registers can be read/written in 16-bit units.

Caution When setting an address of a peripheral I/O register for the source address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DSA0H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF082H	Undefined
DSA1H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF08AH	Undefined
DSA2H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF092H	Undefined
DSA3H	IR	0	0	0	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF09AH	Undefined

Bit position	Bit name	Function
15	IR	Internal RAM Select Specifies the DMA source address. 0: External memory, on-chip peripheral I/O 1: Internal RAM
11 to 0	SA27 to SA16	Source Address Sets the DMA source address (A27 to A16). During DMA transfer, it stores the next DMA transfer source address. During flyby transfer, it stores an external memory address.

(2) DMA source address registers 0L to 3L (DSA0L to DSA3L)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DSA0L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF080H	Undefined
DSA1L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF088H	Undefined
DSA2L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF090H	Undefined
DSA3L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF098H	Undefined

Bit position	Bit name	Function
15 to 0	SA15 to SA0	Source Address Sets the DMA source address (A15 to A0). During DMA transfer, it stores the next DMA transfer source address. During flyby transfer, it stores an external memory address.

6.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)

These registers are used to set the DMA destination address (28 bits) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

Since these registers are configured as 2-stage FIFO buffer registers, a new destination address for DMA transfer can be specified during DMA transfer. (Refer to **6.9 Next Address Setting Function**.)

When flyby transfer is specified with bit TTYP of DMA addressing control register n (DADCn), the setting of DMA destination address register n (DDAn) is ignored (n = 0 to 3).

(1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)

These registers can be read/written in 16-bit units.

Caution When setting an address of a peripheral I/O register for the destination address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DDA0H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF086H	Undefined
DDA1H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF08EH	Undefined
DDA2H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF096H	Undefined
DDA3H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF09EH	Undefined

Bit position	Bit name	Function
15	IR	Internal RAM Select Specifies the DMA destination address. 0: External memory, on-chip peripheral I/O 1: Internal RAM
11 to 0	DA27 to DA16	Destination Address Sets the DMA destination address (A27 to A16). During DMA transfer, it stores the next DMA transfer destination address. This setting is ignored during flyby transfer.

(2) DMA destination address registers 0L to 3L (DDA0L to DDA3L)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DDA0L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF084H	Undefined
DDA1L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF08CH	Undefined
DDA2L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF094H	Undefined
DDA3L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF09CH	Undefined

Bit position	Bit name	Function
15 to 0	DA15 to DA0	Destination Address Sets the DMA destination address (A15 to A0). During DMA transfer, it stores the next DMA transfer destination address. This setting is ignored during flyby transfer.

6.3.3 DMA byte count registers 0 to 3 (DBC0 to DBC3)

These 16-bit registers are used to set the byte transfer count for DMA channel n (n = 0 to 3). They store the remaining transfer count during DMA transfer.

Since these registers are configured as 2-stage FIFO buffer registers, a new DMA byte transfer count for DMA transfer can be specified during DMA transfer. (Refer to **6.9 Next Address Setting Function**.)

These registers are decremented by 1 for each transfer, and transfer ends when a borrow occurs.

These registers can be read/written in 16-bit units.

Remark If the DBCn register is read during DMA transfer after a terminal count has occurred without the register being overwritten, the value set immediately before the DMA transfer will be read out (0000H will not be read, even if DMA transfer has ended).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DBC0	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	FFFFF0C0H	Undefined
DBC1	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	FFFFF0C2H	Undefined
DBC2	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	FFFFF0C4H	Undefined
DBC3	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	FFFFF0C6H	Undefined

Bit position	Bit name	Function
15 to 0	BC15 to BC0	Byte Count Sets the byte transfer count and stores the remaining byte transfer count during DMA transfer.

DBCn (n = 0 to 3)	States
0000H	Byte transfer count 1 or remaining byte transfer count
0001H	Byte transfer count 2 or remaining byte transfer count
:	:
FFFFH	Byte transfer count 65,536 (2^{16}) or remaining byte transfer count

6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)

These 16-bit registers are used to control the DMA transfer mode for DMA channel n (n = 0 to 3). These registers cannot be accessed during DMA operation.

They can be read/written in 16-bit units.

(1/2)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DADC0	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF0D0H	0000H
DADC1	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF0D2H	0000H
DADC2	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF0D4H	0000H
DADC3	DS1	DS0	0	0	0	0	0	0	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF0D6H	0000H

Bit position	Bit name	Function
15, 14	DS1, DS0	Data Size Sets the transfer data size for DMA transfer.
7, 6	SAD1, SAD0	Source Address count Direction Sets the count direction of the source address for DMA channel n.
5, 4	DAD1, DAD0	Destination Address count Direction Sets the count direction of the destination address for DMA channel n.

DS1	DS0	Transfer data size
0	0	8 bits
0	1	16 bits
1	0	Setting prohibited
1	1	Setting prohibited

SAD1	SAD0	Count direction
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DAD1	DAD0	Count direction
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TM1, TM0</td><td>Transfer Mode
Sets the transfer mode during DMA transfer.

<table><tr><th>TM1</th><th>TM0</th><th>Transfer mode</th></tr><tr><td>0</td><td>0</td><td>Single transfer mode</td></tr><tr><td>0</td><td>1</td><td>Single-step transfer mode</td></tr><tr><td>1</td><td>0</td><td>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Block transfer mode</td></tr></table></td></tr>
<tr><td>1</td><td>TTYP</td><td>Transfer Type
Sets the DMA transfer type.
0: 2-cycle transfer
1: Flyby transfer</td></tr>
<tr><td>0</td><td>TDIR</td><td>Transfer Direction
Sets the transfer direction during transfer between I/O and memory. The setting is valid during flyby transfer only and ignored during 2-cycle transfer.
0: Memory → I/O (read)
1: I/O → memory (write)</td></tr>
</table>

6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)

These 8-bit registers are used to control the DMA transfer operating mode for DMA channel n (n = 0 to 3).

These registers can be read/written in 8-bit or 1-bit units. (However, bit 7 is read only and bits 2 and 1 are write only. If bits 2 and 1 are read, the read value is always 0.)

	<7>	6	5	4	<3>	<2>	<1>	<0>	Address	After reset
DCHC0	TC0	0	0	0	MLE0	INIT0	STG0	E00	FFFFF0E0H	00H
DCHC1	TC1	0	0	0	MLE1	INIT1	STG1	E11	FFFFF0E2H	00H
DCHC2	TC2	0	0	0	MLE2	INIT2	STG2	E22	FFFFF0E4H	00H
DCHC3	TC3	0	0	0	MLE3	INIT3	STG3	E33	FFFFF0E6H	00H

Bit position	Bit name	Function
7	TCn (n = 0 to 3)	Terminal Count This status bit indicates whether DMA transfer through DMA channel n has ended or not. This bit is read-only. It is set to 1 when DMA transfer ends and cleared (to 0) when it is read. 0: DMA transfer has not ended. 1: DMA transfer has ended.
3	MLEn (n = 0 to 3)	Multi Link Enable Bit When this bit is set to 1 at terminal count output, the Enn bit is not cleared to 0 and the DMA transfer enable state is retained. Moreover, the next DMA transfer request can be acknowledged even when the TCn bit is not read. When this bit is cleared to 0 at terminal count output, the Enn bit is cleared to 0 and the DMA transfer disable state is entered. At the next DMA request, the Enn bit must be set to 1 and the TCn bit read.
2	INITn (n = 0 to 3)	Initialize When this bit is set to 1, DMA transfer is forcibly terminated.
1	STGn (n = 0 to 3)	Software Trigger If this bit is set to 1 in the DMA transfer enable state (TCn bit = 0, Enn bit = 1), DMA transfer is started.
0	Enn (n = 0 to 3)	Enable Specifies whether DMA transfer through DMA channel n is to be enabled or disabled. This bit is cleared to 0 when DMA transfer ends. It is also cleared to 0 when DMA transfer is forcibly terminated by setting the INITn bit to 1 or by NMI input. 0: DMA transfer disabled 1: DMA transfer enabled

6.3.6 DMA disable status register (DDIS)

This register holds the contents of the Enn bit of the DCHCn register during NMI input (n = 0 to 3).

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

	7	6	5	4	3	2	1	0	Address	After reset
DDIS	0	0	0	0	CH3	CH2	CH1	CH0	FFFFF0F0H	00H

Bit position	Bit name	Function
3 to 0	CH3 to CH0	NMI Interrupt Status Reflects the contents of the Enn bit of the DCHCn register during NMI input. The contents of this register are held until the next NMI input or until the system is reset.

6.3.7 DMA restart register (DRST)

This register is used to restart DMA transfer that has been forcibly interrupted by NMI input. The ENn bit of this register and the Enn bit of the DCHCn register are linked to each other (n = 0 to 3). Following forcible interrupt by NMI input, the DMA channel that was interrupted is confirmed from the contents of the DDIS register, and DMA transfer is restarted by setting the ENn bit of the corresponding channel to 1.

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

	7	6	5	4	3	2	1	0	Address	After reset
DRST	0	0	0	0	EN3	EN2	EN1	EN0	FFFFF0F2H	00H

Bit position	Bit name	Function
3 to 0	EN3 to EN0	Restart Enable Specifies whether DMA transfer through DMA channel n is to be enabled or disabled. This bit is cleared to 0 when DMA transfer is completed in accordance with the terminal count output. It is also cleared to 0 when DMA transfer is forcibly terminated by setting the INITn bit to 1 or by NMI input. 0: DMA transfer disabled 1: DMA transfer enabled

6.3.8 DMA terminal count output control register (DTOC)

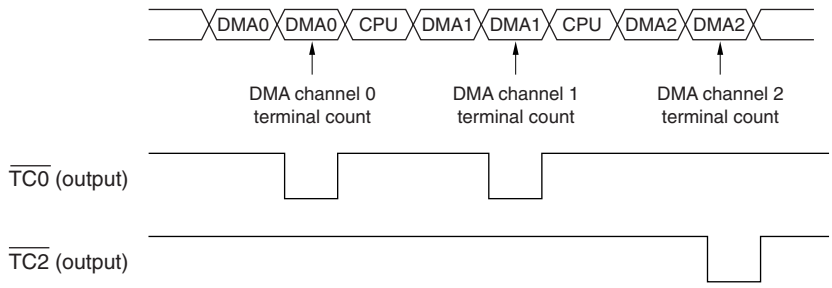
The DMA terminal count output control register (DTOC) is an 8-bit register that controls the terminal count output from each DMA channel. Terminal count signals from each DMA channel can be brought together and output from the $\overline{TC0}$ pin.

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

	7	6	5	4	3	2	1	0	Address	After reset
DTOC	0	0	0	0	TCO3	TCO2	TCO1	TCO0	FFFFF8A0H	01H

Bit position	Bit name	Function
3 to 0	TCO3 to TCO0	Terminal Count Output Indicates the state of the $\overline{TCn}$ pin (n = 0 to 3). 0: Channel n terminal count signal not output from $\overline{TC0}$ pin. 1: Channel n terminal count signal output from $\overline{TC0}$ pin.

The following shows an example of the case when the DTOC register is set to 03H.



6.3.9 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

These 8-bit registers are used to control the DMA transfer start trigger through interrupt requests from on-chip peripheral I/O.

The interrupt requests set by these registers serve as DMA transfer startup factors.

These registers can be read/written in 8-bit units. However, only bit 7 (DFn) can be read/written in 1-bit units.

Caution An interrupt request input in the standby mode (IDLE or software STOP mode) cannot be a DMA transfer start factor.

	<7>	6	5	4	3	2	1	0	Address	After reset
DTFR0	DF0	0	IFC05	IFC04	IFC03	IFC02	IFC01	IFC00	FFFFF810H	00H
DTFR1	DF1	0	IFC15	IFC14	IFC13	IFC12	IFC11	IFC10	FFFFF812H	00H
DTFR2	DF2	0	IFC25	IFC24	IFC23	IFC22	IFC21	IFC20	FFFFF814H	00H
DTFR3	DF3	0	IFC35	IFC34	IFC33	IFC32	IFC31	IFC30	FFFFF816H	00H

Bit position	Bit name	Function
7	DFn	DMA Request Flag This is a DMA transfer request flag. Only 0 can be written to this flag. 0: DMA transfer not requested 1: DMA transfer requested If the interrupt specified as the DMA transfer startup trigger occurs and it is necessary to clear the DMA transfer request while DMA transfer is disabled (including when it is aborted by NMI or forcibly terminated by software), stop the operation of the source causing the interrupt, and then clear the DFn bit to 0 (for example, disable reception in the case of serial reception). If it is clear that the interrupt will not occur until DMA transfer is resumed next, it is not necessary to stop the operation of the source causing the interrupt.
5 to 0	IFCn5 to IFCn0	Interrupt Factor Code This code is used to set the interrupt sources serving as DMA transfer startup factors.

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
0	0	0	0	0	0	DMA request from on-chip peripheral I/O disabled
0	0	0	0	0	1	INTP000/INTM000
0	0	0	0	1	0	INTP001/INTM001
0	0	0	0	1	1	INTP010/INTM010
0	0	0	1	0	0	INTP011/INTM011
0	0	0	1	0	1	INTP020/INTM020
0	0	0	1	1	0	INTP021/INTM021

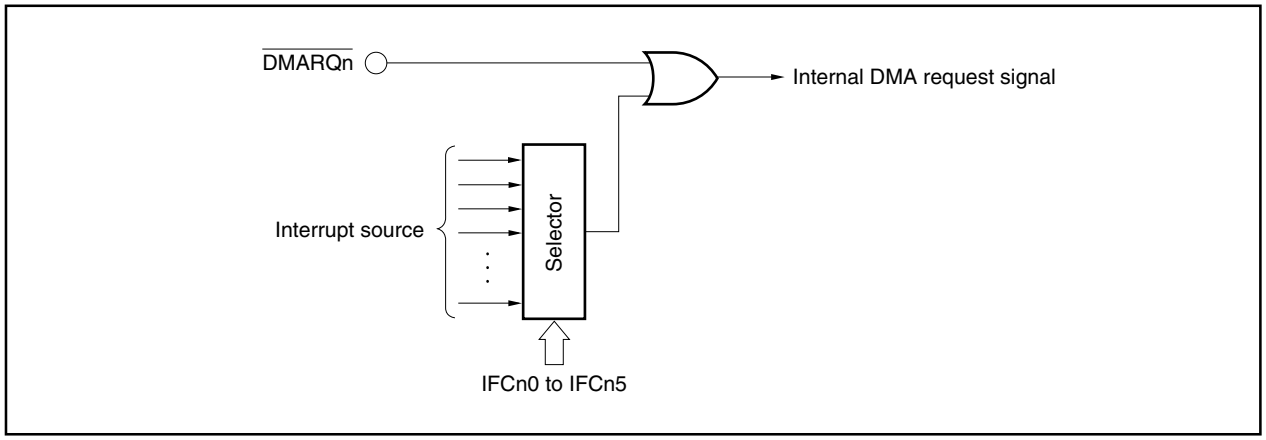
(2/2)

Bit position	Bit name	Function
5 to 0	IFCn5 to IFCn0	

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt source
0	0	0	1	1	1	INTP030/INTM030
0	0	1	0	0	0	INTP031/INTM031
0	0	1	0	0	1	INTP100
0	0	1	0	1	0	INTP101
0	0	1	0	1	1	INTP102
0	0	1	1	0	0	INTP103
0	0	1	1	0	1	INTP110
0	0	1	1	1	0	INTP111
0	0	1	1	1	1	INTP112
0	1	0	0	0	0	INTP113
0	1	0	0	0	1	INTP120
0	1	0	0	1	0	INTP121
0	1	0	0	1	1	INTP122
0	1	0	1	0	0	INTP123
0	1	0	1	0	1	INTP130
0	1	0	1	1	0	INTP131
0	1	0	1	1	1	INTP132
0	1	1	0	0	0	INTP133
0	1	1	0	0	1	INTCMD0
0	1	1	0	1	0	INTCMD1
0	1	1	0	1	1	INTCMD2
0	1	1	1	0	0	INTCMD3
0	1	1	1	0	1	INTCSI0
0	1	1	1	1	0	INTSR0
0	1	1	1	1	1	INTST0
1	0	0	0	0	0	INTCSI1
1	0	0	0	0	1	INTSR1
1	0	0	0	1	0	INTST1
1	0	0	0	1	1	INTCSI2
1	0	0	1	0	0	INTSR2
1	0	0	1	0	1	INTST2
1	0	0	1	1	0	INTAD
Other than above						Setting prohibited

Remark n = 0 to 3

The relationship between the $\overline{\text{DMARQn}}$ signal and the interrupt source that serves as a DMA transfer trigger is as follows (n = 0 to 3)



Remark If an interrupt request is specified as the DMA transfer start factor, an interrupt request will be generated if DMA transfer starts. To prevent an interrupt from being generated, mask the interrupt by setting the interrupt request control register. DMA transfer starts even if an interrupt is masked.

6.4 DMA Bus States

6.4.1 Types of bus states

The DMAC bus states consist of the following 13 states.

(1) TI state

The TI state is an idle state, during which no access request is issued.

The $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ signals are sampled at the rising edge of the CLKOUT signal.

(2) T0 state

DMA transfer ready state (state in which a DMA transfer request has been issued and the bus mastership is acquired for the first DMA transfer).

(3) T1R state

The bus enters the T1R state at the beginning of a read operation in the 2-cycle transfer mode.

Address driving starts. After entering the T1R state, the bus invariably enters the T2R state.

(4) T1RI state

The T1RI state is a state in which the bus waits for the acknowledge signal corresponding to an external memory read request.

After entering the last T1RI state, the bus invariably enters the T2R state.

(5) T2R state

The T2R state corresponds to the last state of a read operation in the 2-cycle transfer mode, or to a wait state.

In the last T2R state, read data is sampled. After entering the last T2R state, the bus invariably enters the T1W state.

(6) T2RI state

State in which the bus is ready for DMA transfer to on-chip peripheral I/O or internal RAM (state in which the bus mastership is acquired for DMA transfer to on-chip peripheral I/O or internal RAM).

After entering the last T2RI state, the bus invariably enters the T1W state.

(7) T1W state

The bus enters the T1W state at the beginning of a write operation in the 2-cycle transfer mode.

Address driving starts. After entering the T1W state, the bus invariably enters the T2W state.

(8) T1WI state

State in which the bus waits for the acknowledge signal corresponding to an external memory write request.

After entering the last T1WI state, the bus invariably enters the T2W state.

(9) T2W state

The T2W state corresponds to the last state of a write operation in the 2-cycle transfer mode, or to a wait state.

In the last T2W state, the write strobe signal is made inactive.

(10) T1FH state

The basic flyby transfer state, this state corresponds to the transfer execution cycle.

After entering the T1FH state, the bus enters the T2FH state.

(11) T1FHI state

The T1FHI state corresponds to the last state of a flyby transfer, during which the end of transfer is waited for.

After entering the T1FHI state, the bus is released and enters the TE state.

(12) T2FH state

The T2FH state is the state during which it is judged whether flyby transfer is to be continued or not.

If the next transfer is executed in the block transfer mode, the bus enters the T1FH state after the T2FH state.

Under other conditions, the bus enters the T1FHI state when a wait is issued. If no wait is issued, the bus is released and enters the TE state.

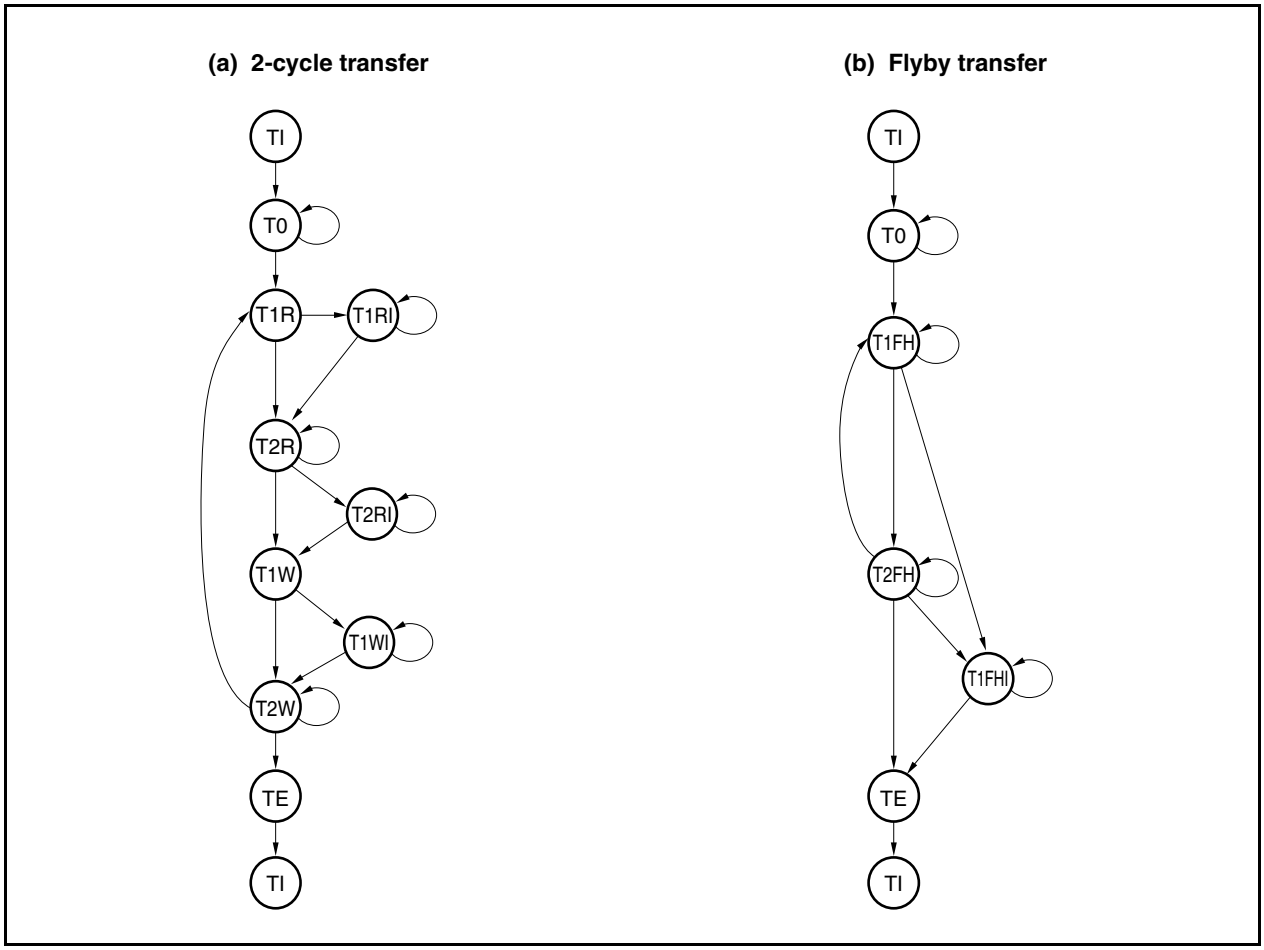
(13) TE state

The TE state corresponds to DMA transfer completion. The DMAC generates the internal DMA transfer completion signal (TCn) and various internal signals are initialized (n = 0 to 3). After entering the TE state, the bus invariably enters the TI state.

6.4.2 DMAC bus cycle state transition

Except for the block transfer mode, each time the processing for a DMA transfer is completed, the bus mastership is released.

Figure 6-1. DMAC Bus Cycle State Transition



6.5 Transfer Modes

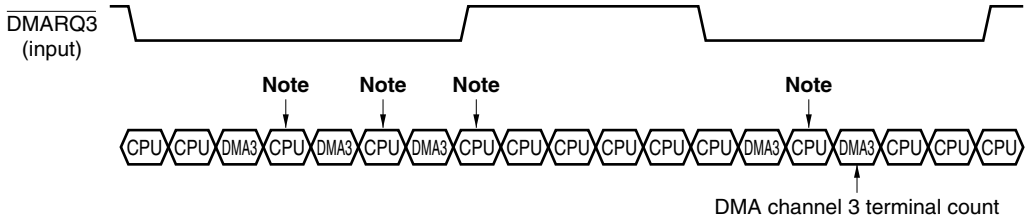
6.5.1 Single transfer mode

In single transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again once. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request takes precedence. If a single transfer is executed, the internal DMA request is cleared each time one DMA cycle has been completed. If any other channel requests DMA after completion of one DMA cycle, therefore, the DMA transfer request with the highest priority is selected from the channels other than the one for which the DMA cycle has just been completed.

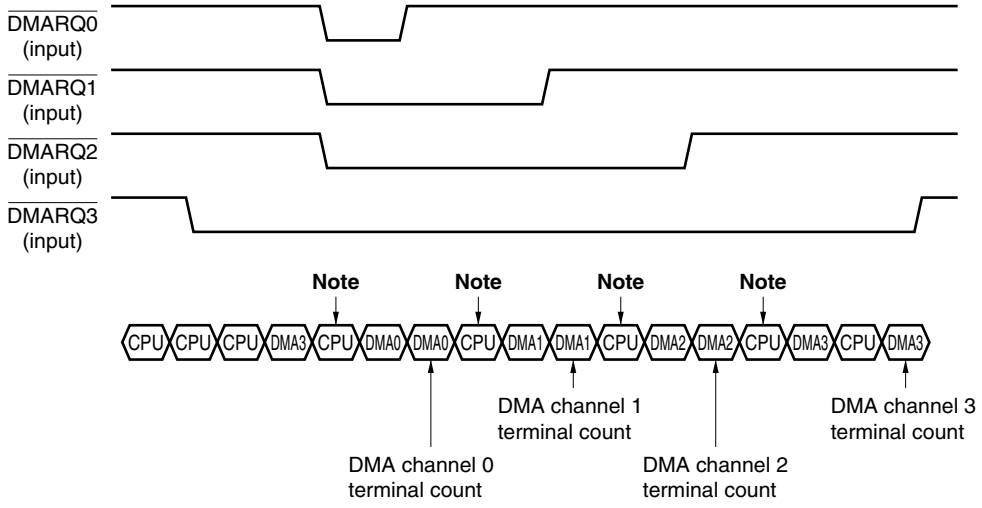
The following shows an example of a single transfer. Figure 6-3 shows an example of a single transfer in which a higher priority DMA request is issued. DMA channels 0 to 2 are in the block transfer mode and channel 3 is in the single transfer mode.

Figure 6-2. Single Transfer Example 1



Note The bus is always released.

Figure 6-3. Single Transfer Example 2



Note The bus is always released.

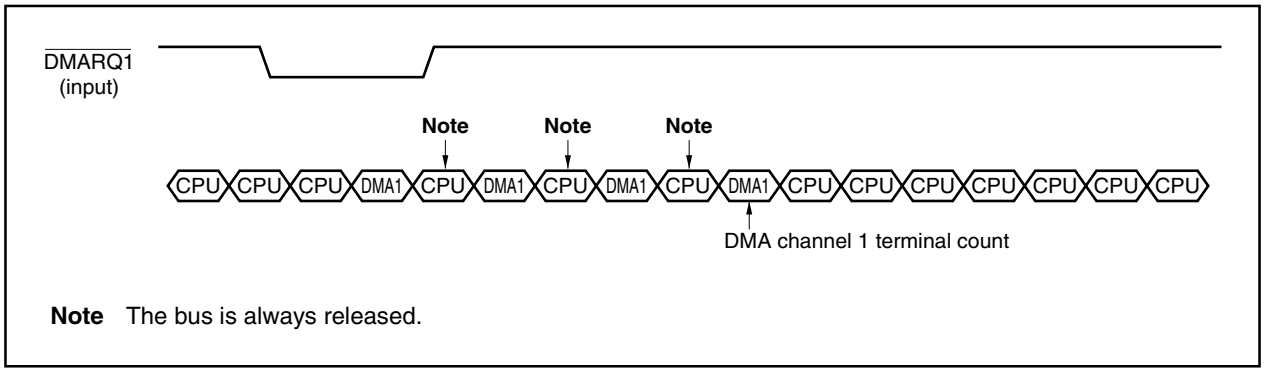
6.5.2 Single-step transfer mode

In single-step transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request signal ($\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$), transfer is performed again. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

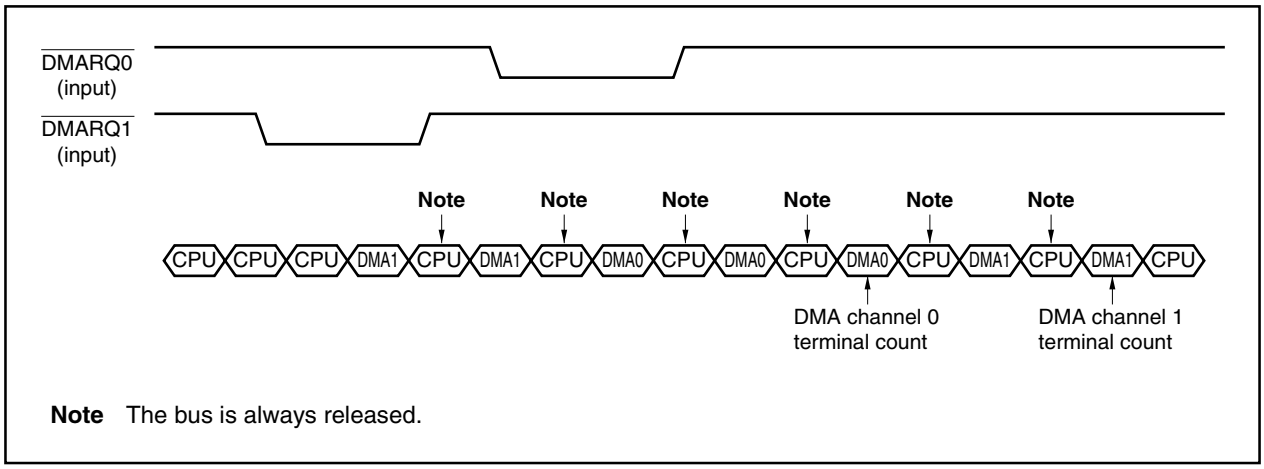
The following shows an example of a single-step transfer. Figure 6-5 shows an example of single-step transfer made in which a higher priority DMA request is issued. DMA channels 0 and 1 are in the single-step transfer mode.

Figure 6-4. Single-Step Transfer Example 1



Note The bus is always released.

Figure 6-5. Single-Step Transfer Example 2



Note The bus is always released.

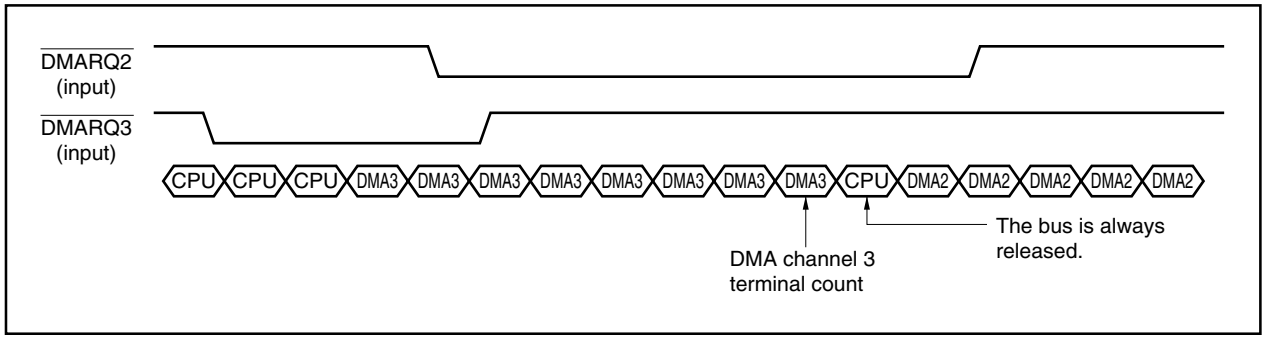
6.5.3 Block transfer mode

In the block transfer mode, once transfer starts, the DMAC continues the transfer operation without releasing the bus until a terminal count occurs. No other DMA requests are acknowledged during block transfer.

After the block transfer ends and the DMAC releases the bus, another DMA transfer can be acknowledged.

The following shows an example of block transfer in which a higher priority DMA request is issued. DMA channels 2 and 3 are in the block transfer mode.

Figure 6-6. Block Transfer Example



6.6 Transfer Types

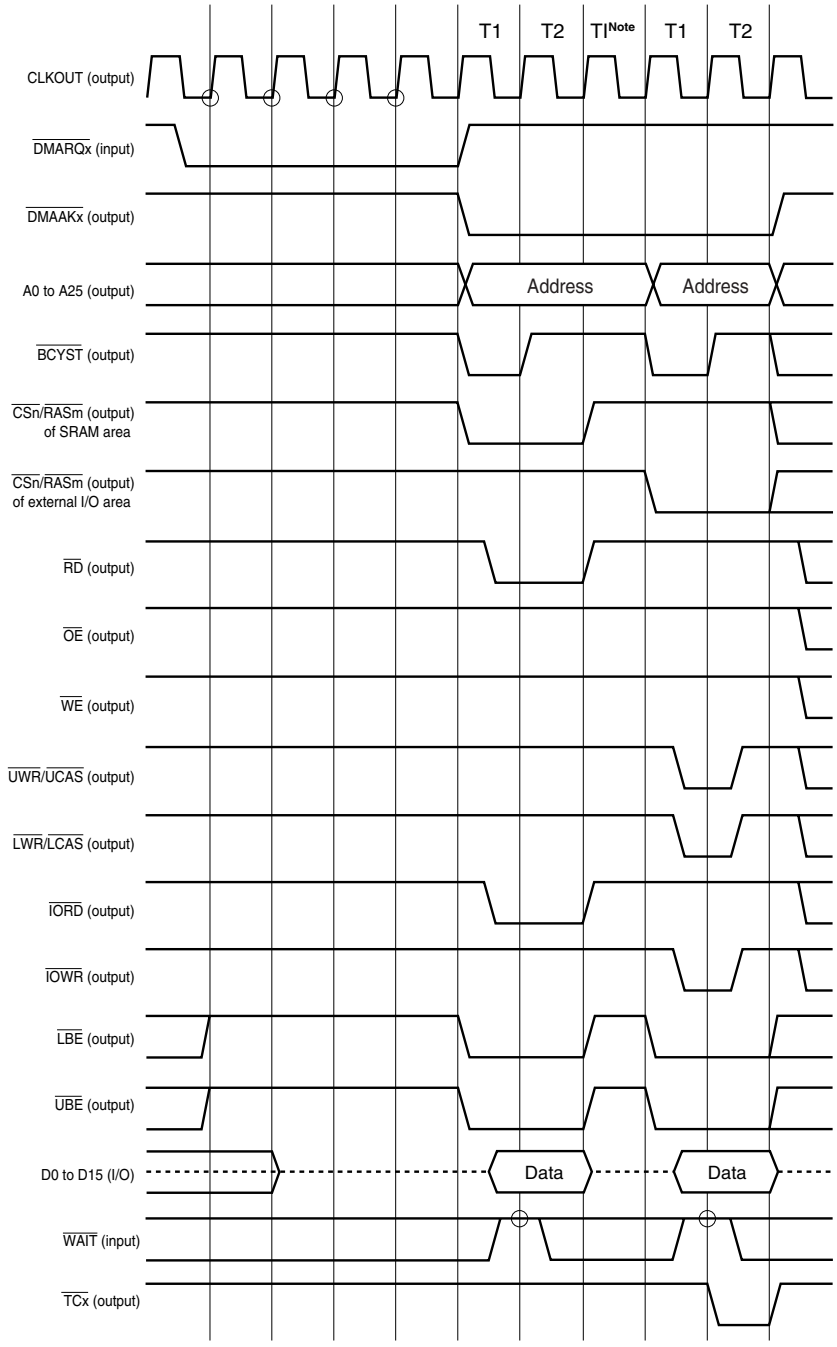
6.6.1 2-cycle transfer

In 2-cycle transfer, data transfer is performed in two cycles, a read cycle (source to DMAC) and a write cycle (DMAC to destination).

In the first cycle, the source address is output and reading is performed from the source to the DMAC. In the second cycle, the destination address is output and writing is performed from the DMAC to the destination.

Figure 6-7. Timing of Access to SRAM, External ROM, and External I/O During 2-Cycle DMA Transfer (1/2)

(a) SRAM → External I/O (BCC register setting for SRAM: BCn1, BCn0 = 00B)
(BCC register setting for external I/O: BCn1, BCn0 = 00B)



Note This idle state (TI) is independent of the BCC register setting.

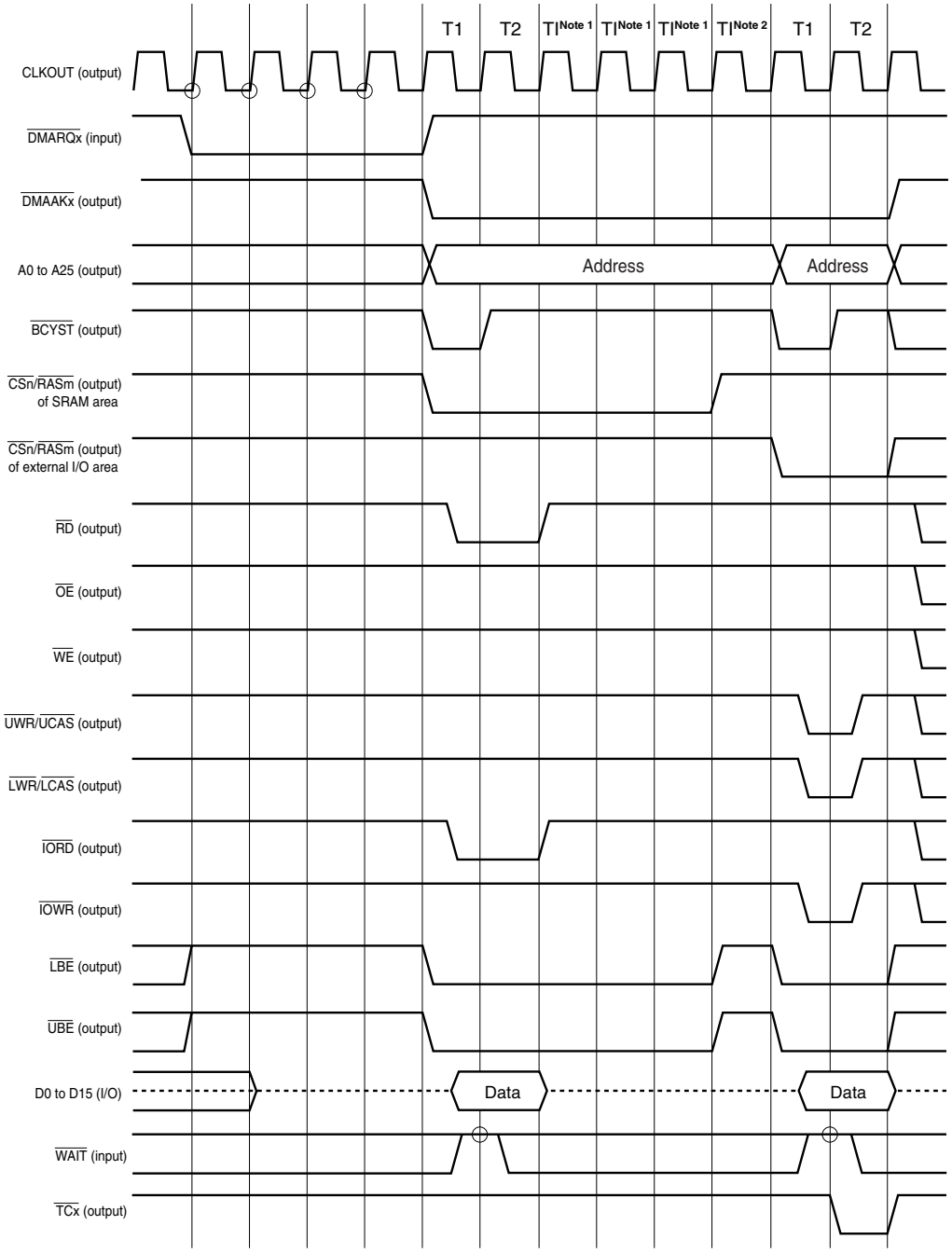
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-7. Timing of Access to SRAM, External ROM, and External I/O During 2-Cycle DMA Transfer (2/2)

(b) SRAM → External I/O (BCC register setting for SRAM: BCn1, BCn0 = 11B)
(BCC register setting for external I/O: BCn1, BCn0 = 00B)



Notes 1. This idle state (TI) is inserted by means of a BCC register setting.

2. This idle state (TI) is independent of the BCC register setting.

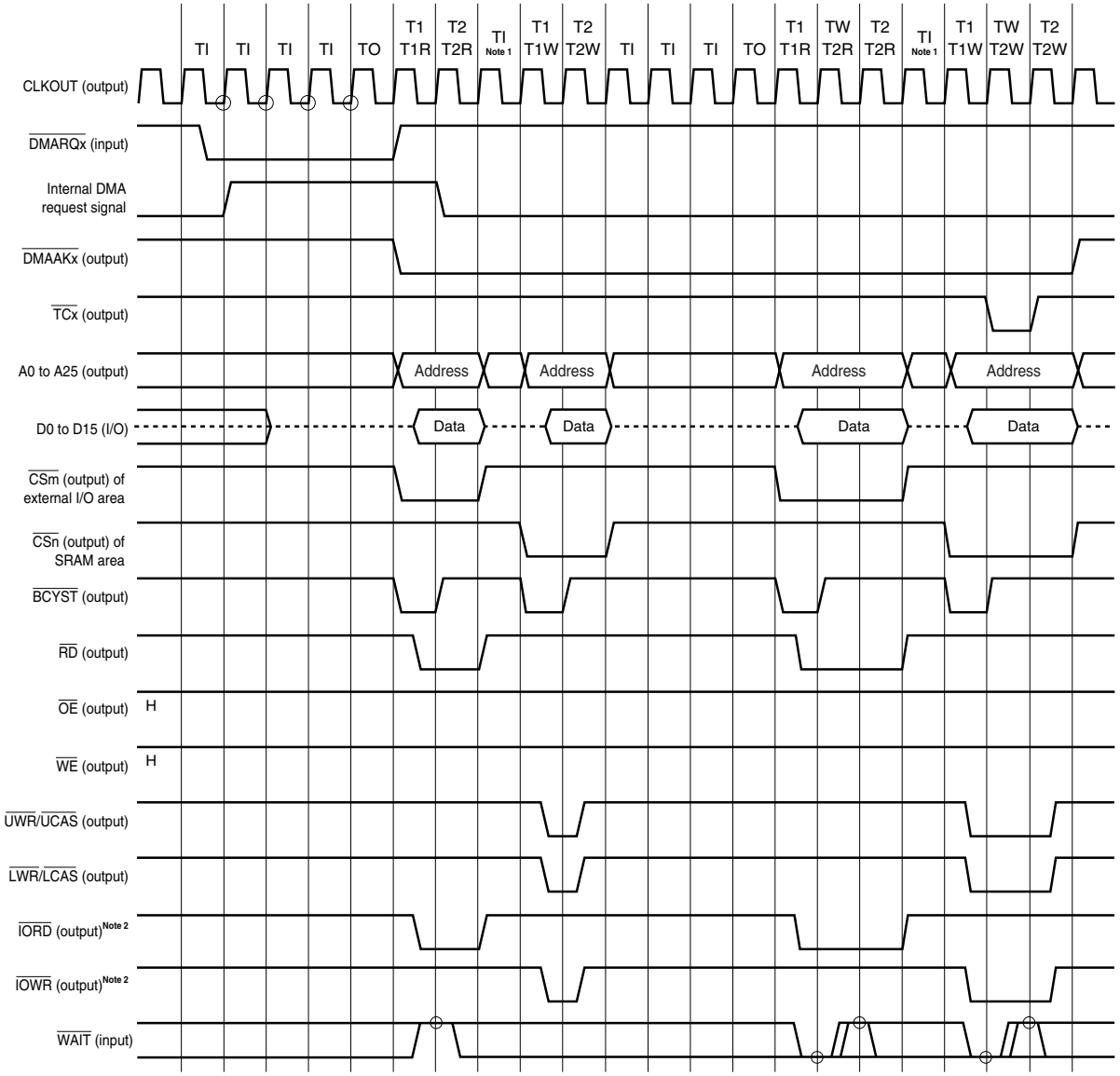
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-8. Timing of 2-Cycle DMA Transfer (External I/O → SRAM)

(a) Single-step transfer mode



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. When the IOEN bit of the BCP register is set to 1.

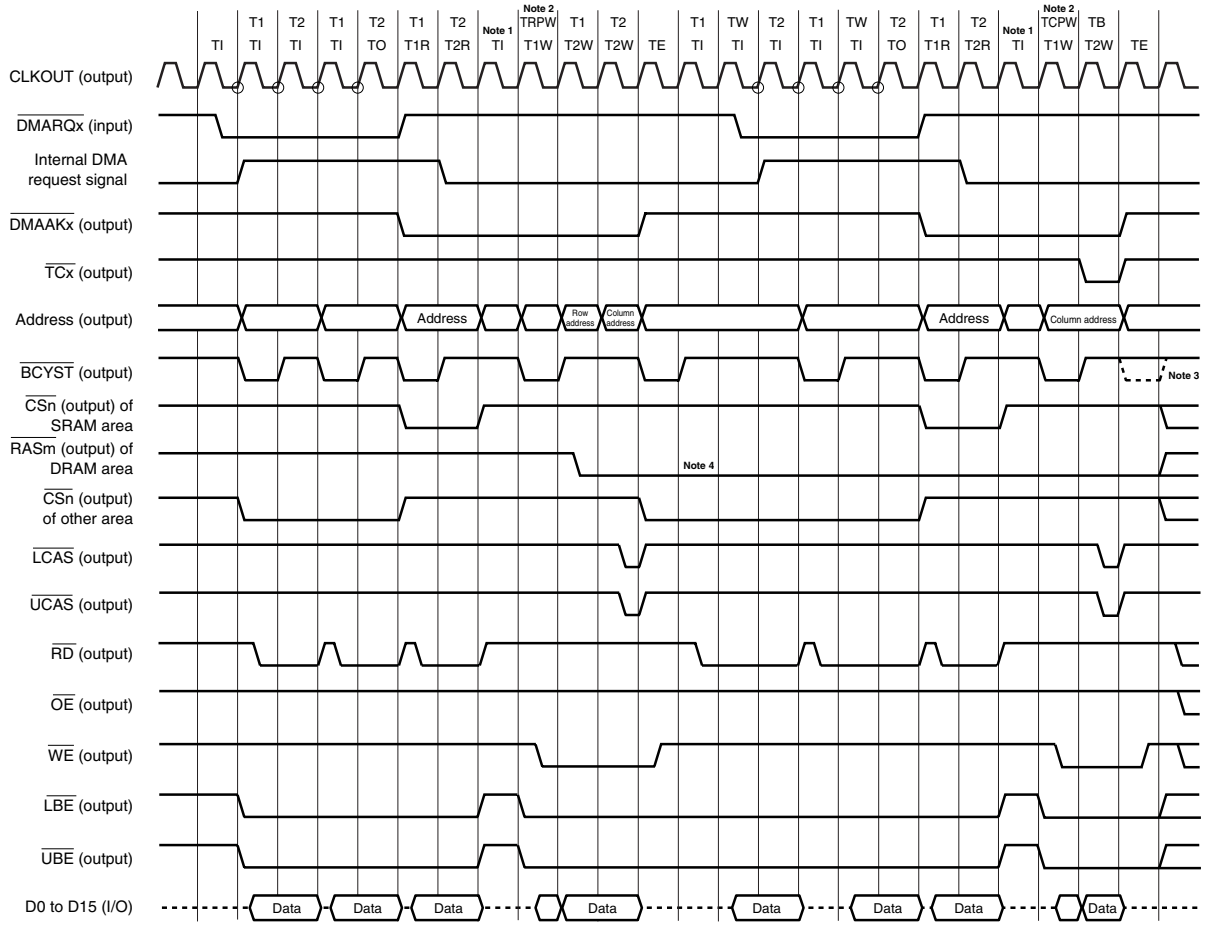
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 0 to 7, x = 0 to 3 (n ≠ m)

Figure 6-9. Timing of 2-Cycle DMA Transfer (SRAM → EDO DRAM) (1/3)

(a) Single transfer mode



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. TRPW and TCPW are always inserted for one or more cycles.

3. When a bus cycle accessing another CS space or a read cycle accessing the same CS space follows this write cycle.

4. In the case of the RAS hold mode

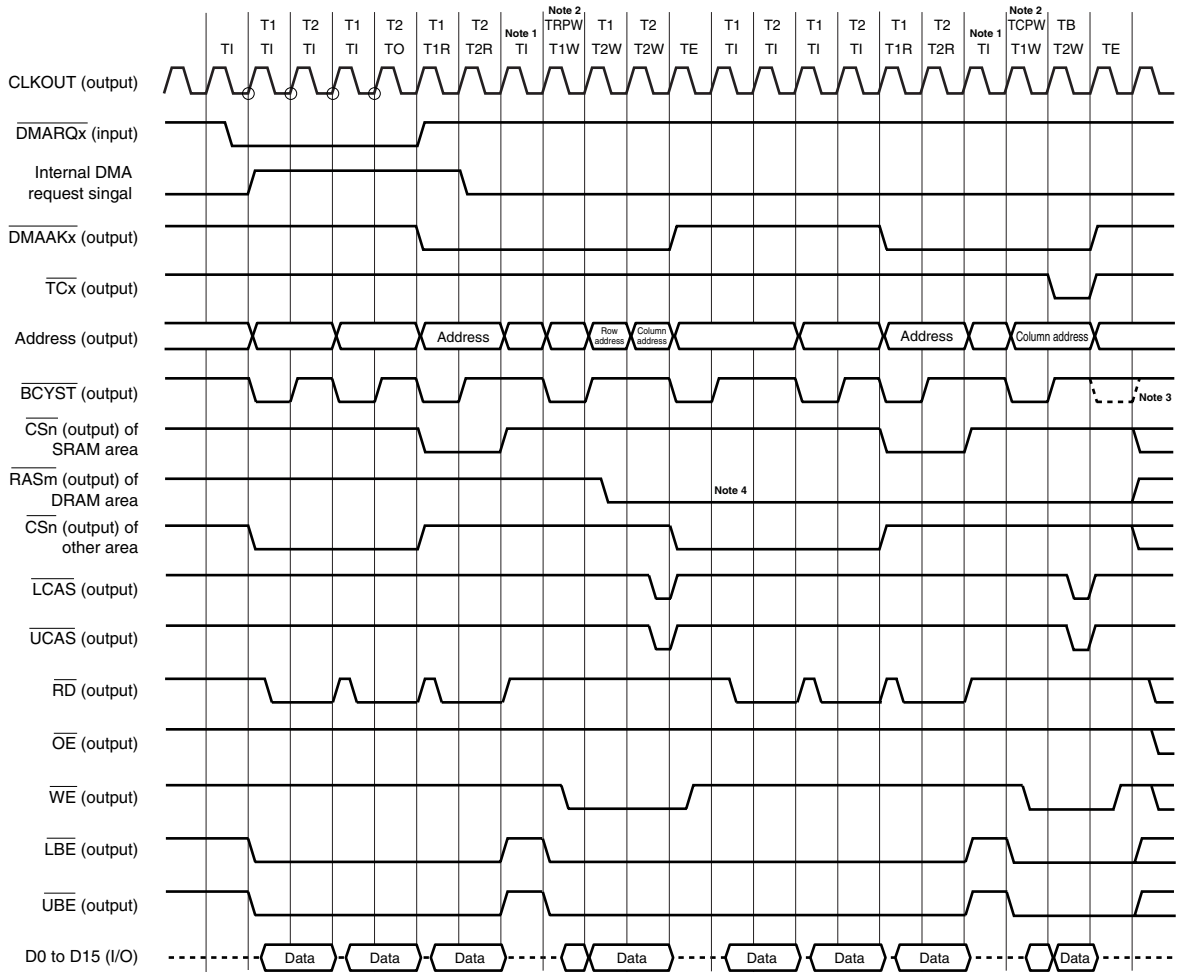
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-9. Timing of 2-Cycle DMA Transfer (SRAM → EDO DRAM) (2/3)

(b) Single-step transfer mode



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. TRPW and TCPW are always inserted for one or more cycles.

3. When a bus cycle accessing another CS space or a read cycle accessing the same CS space follows this write cycle.

4. In the case of the RAS hold mode

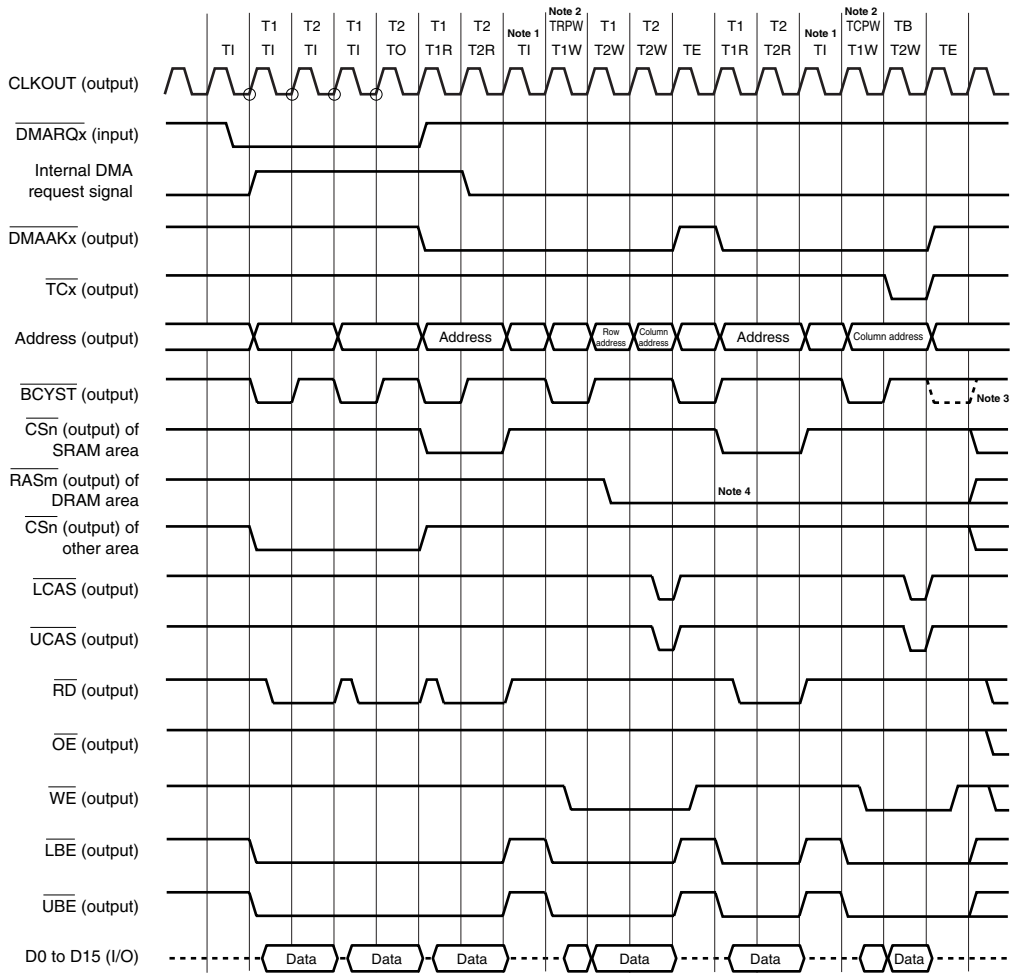
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-9. Timing of 2-Cycle DMA Transfer (SRAM → EDO DRAM) (3/3)

(c) Block transfer mode



Notes 1. This idle state (TI) is independent of the BCC register setting.

2. TRPW and TCPW are always inserted for one or more cycles.

3. When a bus cycle accessing another CS space or a read cycle accessing the same CS space follows this write cycle.

4. In the case of the RAS hold mode

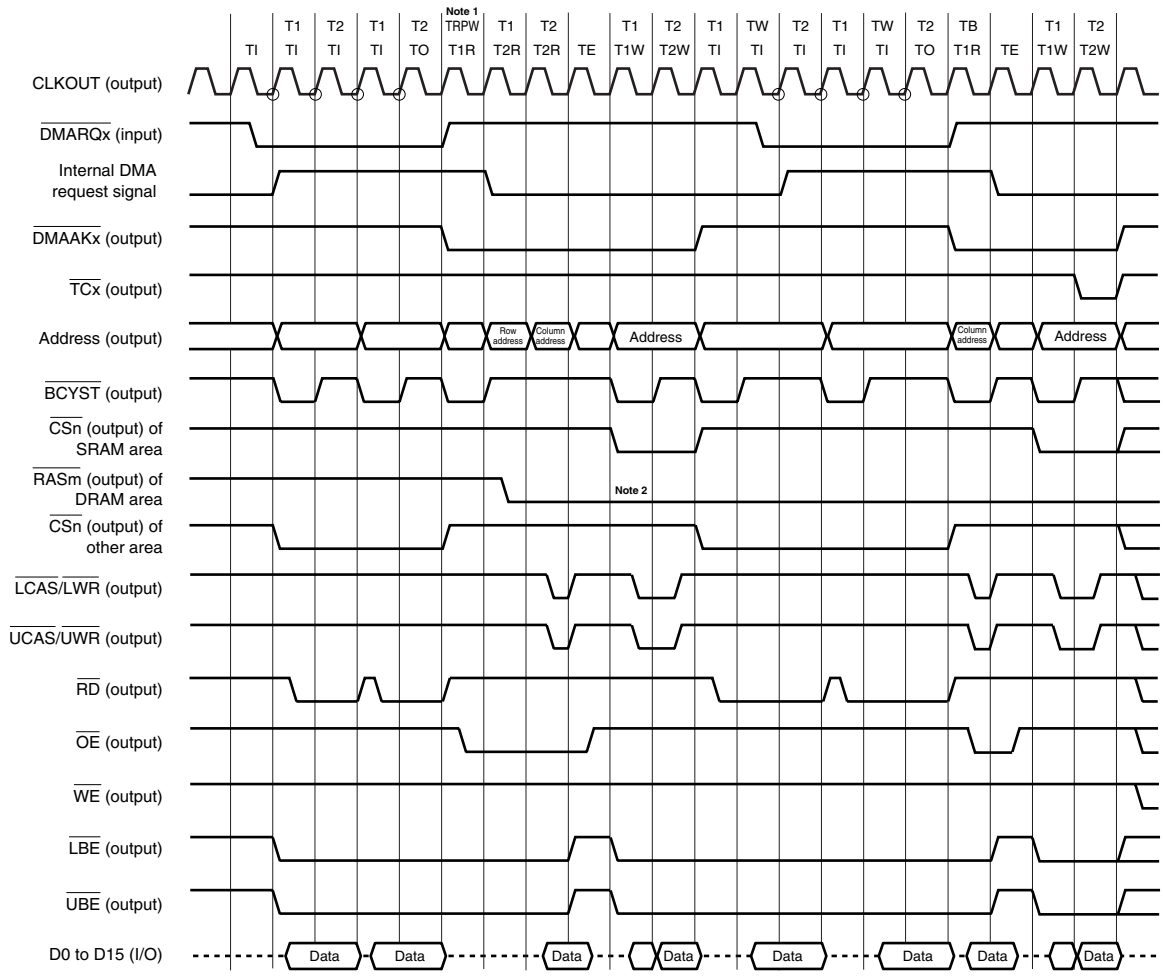
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-10. Timing of 2-Cycle DMA Transfer (EDO DRAM → SRAM) (1/3)

(a) Single transfer mode



Notes 1. TRPW is always inserted for one or more cycles.

2. In the case of the RAS hold mode

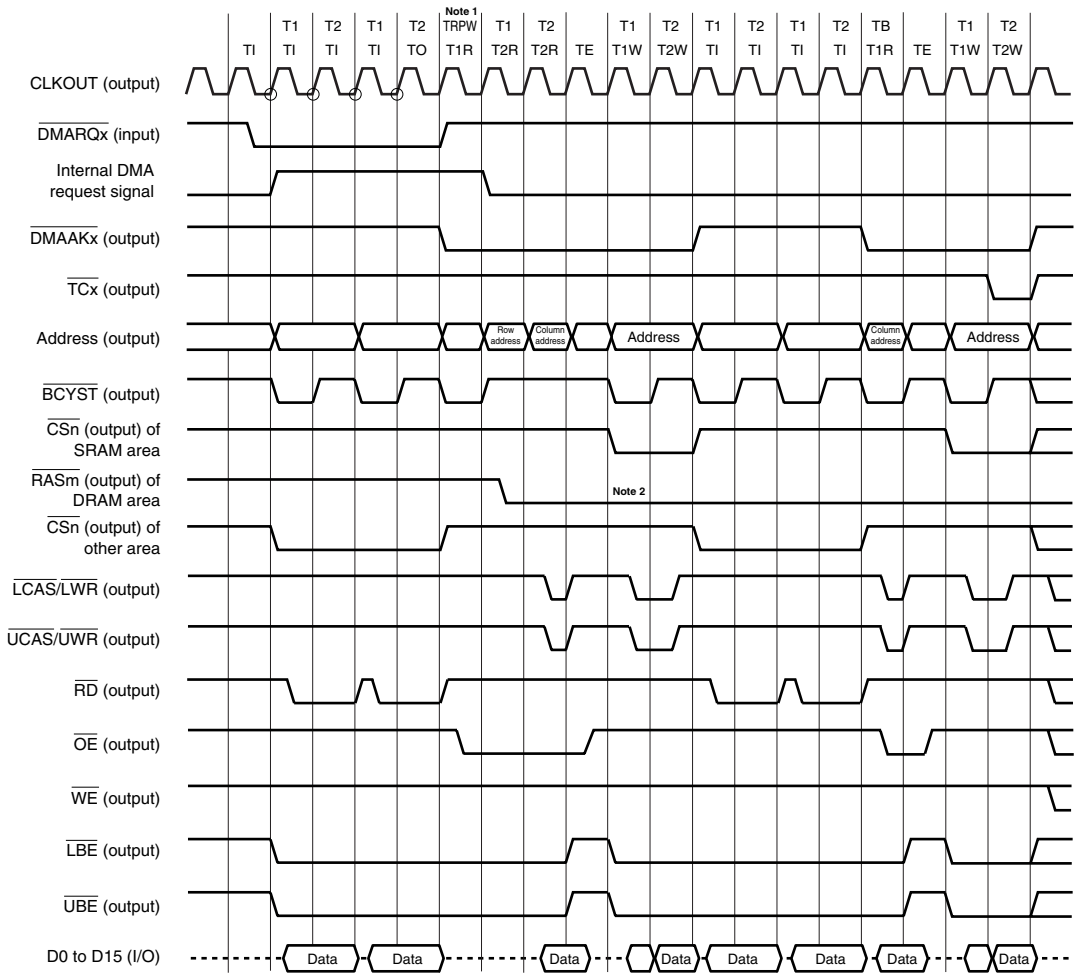
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-10. Timing of 2-Cycle DMA Transfer (EDO DRAM → SRAM) (2/3)

(b) Single-step transfer mode



Notes 1. TRPW is always inserted for one or more cycles.

2. In the case of the RAS hold mode

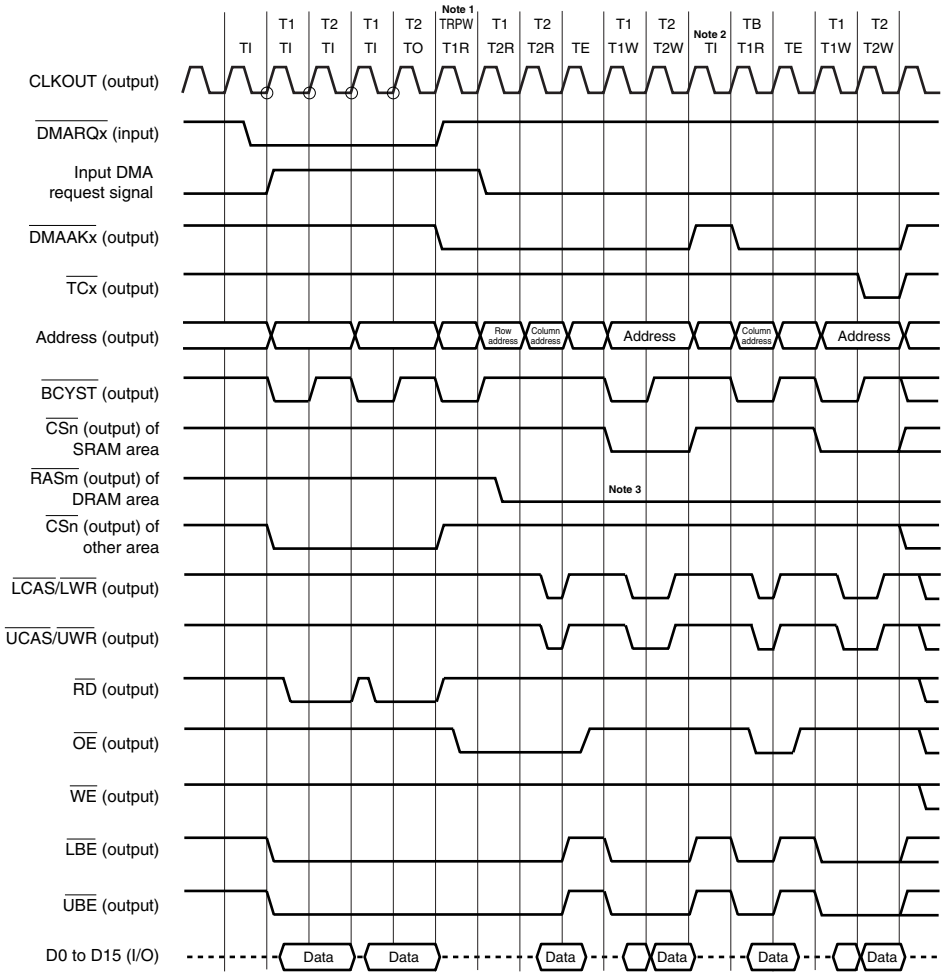
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-10. Timing of 2-Cycle DMA Transfer (EDO DRAM → SRAM) (3/3)

(c) Block transfer mode



Notes 1. TRPW is always inserted for one or more cycles.

2. This idle state (TI) is independent of the BCC register setting.

3. In the case of the RAS hold mode

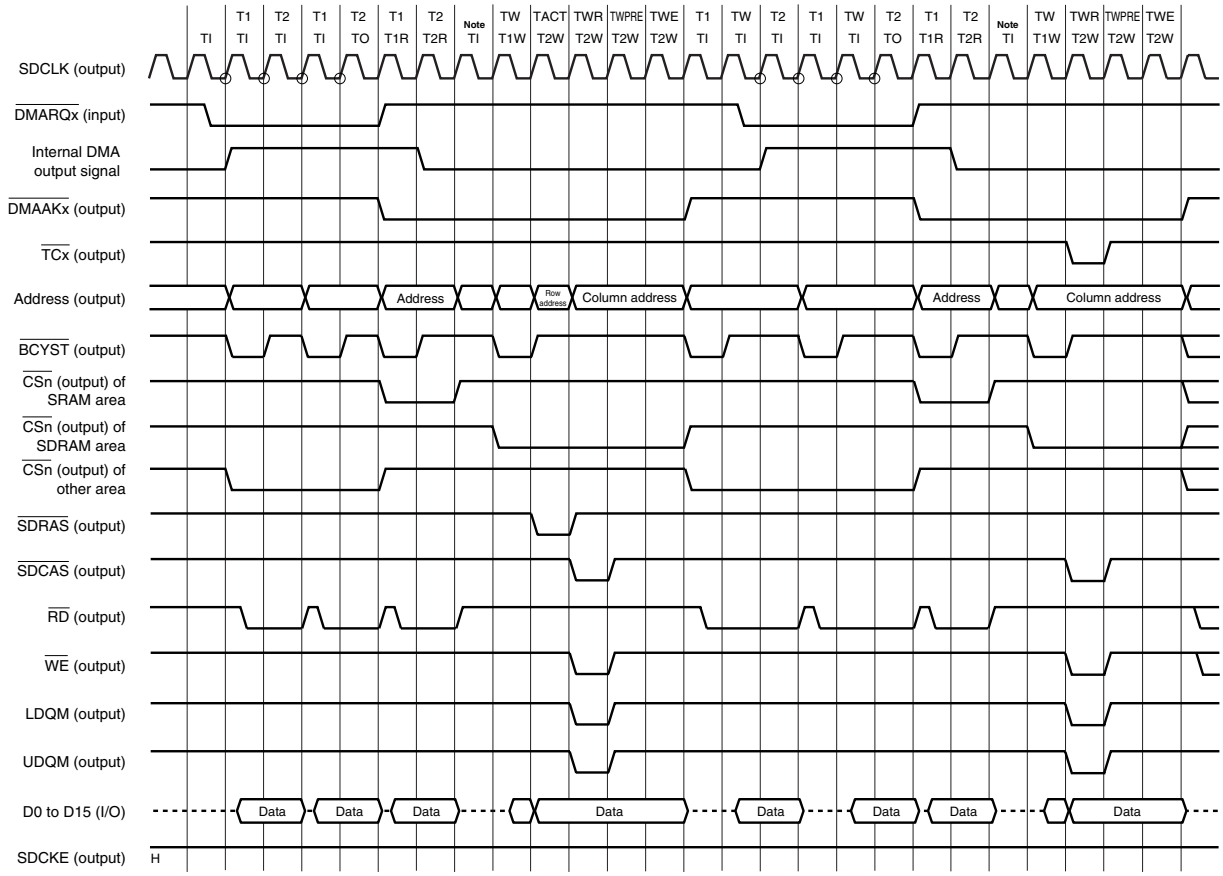
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-11. Timing of 2-Cycle DMA Transfer (SRAM → SDRAM) (1/3)

(a) Single transfer mode



Note This idle state (TI) is independent of the BCC register setting.

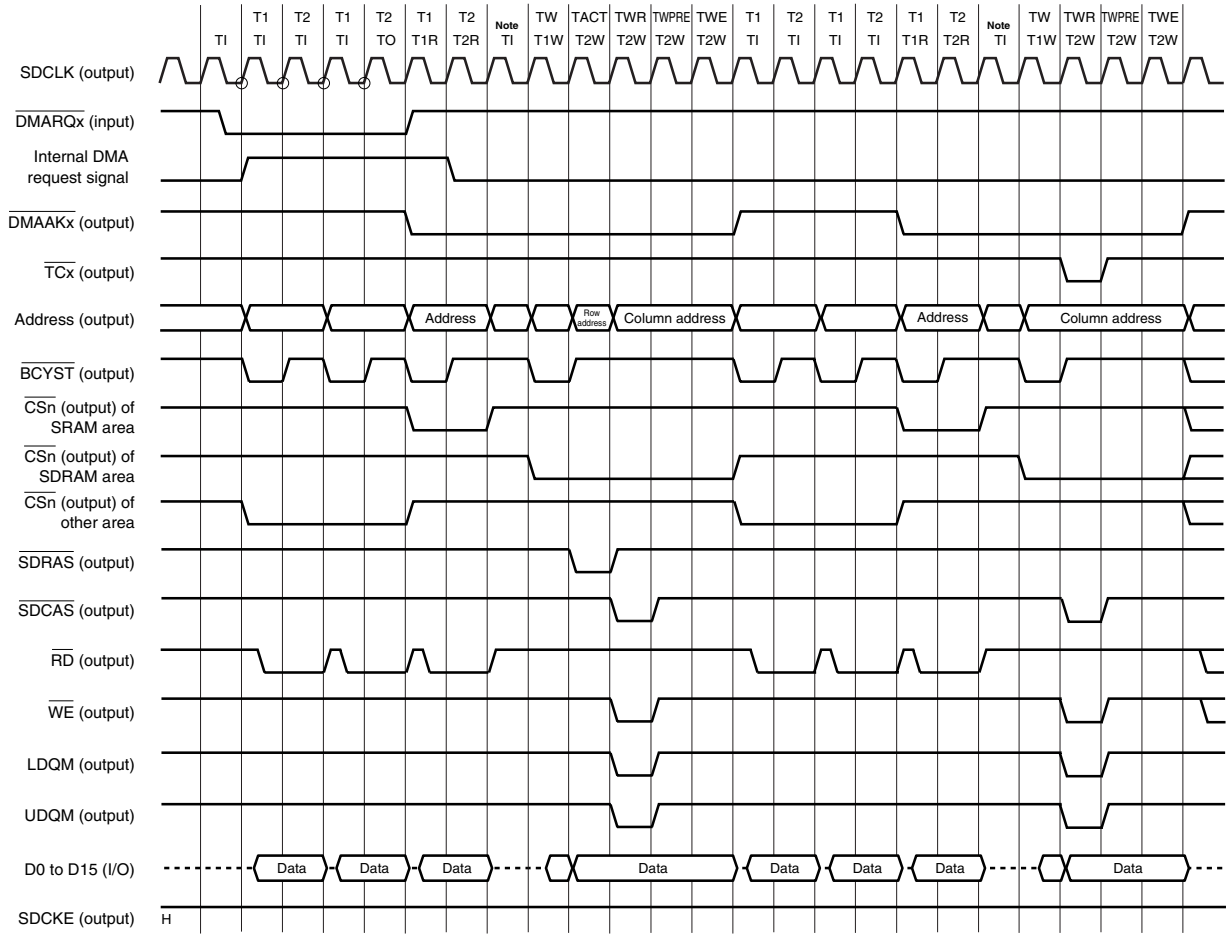
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, x = 0 to 3

Figure 6-11. Timing of 2-Cycle DMA Transfer (SRAM → SDRAM) (2/3)

(b) Single-step transfer mode



Note This idle state (TI) is independent of the BCC register setting.

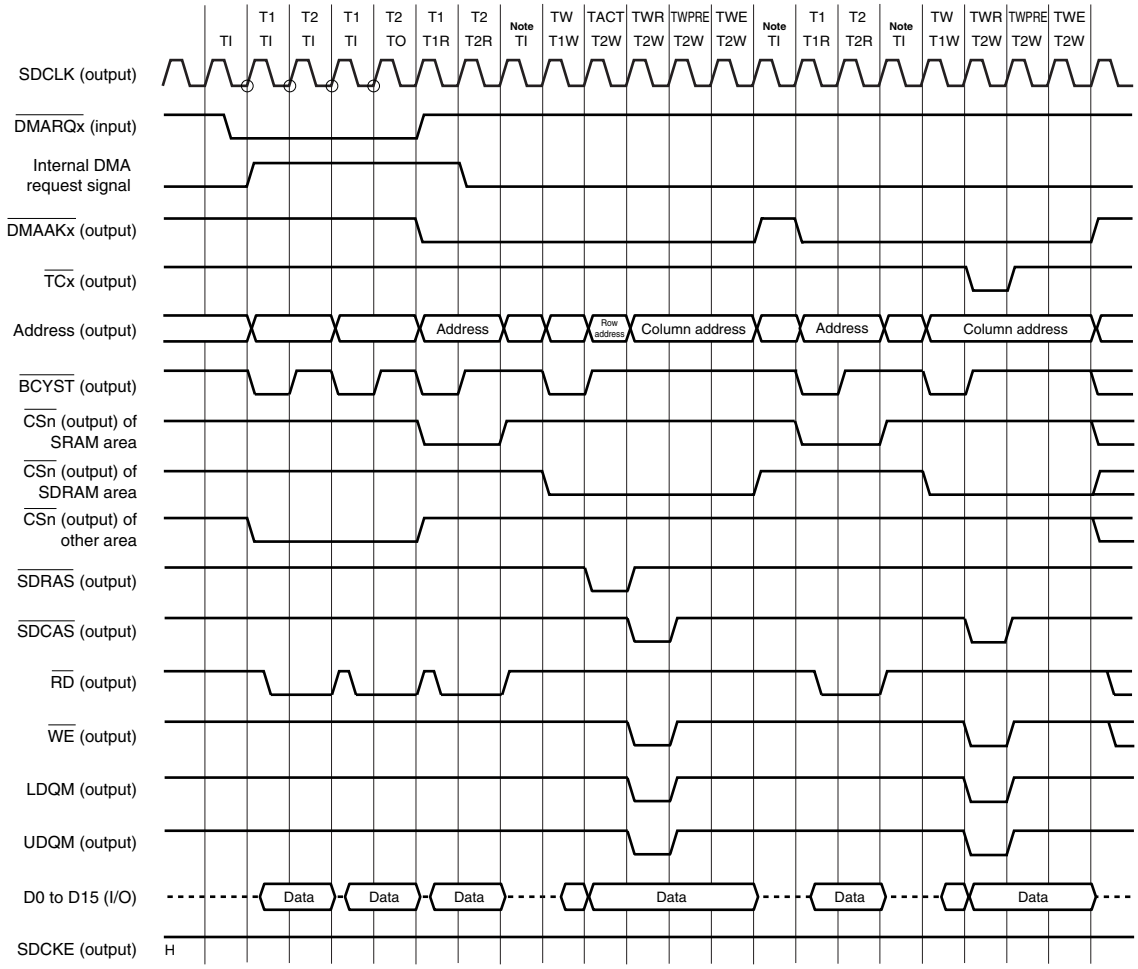
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, x = 0 to 3

Figure 6-11. Timing of 2-Cycle DMA Transfer (SRAM → SDRAM) (3/3)

(c) Block transfer mode



Note This idle state (TI) is independent of the BCC register setting.

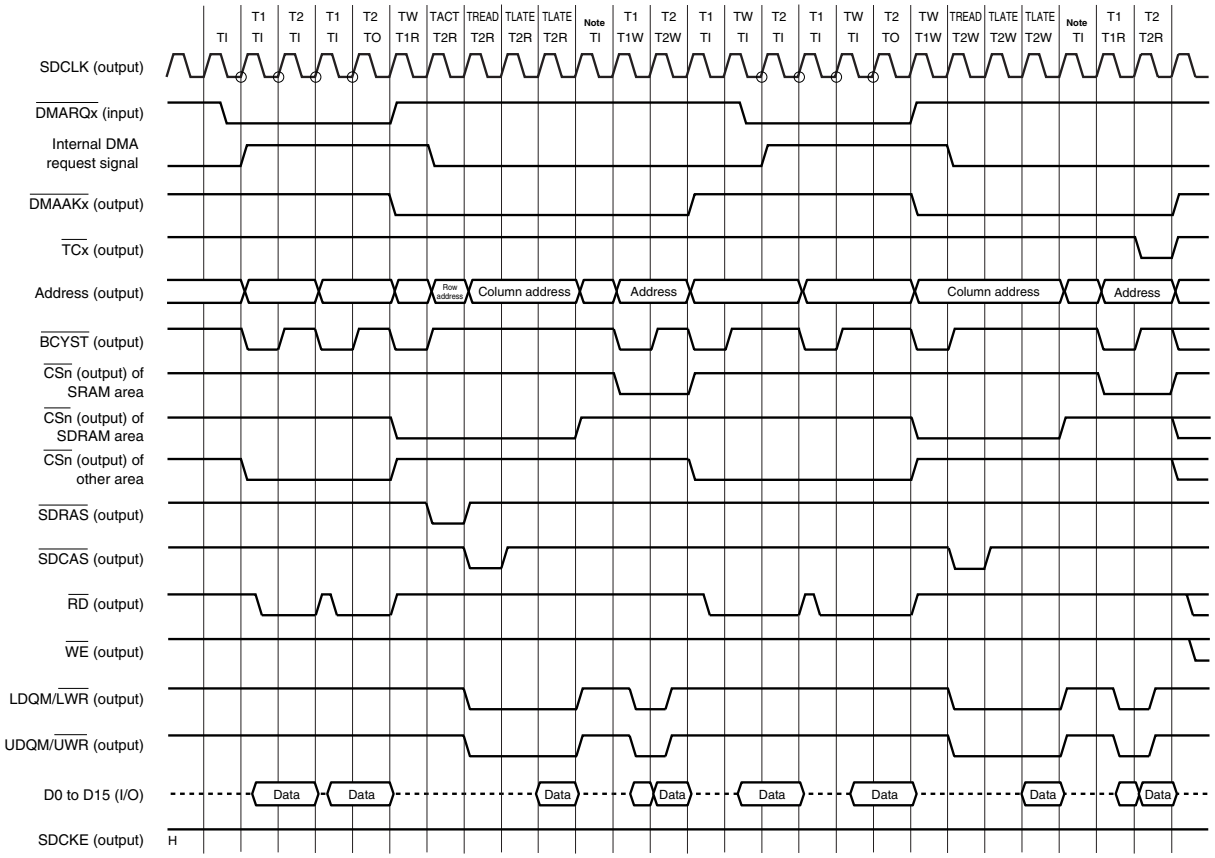
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, x = 0 to 3

Figure 6-12. Timing of 2-Cycle DMA Transfer (SDRAM → SRAM) (1/3)

(a) Single transfer mode



Note This idle state (TI) is independent of the BCC register setting.

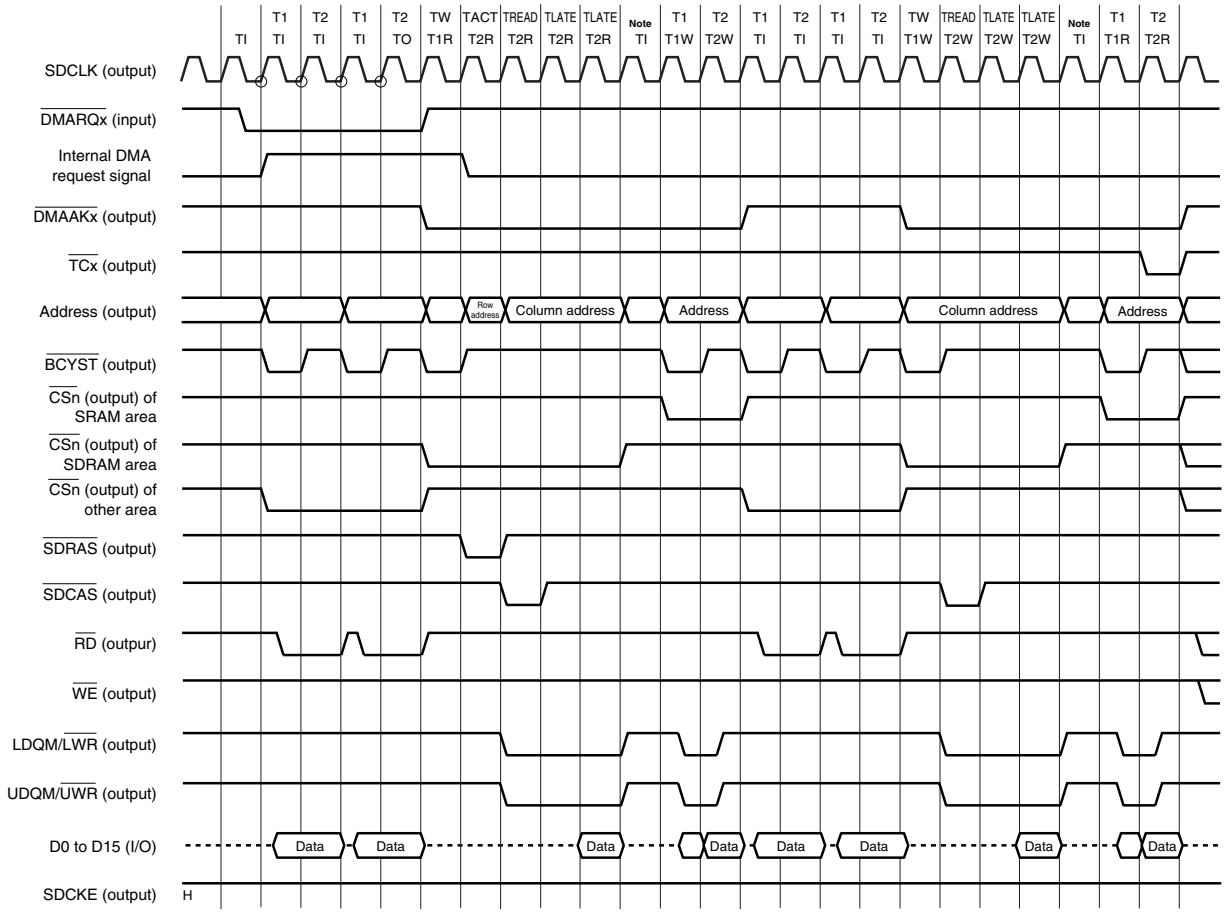
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, x = 0 to 3

Figure 6-12. Timing of 2-Cycle DMA Transfer (SDRAM → SRAM) (2/3)

(b) Single-step transfer mode



Note This idle state (TI) is independent of the BCC register setting.

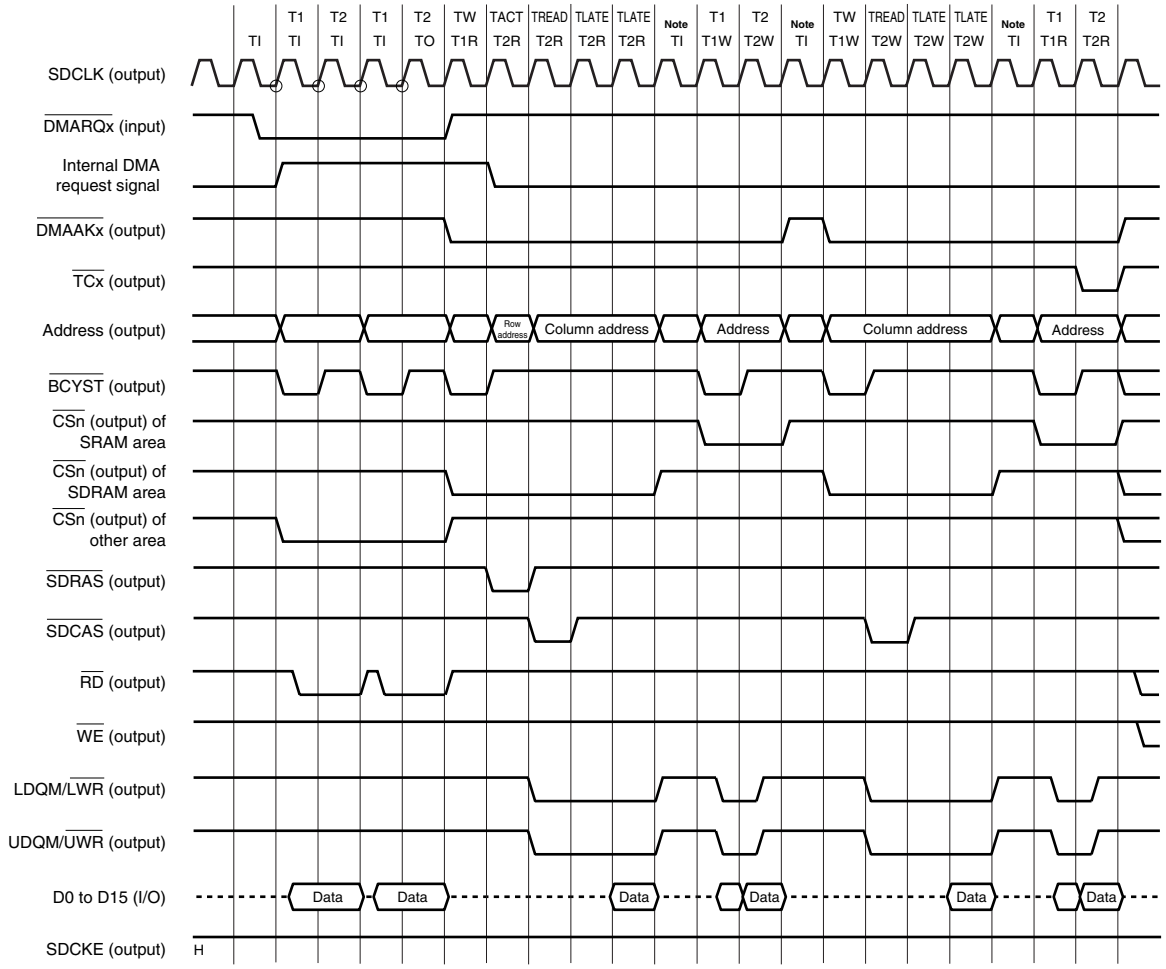
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, x = 0 to 3

Figure 6-12. Timing of 2-Cycle DMA Transfer (SDRAM → SRAM) (3/3)

(c) Block transfer mode



Note This idle state (TI) is independent of the BCC register setting.

Remarks 1. The circle ○ indicates the sampling timing.

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

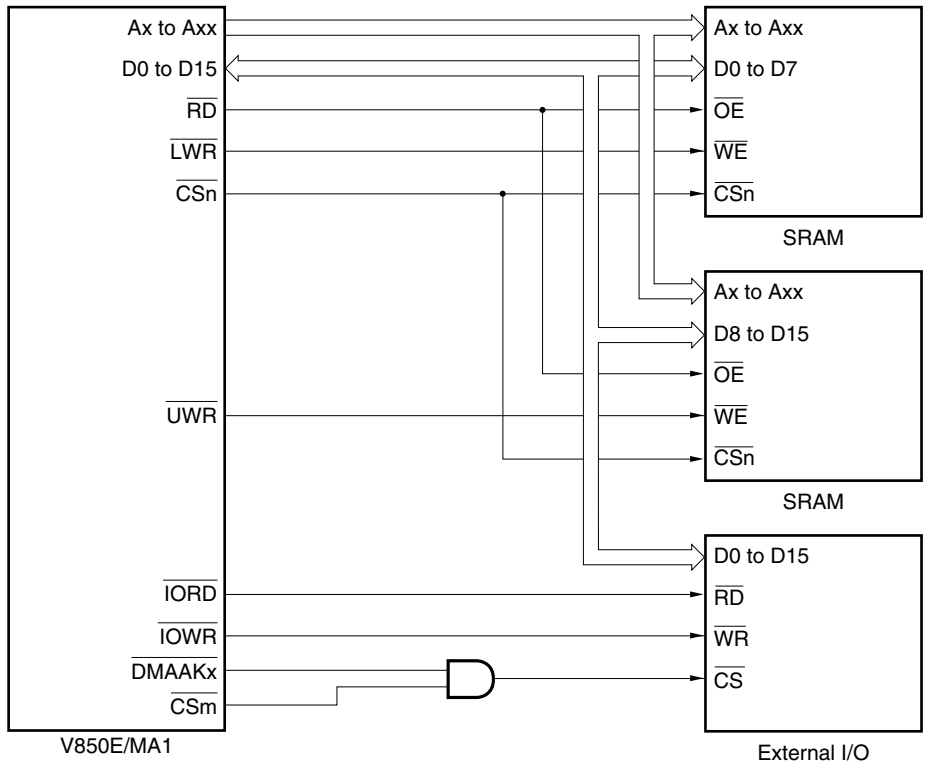
3. n = 0 to 7, x = 0 to 3

6.6.2 Flyby transfer

Since data is transferred in 1 cycle during a flyby transfer, a memory address is always output irrespective whether it is a source address or a destination address, and read/write signals of the memory and peripheral I/O become active at the same time. Therefore, the external I/O is selected by the $\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$ signals.

To perform a normal access to the external I/O by means other than DMA transfer, externally AND the $\overline{\text{CSm}}$ and $\overline{\text{DMAAKx}}$ signals (m = 0 to 7, x = 0 to 3), and connect the resultant signal to the chip select signal of the external I/O. A circuit example of a normal access, other than DMA transfer, to external I/O is shown below.

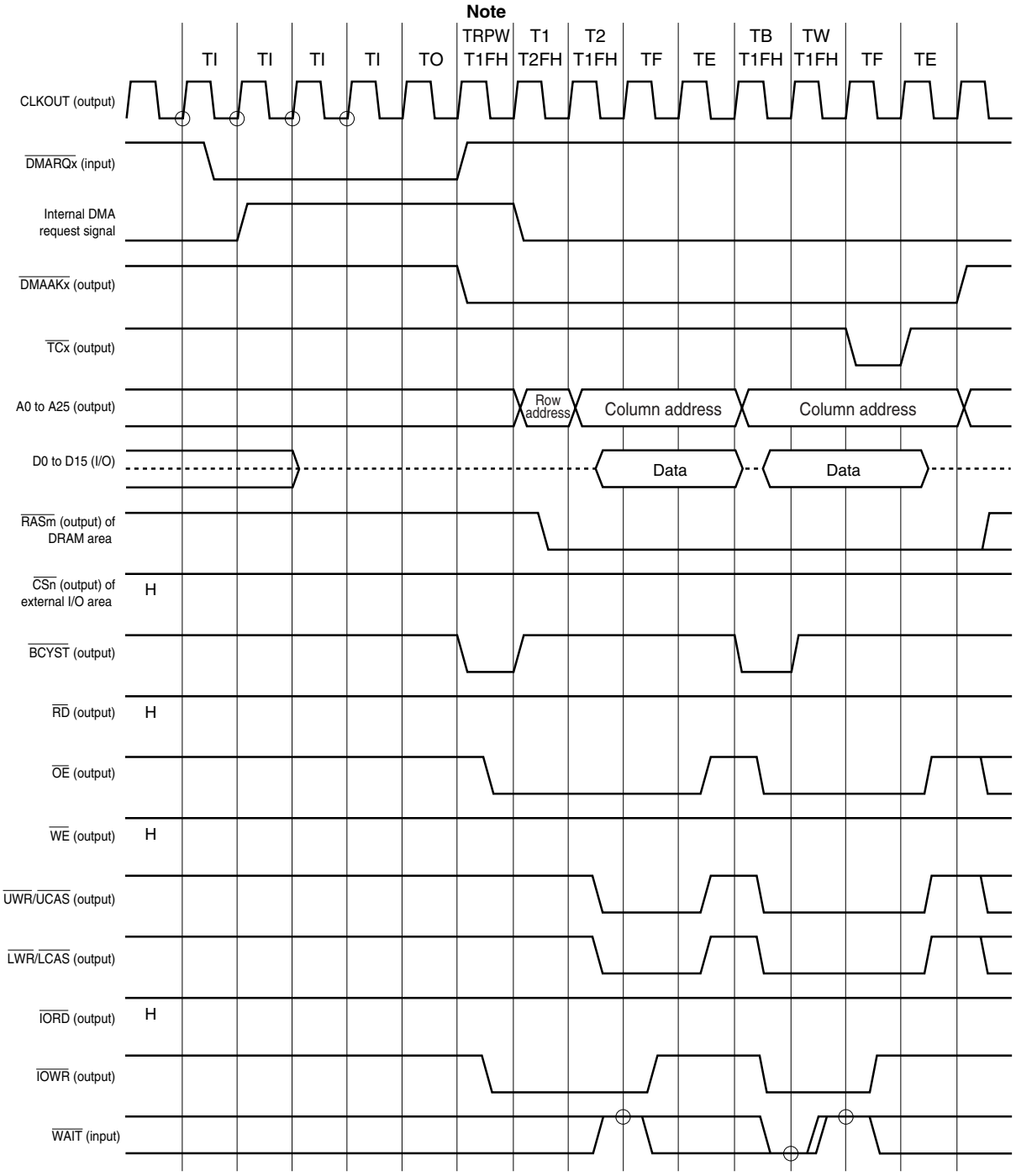
Figure 6-13. Circuit Example When Flyby Transfer Is Performed Between External I/O and SRAM



Remark n = 0 to 7, m = 0 to 7 (n ≠ m)
x = 0 to 3

Figure 6-14. Timing of Flyby Transfer (DRAM → External I/O) (1/3)

(a) Block transfer mode



Note TRPW is always inserted for one or more cycles.

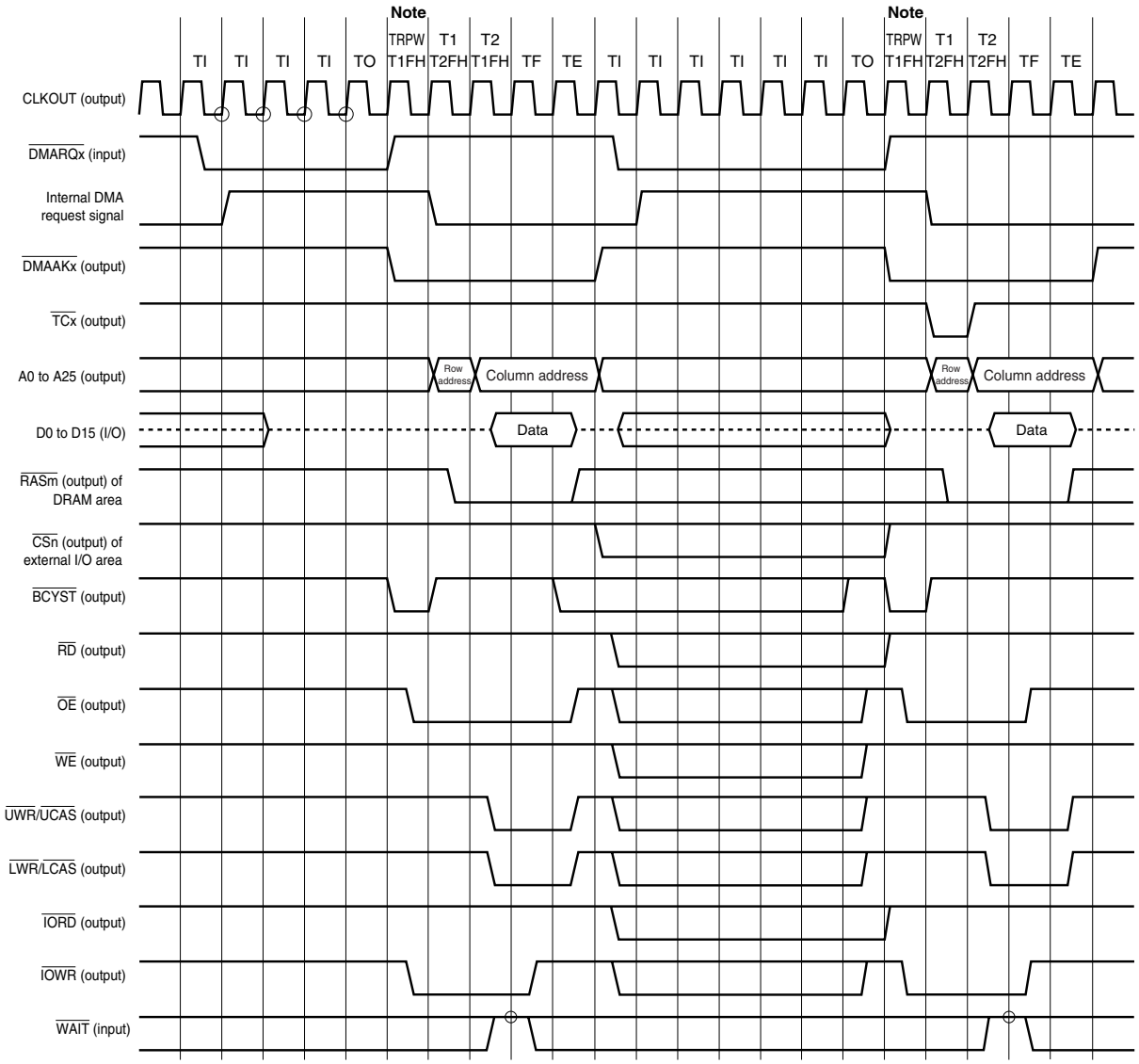
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3 (n ≠ m)

Figure 6-14. Timing of Flyby Transfer (DRAM → External I/O) (2/3)

(b) Single transfer mode



Note TRPW is always inserted for one or more cycles.

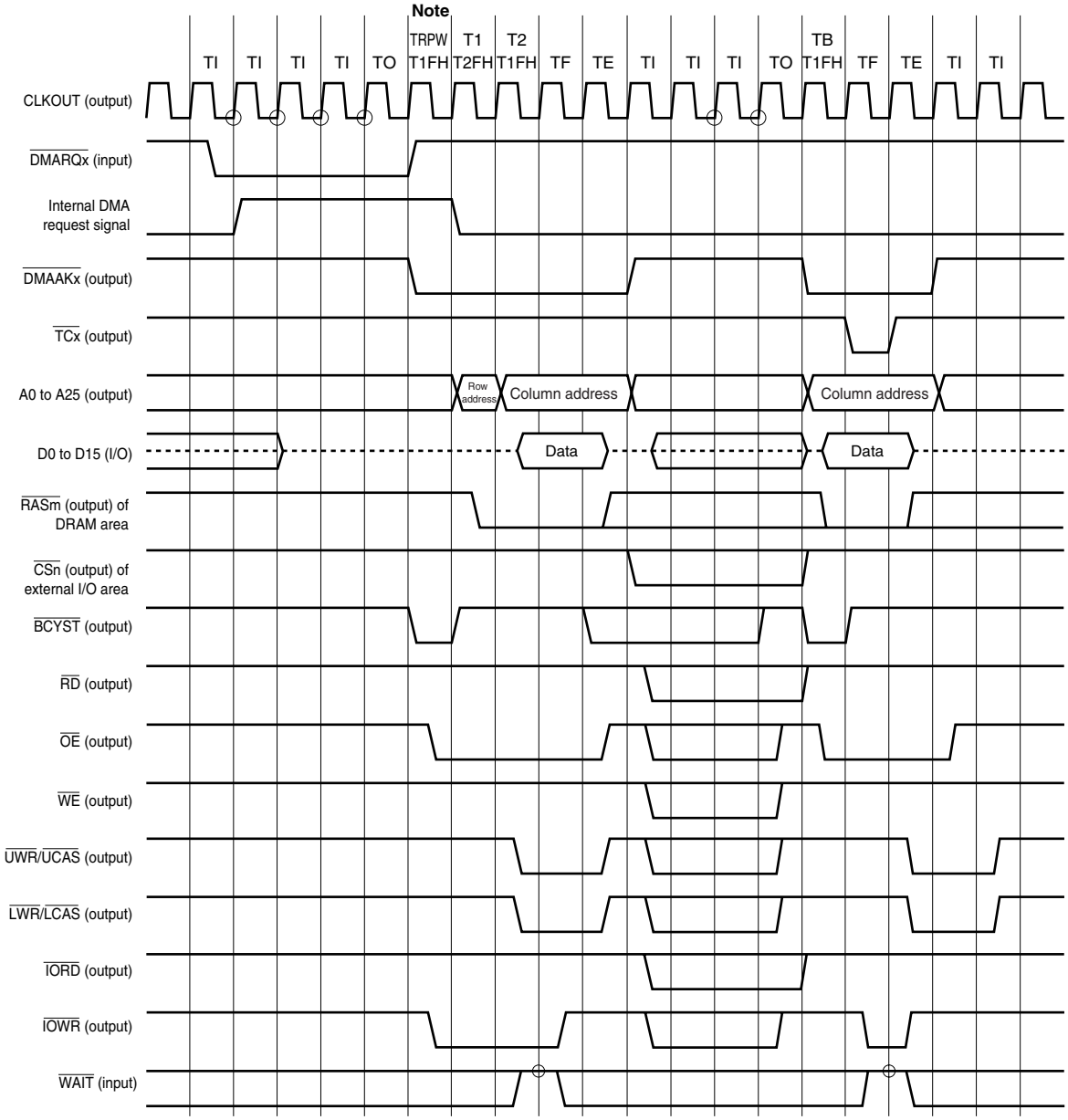
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3 (n ≠ m)

Figure 6-14. Timing of Flyby Transfer (DRAM → External I/O) (3/3)

(c) Single-step transfer mode



Note TRPW is always inserted for one or more cycles.

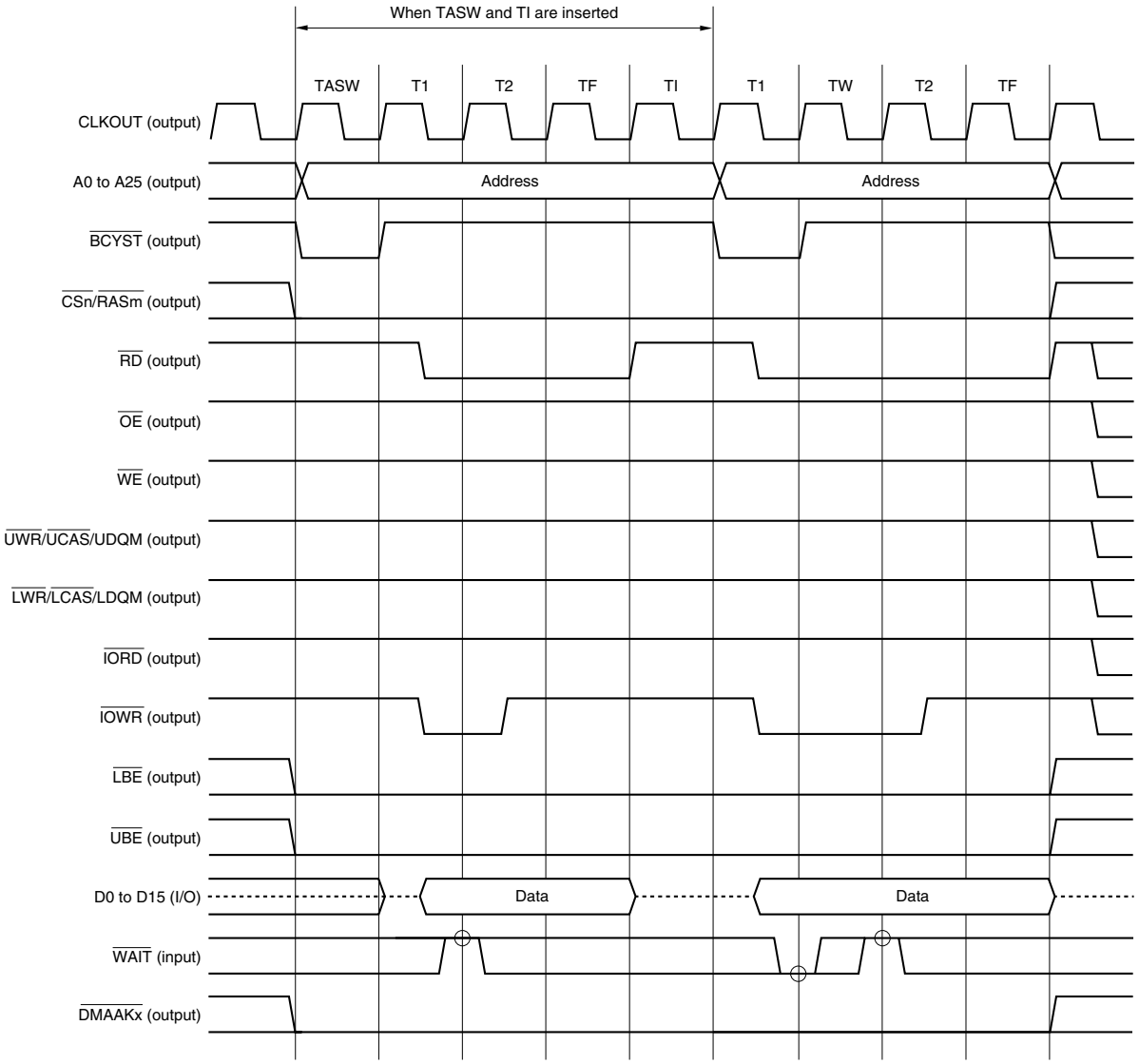
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3 (n ≠ m)

Figure 6-15. Timing of Access to SRAM, External ROM, and External I/O During DMA Flyby Transfer (1/2)

(a) SRAM → external I/O



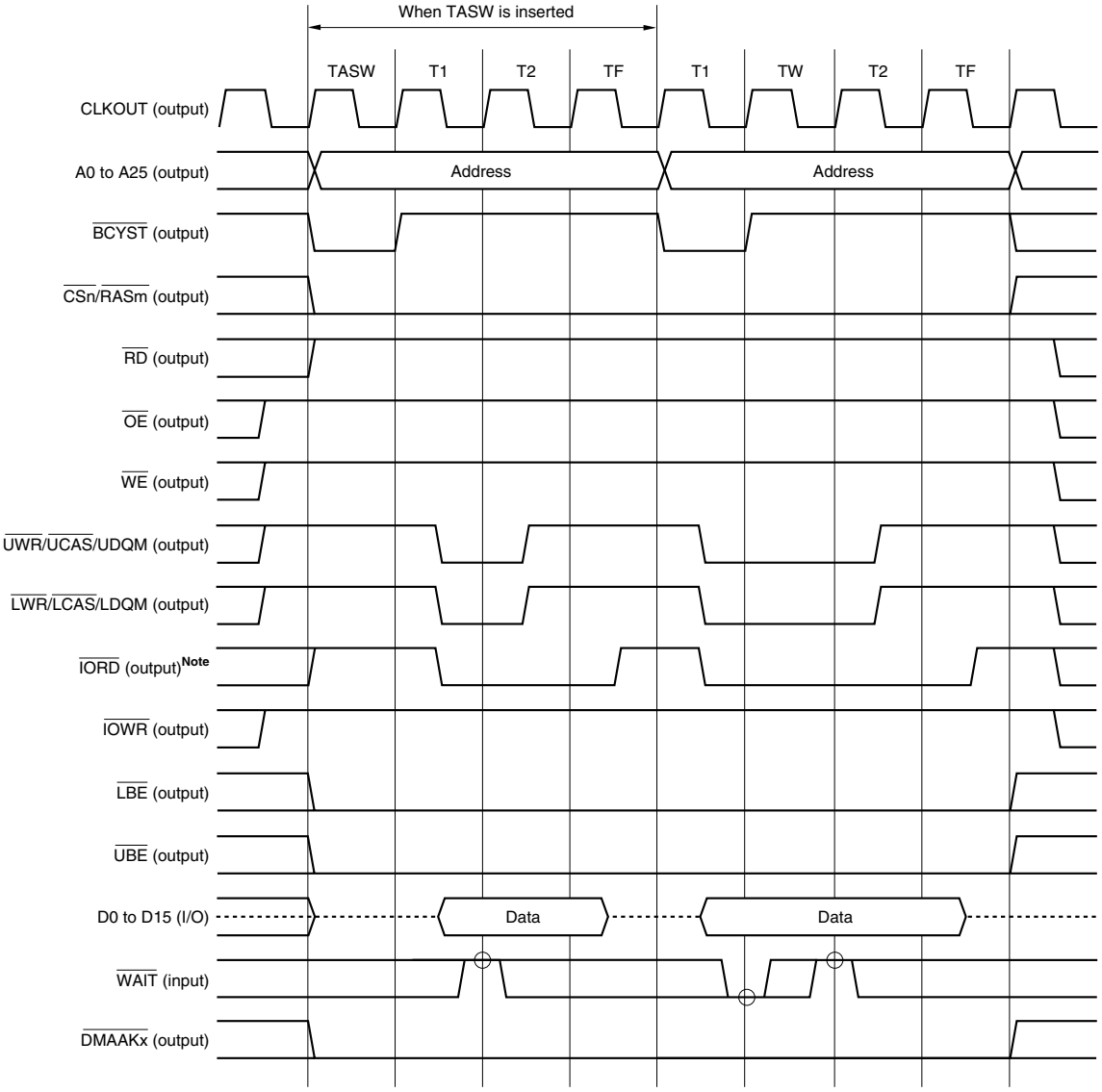
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-15. Timing of Access to SRAM, External ROM, and External I/O During DMA Flyby Transfer (2/2)

(b) External I/O → SRAM



Note During DMA flyby transfer, the rise timing of this read cycle is different from that of other transfer operations.

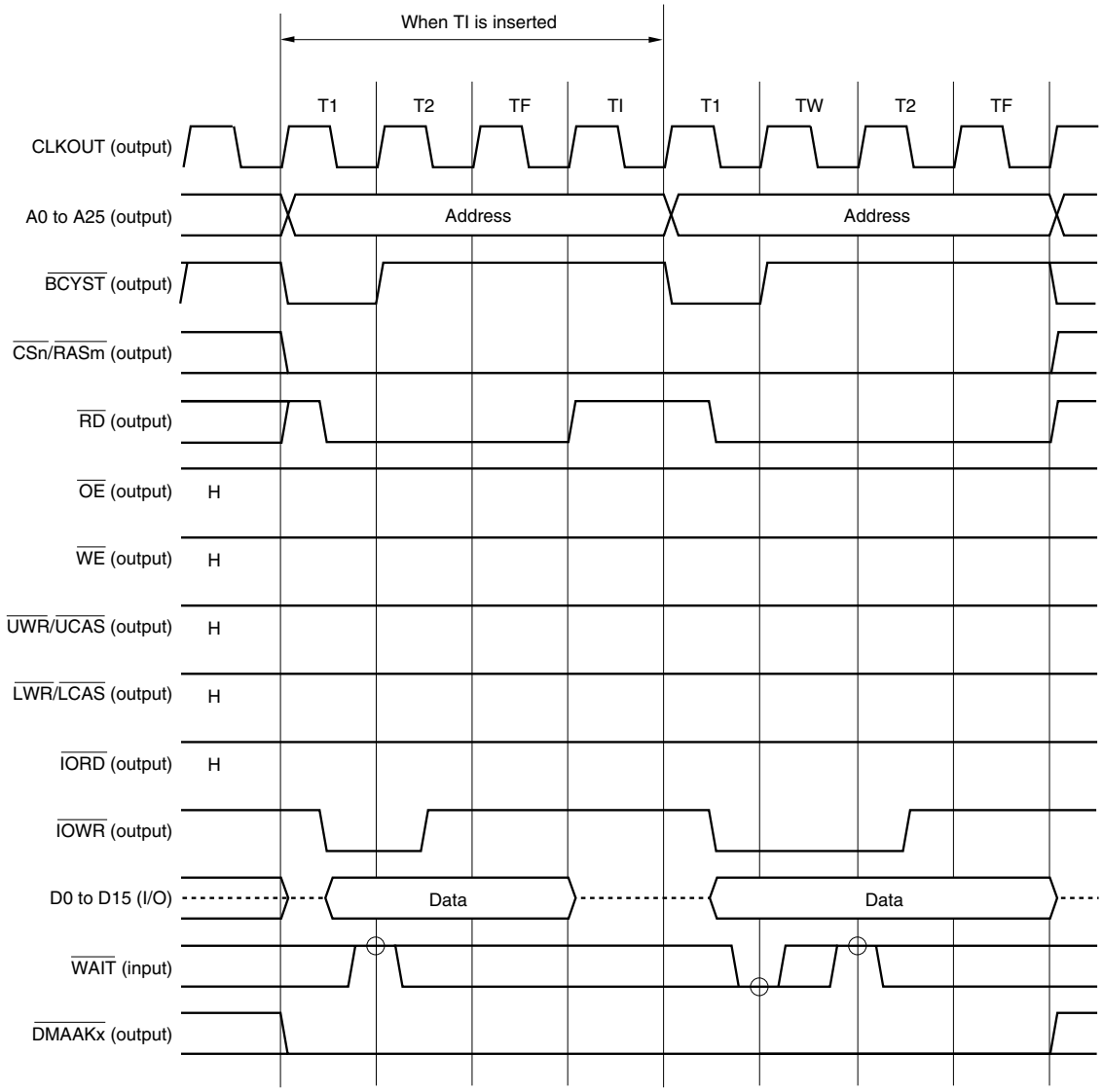
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-16. Page ROM Access Timing During DMA Flyby Transfer

(a) Page ROM → external I/O



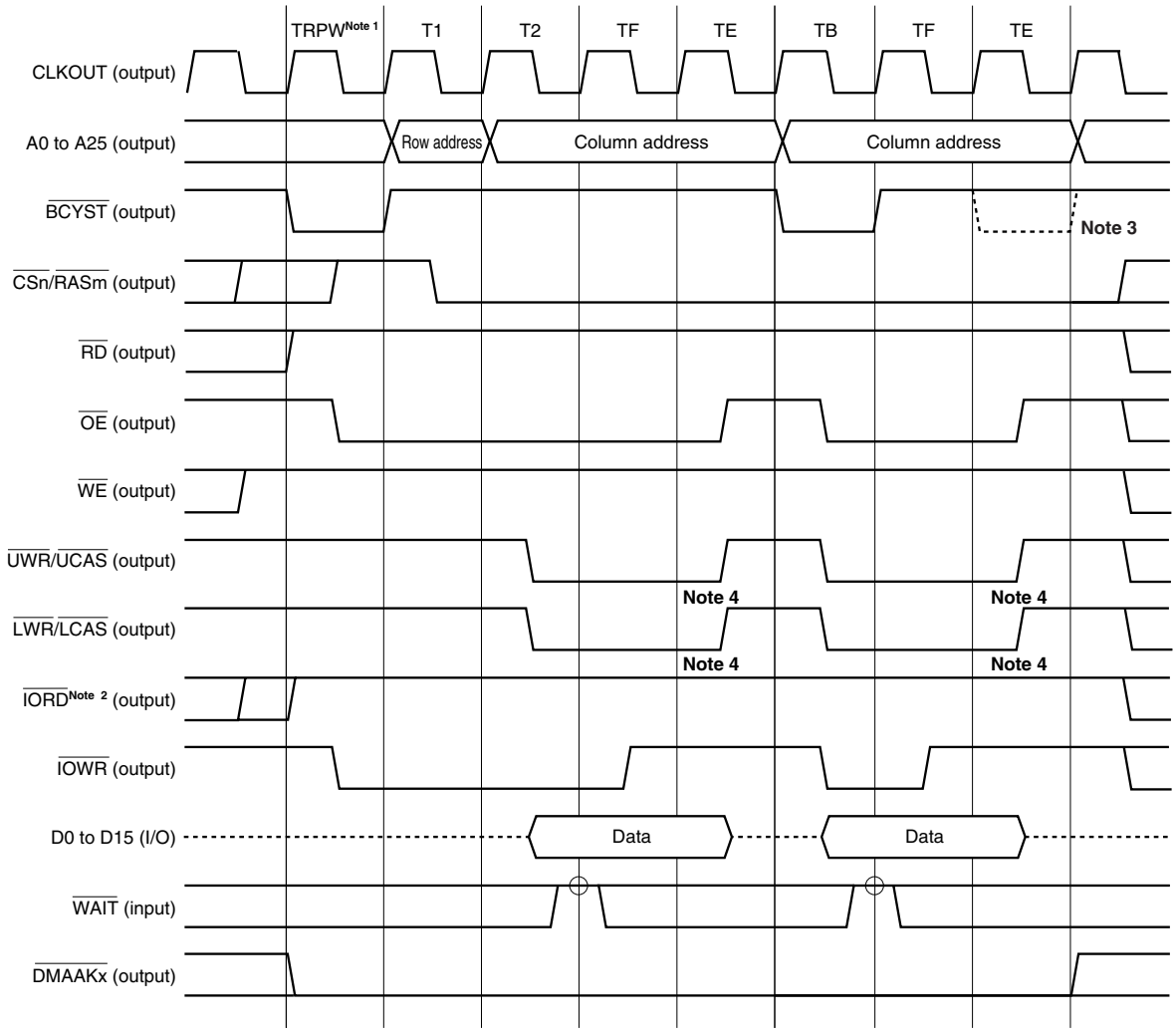
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-17. DRAM Access Timing During DMA Flyby Transfer (1/4)

(a) DRAM → external I/O (when no wait is inserted)



Notes 1. TRPW is always inserted for 1 or more cycles.

2. During DMA flyby transfer, the rise timing of this read cycle is different from that of other transfer operations.

3. When a bus cycle accessing another CS space or a write cycle accessing the same CS space follows this cycle.

4. The rise timing of this write cycle is different from that of a normal EDO DRAM write cycle.

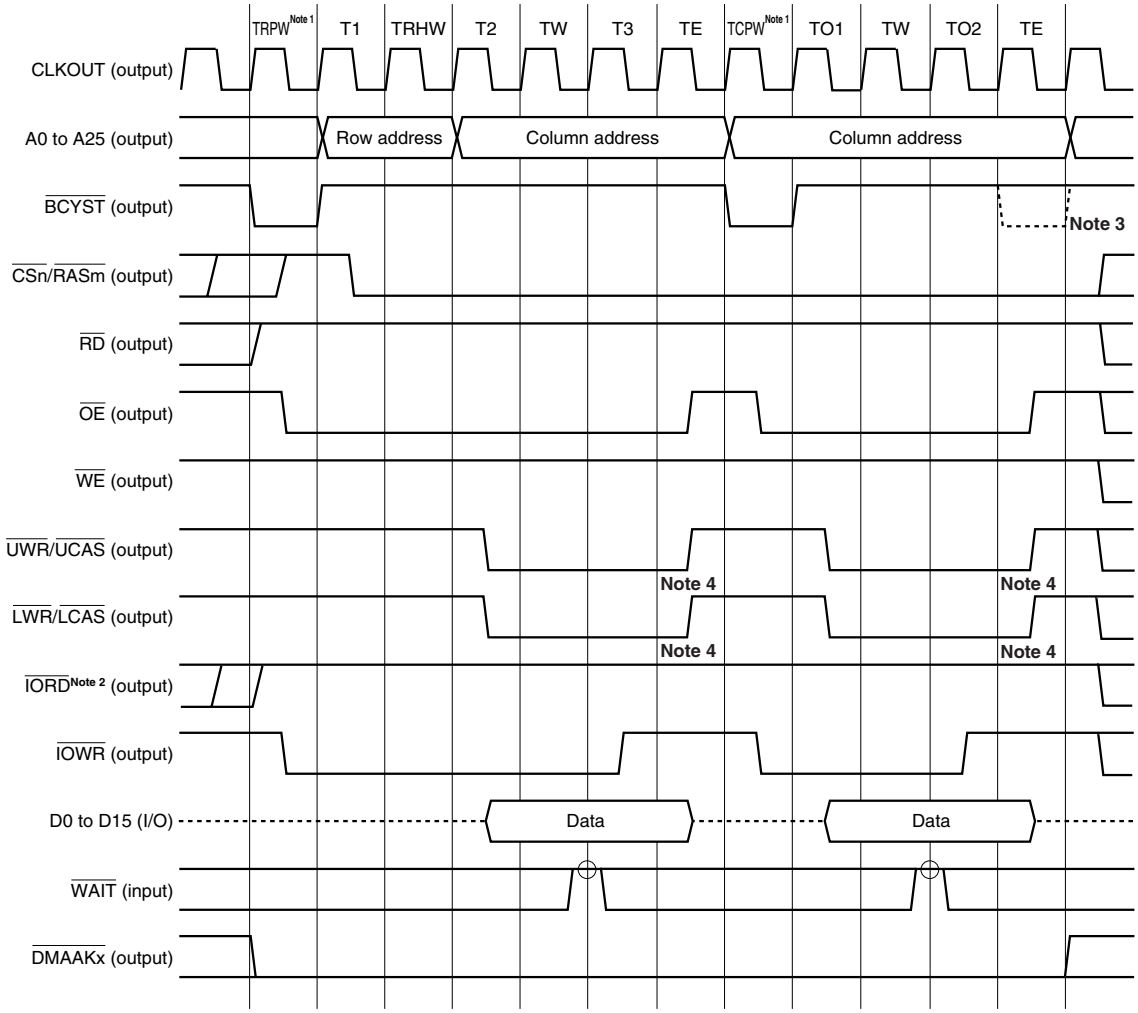
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-17. DRAM Access Timing During DMA Flyby Transfer (2/4)

(b) DRAM → External I/O (when TRHW and TW are inserted)



Notes 1. TRPW and TCPW are always inserted for 1 or more cycles.

2. During DMA flyby transfer, the rise timing of this read cycle is different from that of other transfer operations.

3. When a bus cycle accessing another CS space or a write cycle accessing the same CS space follows this cycle.

4. The rise timing of this write cycle is different from that of a normal EDO DRAM write cycle.

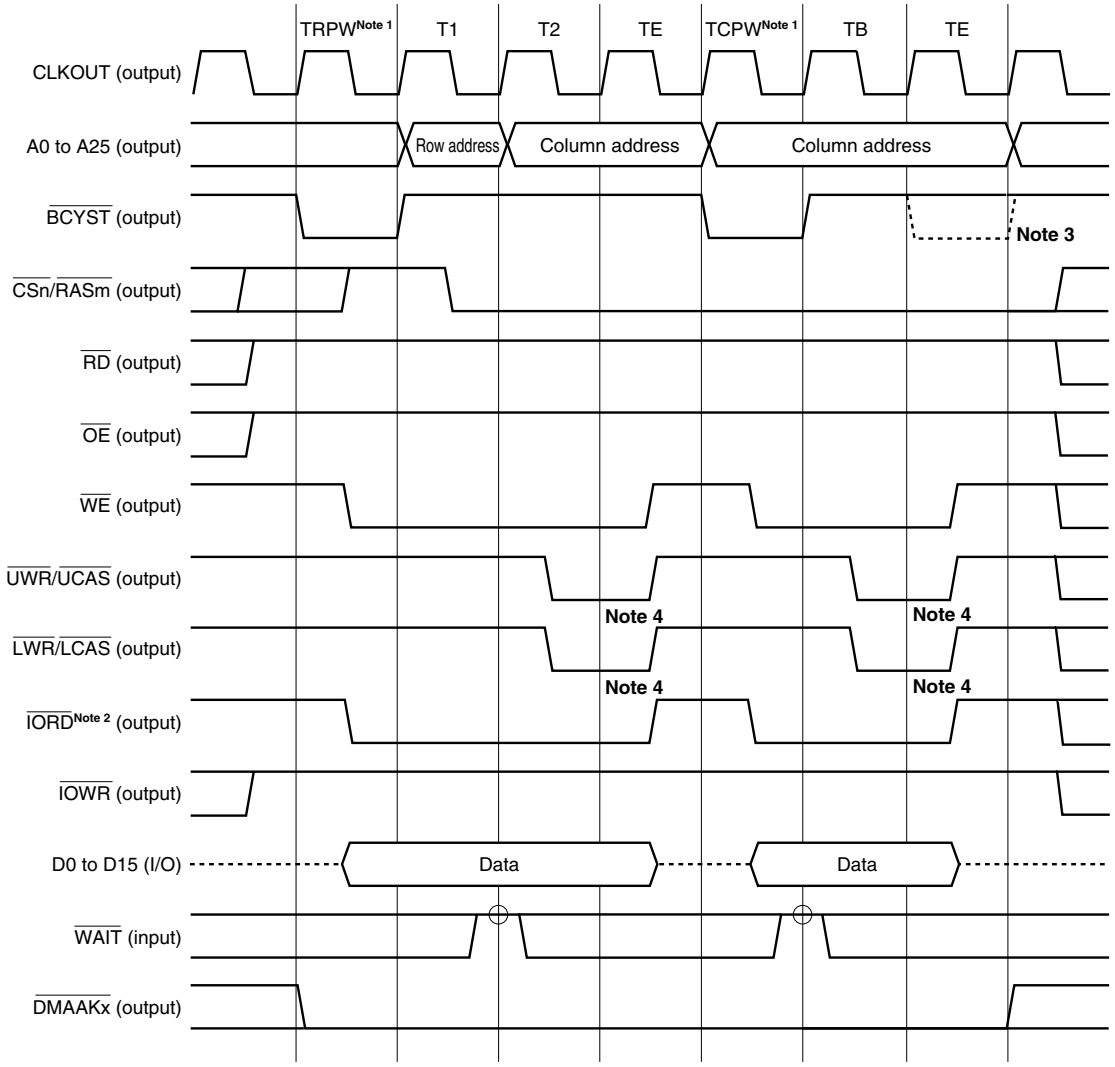
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-17. DRAM Access Timing During DMA Flyby Transfer (3/4)

(c) External I/O → DRAM (when no waits are inserted)



Notes 1. TRPW and TCPW are always inserted for 1 or more cycles.

2. During DMA flyby transfer, the rise timing of this read cycle is different from that of other transfer operations.

3. When a bus cycle accessing another CS space or a write cycle accessing the same CS space follows this cycle.

4. The rise timing of this write cycle is different from that of a normal EDO DRAM write cycle.

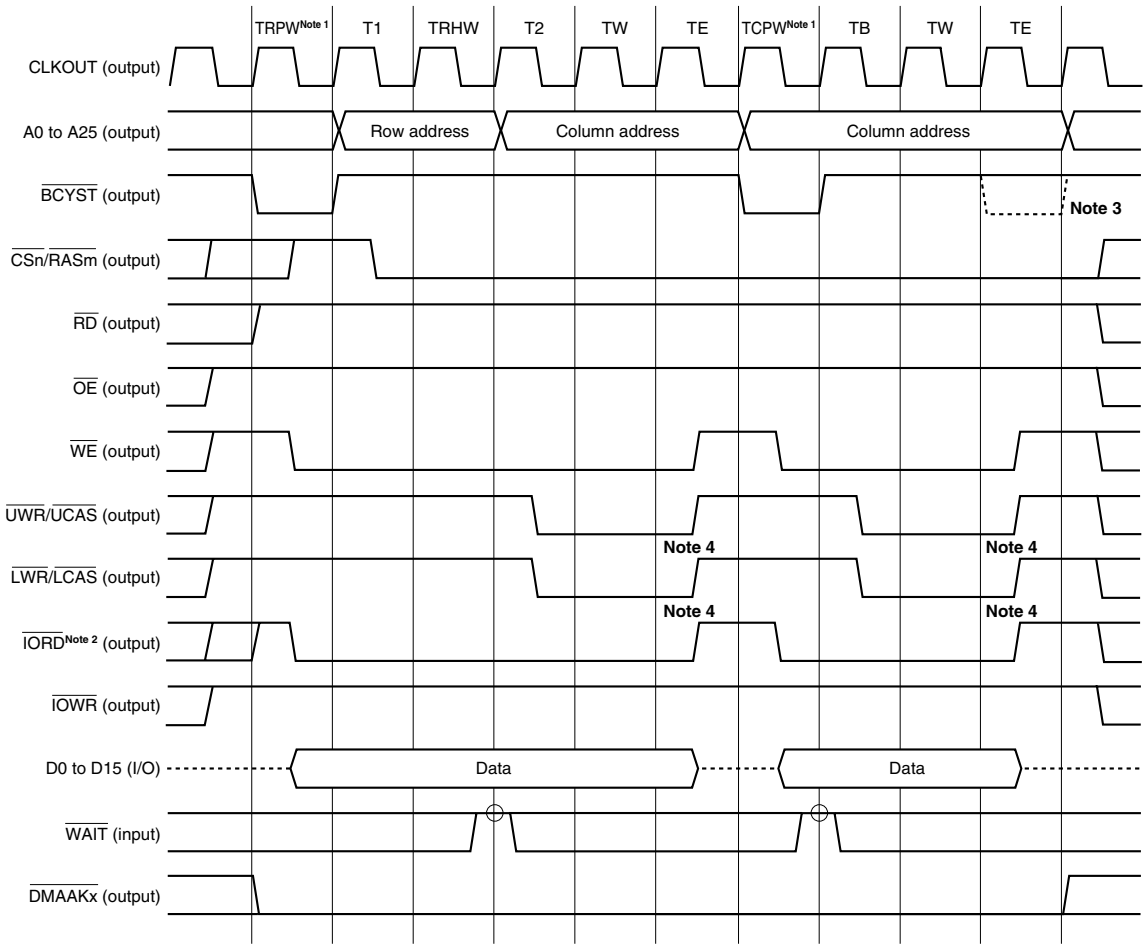
Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

Figure 6-17. DRAM Access Timing During DMA Flyby Transfer (4/4)

(d) External I/O → DRAM (when TRHW and TW are inserted)



Notes 1. TRPW and TCPW are always inserted for 1 or more cycles.

2. During DMA flyby transfer, the rise timing of this read cycle is different from that of other transfer operations.

3. When a bus cycle accessing another CS space or a write cycle accessing the same CS space follows this cycle.

4. The rise timing of this write cycle is different from that of a normal EDO DRAM write cycle.

Remarks 1. The circle ○ indicates the sampling timing.

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

3. n = 0 to 7, m = 1, 3, 4, 6, x = 0 to 3

6.7 Transfer Object

6.7.1 Transfer type and transfer object

Table 6-1 lists the relationships between transfer type and transfer object. The mark "√" means "transfer possible", and the mark "–" means "transfer impossible".

Table 6-1. Relationship Between Transfer Type and Transfer Object

		Destination									
		2-Cycle Transfer					Flyby Transfer				
		Internal ROM	On-chip Peripheral I/O	External I/O	Internal RAM	External Memory	Internal ROM	On-chip Peripheral I/O	External I/O	Internal RAM	External Memory
Source	On-chip peripheral I/O	–	√	√	√	√	–	–	–	–	–
	External I/O	–	√	√	√	√	–	–	–	–	√Note
	Internal RAM	–	√	√	–	√	–	–	–	–	–
	External memory	–	√	√	√	√	–	–	√Note	–	–
	Internal ROM	–	–	–	–	×	–	–	–	–	–

Note In the case of flyby transfer, data cannot be transferred to/from SDRAM.

Cautions 1. The operation is not guaranteed for combinations of transfer destination and source marked with "–" in Table 6-1.

2. In the case of flyby transfer, make the data bus width the same for the source and destination.

3. Addresses between 3FFF000H and 3FFFFFFH cannot be specified for the source and destination address of DMA transfer. Be sure to specify an address between FFFF000H and FFFFFFFH.

Remarks 1. During 2-cycle DMA transfer, if the data bus width of the transfer source and that of the transfer destination are different, the operation becomes as follows.

<16-bit transfer>

- Transfer from a 16-bit bus to an 8-bit bus

★ A read cycle (16 bits) is generated and then a write cycle (8 bits) is generated twice consecutively.

- Transfer from an 8-bit bus to a 16-bit bus

★ A read cycle (8 bits) is generated twice consecutively and then a write cycle (16 bits) is generated.

<8-bit transfer>

- Transfer from 16-bit bus to 8-bit bus

A read cycle (the higher 8 bits go into a high-impedance state) is generated and then a write cycle (8 bits) is generated.

- Transfer from 8-bit bus to 16-bit bus

★ A read cycle (8 bits) is generated and then a write cycle is generated (the higher 8 bits go into a high-impedance state). Data is written in the order from lower bits to higher bits to the transfer destination in the case of little endian and in reverse order in the case of big endian.

2. Transfer between the little endian area and the big endian area is possible.

6.7.2 External bus cycles during DMA transfer

The external bus cycles during DMA transfer are shown below.

Table 6-2. External Bus Cycles During DMA Transfer

Transfer Type	Transfer Object	External Bus Cycle	
2-cycle transfer	On-chip peripheral I/O, internal RAM	None[Note]	–
	External I/O	Yes	SRAM cycle
	External memory	Yes	Memory access cycle set by the BCT register
Flyby transfer	Between external memory and external I/O	Yes	DMA flyby transfer cycle accessing memory that is set as external memory by the BCT register

Note Other external cycles, such as a CPU-based bus cycle can be started.

6.8 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

These priorities are valid in the TI state only. In the block transfer mode, the channel used for transfer is never switched.

In the single-step transfer mode, if a higher priority DMA transfer request is issued while the bus is released (in the TI state), the higher priority DMA transfer request is acknowledged.

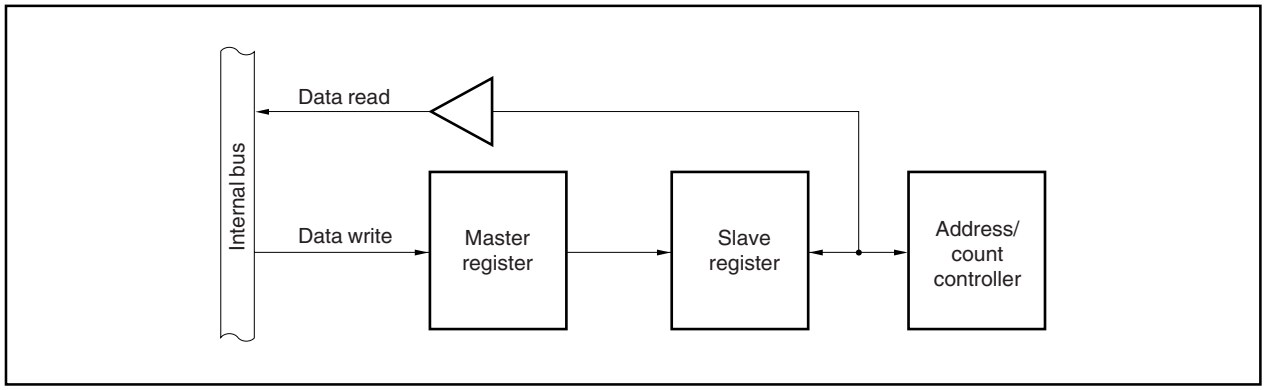
6.9 Next Address Setting Function

The DMA source address registers (DSAnH, DSAnL), DMA destination address registers (DDAnH, DDAnL), and DMA transfer count register (DBCn) are buffer registers with a 2-stage FIFO configuration (n = 0 to 3).
When the terminal count is issued, these registers are automatically rewritten with the value that was set immediately before.

Therefore, during DMA transfer, transfer is automatically started when a new DMA transfer setting is made for these registers and the MLEn bit of the DCHCn register is set to 1 (however, the DMA transfer end interrupt may be issued even if DMA transfer is automatically started).

Figure 6-14 shows the configuration of the buffer register.

Figure 6-18. Buffer Register Configuration



6.10 DMA Transfer Start Factors

There are 3 types of DMA transfer start factors, as shown below.

(1) Request from an external pin ($\overline{\text{DMARQn}}$)

Requests from the $\overline{\text{DMARQn}}$ pin are sampled each time the CLKOUT signal rises (n = 0 to 3).
Hold the request from $\overline{\text{DMARQn}}$ pin until the corresponding $\overline{\text{DMAAKn}}$ signal becomes active.
If a state whereby the Enn bit of the DCHCn register = 1 and the TCn bit = 0 is set, the $\overline{\text{DMARQn}}$ signal in the TI state becomes valid. If the $\overline{\text{DMARQn}}$ signal becomes active in the TI state, it changes to the T0 state and DMA transfer is started.

(2) Request from software

If the STGn, Enn, and TCn bits of the DCHCn register are set as follows, DMA transfer starts (n = 0 to 3).

- STGn bit = 1
- Enn bit = 1
- TCn bit = 0

(3) Request from on-chip peripheral I/O

If, when the Enn and TCn bits of the DCHCn register are set as shown below, an interrupt request is issued from the on-chip peripheral I/O that is set in the DTFRn register, DMA transfer starts (n = 0 to 3).

- Enn bit = 1
- TCn bit = 0

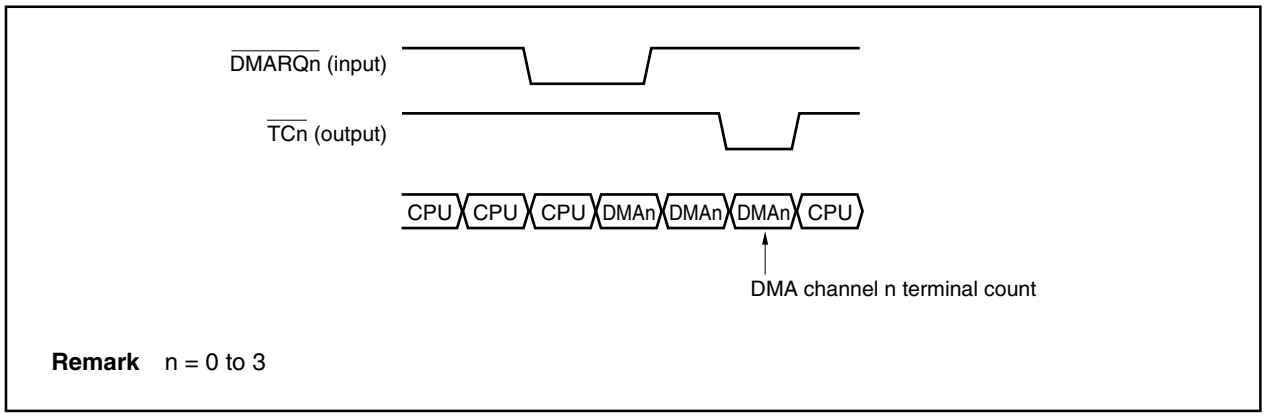
Remark Since the $\overline{\text{DMARQn}}$ signal is level-sampled and not edge-detected, to enable edge detection of a DMA request, set an external interrupt request for the DMA start trigger instead of using the $\overline{\text{DMARQn}}$ signal (n = 0 to 3).

6.11 Terminal Count Output upon DMA Transfer End

The terminal count signal ($\overline{TCn}$) becomes active for one clock during the last DMA transfer cycle (n = 3 to 0).

The $\overline{TCn}$ signal becomes active in the clock following the clock in which the $\overline{BCYST}$ signal becomes active during the last DMA transfer cycle. In 2-cycle transfer, the $\overline{TCn}$ signal becomes active in the write cycle of the last DMA transfer.

Figure 6-19. Terminal Count Signal ($\overline{TCn}$) Timing Example



Remark n = 0 to 3

6.12 Forcible Interruption

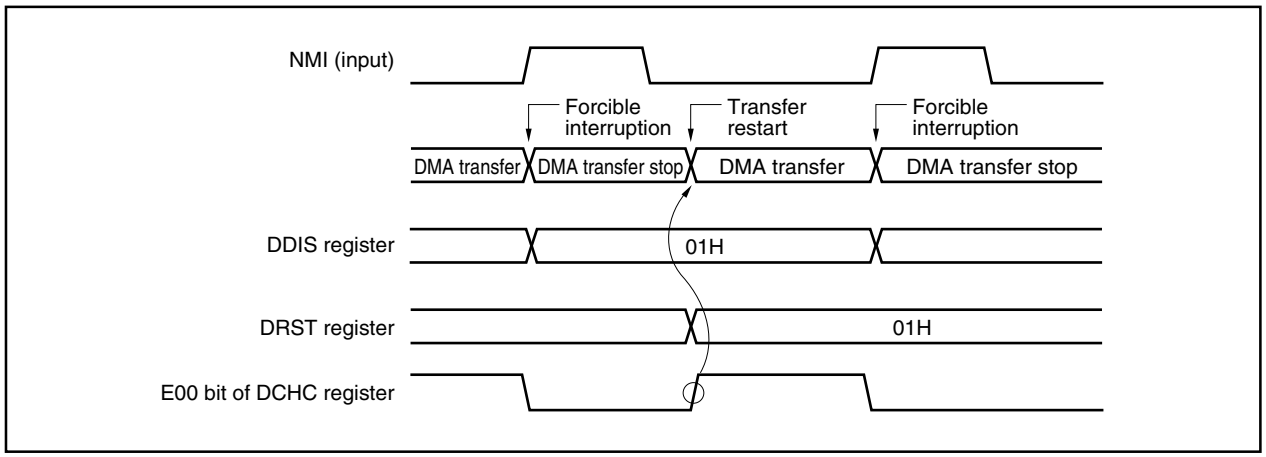
DMA transfer can be forcibly interrupted by NMI input during DMA transfer.

At such a time, the DMAC resets the Enn bit of the DCHCn register of all channels to 0 and the DMA transfer disabled state is entered. An NMI request can then be acknowledged after the DMA transfer that was being executed when the NMI was input is complete (n = 0 to 3).

In the single-step transfer mode or block transfer mode, the DMA transfer request is held in the DMAC. If the Enn bit is set to 1, DMA transfer restarts from the point where it was interrupted.

In the single transfer mode, if the Enn bit is set to 1, the next DMA transfer request is received and DMA transfer starts.

Figure 6-20. Example of Forcible Interrupt of DMA Transfer



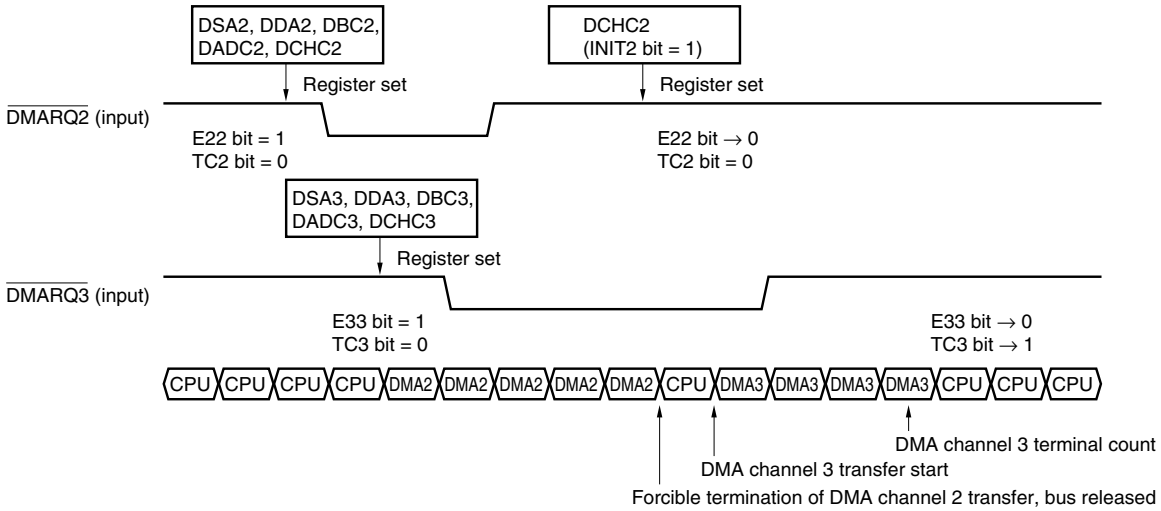
6.13 Forcible Termination

DMA transfer can be forcibly terminated by the INITn bit of the DCHCn register, in addition to the forcible interruption operation by means of NMI input (n = 0 to 3).

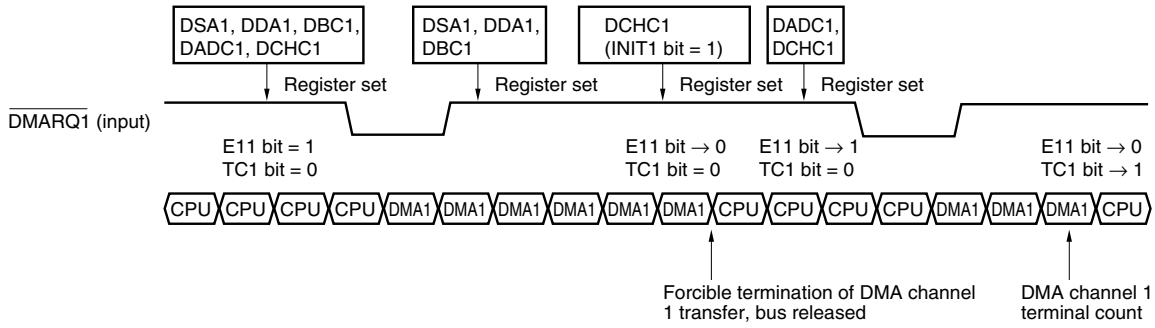
An example of forcible termination by the INITn bit of the DCHCn register is illustrated below (n = 0 to 3).

Figure 6-21. Example of Forcible Termination of DMA Transfer

(a) Block transfer through DMA channel 3 is started during block transfer through DMA channel 2



(b) When transfer is aborted during DMA channel 1 block transfer, and transfer under another condition is executed



Remark During DMA transfer, the next condition can be set because the DSAn, DDAn, and DBCn registers are buffer registers. However, the setting to the DADCn register is invalid (refer to **6.9 Next Address Setting Function** and **6.3.4 DMA addressing control registers 0 to 3 (DADC to DADC3**).

6.14 Times Related to DMA Transfer

The overhead before and after DMA transfer and minimum execution clock for DMA transfer are shown below. In the case of external memory access, the time depends on the type of external memory connected.

Table 6-3. Number of Minimum Execution Clocks in DMA Cycle

DMA Cycle	Number of Minimum Execution Clocks
From $\overline{\text{DMARQn}}$ signal acknowledgement to $\overline{\text{DMAAKn}}$ signal rising	4 clocks
External memory access	Depends on the memory connected.
Internal RAM access	1 clock

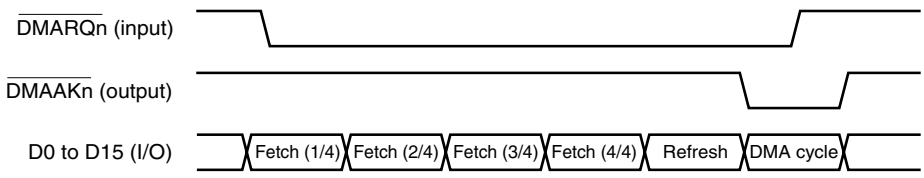
Remark n = 0 to 3

6.15 Maximum Response Time for DMA Transfer Request

The response time for a DMA transfer request becomes the longest under the following conditions (in the DRAM refresh cycle enabled state). However, the case when a higher priority DMA transfer is generated is excluded.

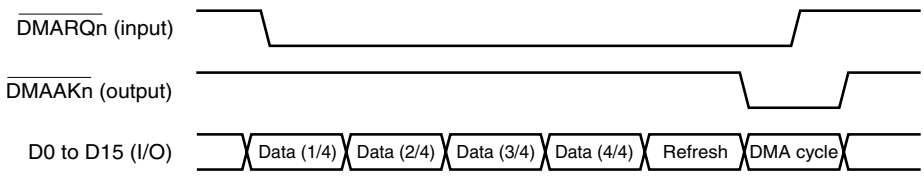
(1) Condition 1

Condition	Instruction fetch from an external memory in 8-bit data bus width
Response time	Tinst × 4 + Tref



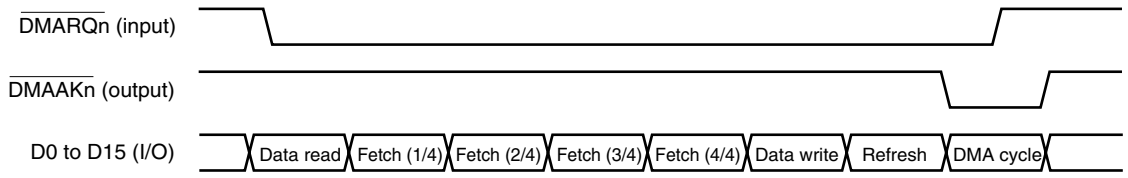
(2) Condition 2

Condition	Word data access with an external memory in 8-bit data bus width
Response time	Tdata × 4 + Tref



(3) Condition 3

Condition	Instruction fetch from an external memory in 8-bit data bus width Execution of a bit manipulation instruction (SET1, CLR1, or NOT1)
Response time	Tinst × 4 + Tdata × 2 + Tref



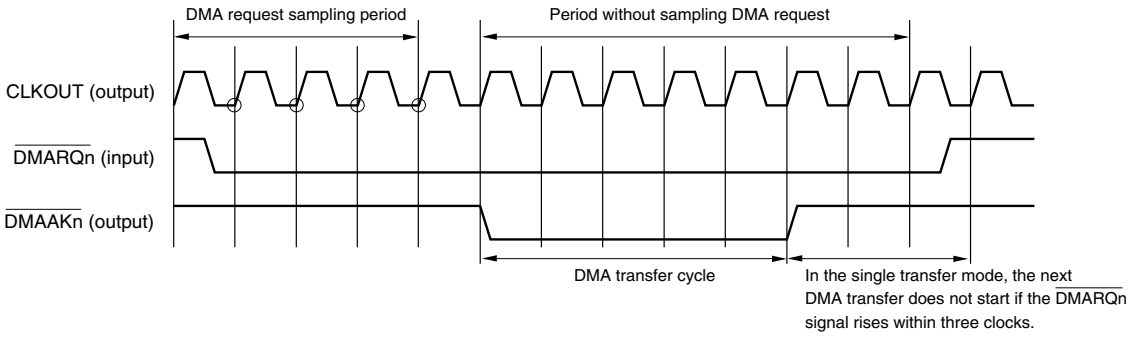
Remarks 1. Tinst: Number of clocks per bus cycle during instruction fetch
Tdata: Number of clocks per bus cycle during data access
Tref: Number of clocks per refresh cycle

2. n = 0 to 3

6.16 One-Time Transfer During Single Transfer via DMARQ0 to DMARQ3 Signals

To perform transfer only one time when single transfer is executed for an external memory via the DMARQn signal, the DMARQn signal must be made inactive within 3 clocks from when the DMAAKn signal becomes inactive (n = 0 to 3).

Figure 6-22. Time to Perform Single Transfer One Time



Remarks 1. The circle ○ indicates the sampling timing.

2. n = 0 to 3

6.17 Cautions

(1) Memory boundary

The transfer operation is not guaranteed if the source or the destination address exceeds the area of DMA objects (external memory, internal RAM, or peripheral I/O) during DMA transfer.

(2) Transfer of misaligned data

DMA transfer of 16-bit bus width misaligned data is not supported. If the source or the destination address is set to an odd address, the LSB of the address is forcibly handled as "0".

(3) Bus arbitration for CPU

When an external device is targeted for DMA transfer, the CPU can access the internal ROM and internal RAM (if they are not subject to DMA transfer).

When DMA transfer is executed between the on-chip peripheral I/O and internal RAM, the CPU can access the internal ROM.

(4) $\overline{\text{DMAAK}}$n signal output

When the transfer object is internal RAM, the $\overline{\text{DMAAK}}$n signal is not output during a DMA cycle for internal RAM (for example, if 2-cycle transfer is performed from internal RAM to an external memory, the $\overline{\text{DMAAK}}$n signal is output only during a DMA write cycle for the external memory).

6.17.1 Interrupt factors

DMA transfer is interrupted if the following factors are issued.

- Bus hold
- Refresh cycle

If the factor that is interrupting DMA transfer disappears, DMA transfer promptly restarts.

6.18 DMA Transfer End

When DMA transfer ends and the TCn bit of the DCHCn register is set to 1, a DMA transfer end interrupt (INTDMAn) is issued to the interrupt controller (INTC) (n = 0 to 3).

CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850E/MA1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 50 interrupt requests.

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

The V850E/MA1 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

7.1 Features

- Interrupts
  - Non-maskable interrupts: 1 source
  - Maskable interrupts: 49 sources
  - 8 levels of programmable priorities (maskable interrupts)
  - Multiple interrupt control according to priority
  - Masks can be specified for each maskable interrupt request.
  - Noise elimination, edge detection, and valid edge specification for external interrupt request signals.
- Exceptions
  - Software exceptions: 32 sources
  - Exception traps: 2 sources (illegal opcode exception and debug trap)

Interrupt/exception sources are listed in Table 7-1.

Table 7-1. Interrupt/Exception Source List (1/2)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	–	Reset input	–	–	0000H	00000000H	Undefined
Non-maskable	Interrupt	NMI0	–	NMI input	–	–	0010H	00000010H	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/ DBG0	–	Illegal opcode/ DBTRAP instruction	–	–	0060H	00000060H	nextPC
Maskable	Interrupt	INTOV00	OVIC00	Timer 00 overflow	RPU	0	0080H	00000080H	nextPC
	Interrupt	INTOV01	OVIC01	Timer 01 overflow	RPU	1	0090H	00000090H	nextPC
	Interrupt	INTOV02	OVIC02	Timer 02 overflow	RPU	2	00A0H	000000A0H	nextPC
	Interrupt	INTOV03	OVIC03	Timer 03 overflow	RPU	3	00B0H	000000B0H	nextPC
	Interrupt	INTP000/ INTM000	P00IC0	Match of INTP000 pin/CCC00	Pin/RPU	4	00C0H	000000C0H	nextPC
	Interrupt	INTP001/ INTM001	P00IC1	Match of INTP001 pin/CCC01	Pin/RPU	5	00D0H	000000D0H	nextPC
	Interrupt	INTP010/ INTM010	P01IC0	Match of INTP010 pin/CCC10	Pin/RPU	6	00E0H	000000E0H	nextPC
	Interrupt	INTP011/ INTM011	P01IC1	Match of INTP011 pin/CCC11	Pin/RPU	7	00F0H	000000F0H	nextPC
	Interrupt	INTP020/ INTM020	P02IC0	Match of INTP020 pin/CCC20	Pin/RPU	8	0100H	00000100H	nextPC
	Interrupt	INTP021/ INTM021	P02IC1	Match of INTP021 pin/CCC21	Pin/RPU	9	0110H	00000110H	nextPC
	Interrupt	INTP030/ INTM030	P03IC0	Match of INTP030 pin/CCC30	Pin/RPU	10	0120H	00000120H	nextPC
	Interrupt	INTP031/ INTM031	P03IC1	Match of INTP031 pin/CCC31	Pin/RPU	11	0130H	00000130H	nextPC
	Interrupt	INTP100	P10IC0	$\overline{\text{INTP100}}$ pin	Pin	12	0140H	00000140H	nextPC
	Interrupt	INTP101	P10IC1	$\overline{\text{INTP101}}$ pin	Pin	13	0150H	00000150H	nextPC
	Interrupt	INTP102	P10IC2	$\overline{\text{INTP102}}$ pin	Pin	14	0160H	00000160H	nextPC
	Interrupt	INTP103	P10IC3	$\overline{\text{INTP103}}$ pin	Pin	15	0170H	00000170H	nextPC
	Interrupt	INTP110	P11IC0	$\overline{\text{INTP110}}$ pin	Pin	16	0180H	00000180H	nextPC
	Interrupt	INTP111	P11IC1	$\overline{\text{INTP111}}$ pin	Pin	17	0190H	00000190H	nextPC
	Interrupt	INTP112	P11IC2	$\overline{\text{INTP112}}$ pin	Pin	18	01A0H	000001A0H	nextPC
	Interrupt	INTP113	P11IC3	$\overline{\text{INTP113}}$ pin	Pin	19	01B0H	000001B0H	nextPC
	Interrupt	INTP120	P12IC0	$\overline{\text{INTP120}}$ pin	Pin	20	01C0H	000001C0H	nextPC
	Interrupt	INTP121	P12IC1	$\overline{\text{INTP121}}$ pin	Pin	21	01D0H	000001D0H	nextPC
	Interrupt	INTP122	P12IC2	$\overline{\text{INTP122}}$ pin	Pin	22	01E0H	000001E0H	nextPC
	Interrupt	INTP123	P12IC3	$\overline{\text{INTP123}}$ pin	Pin	23	01F0H	000001F0H	nextPC
	Interrupt	INTP130	P13IC0	$\overline{\text{INTP130}}$ pin	Pin	24	0200H	00000200H	nextPC
	Interrupt	INTP131	P13IC1	$\overline{\text{INTP131}}$ pin	Pin	25	0210H	00000210H	nextPC
	Interrupt	INTP132	P13IC2	$\overline{\text{INTP132}}$ pin	Pin	26	0220H	00000220H	nextPC
	Interrupt	INTP133	P13IC3	$\overline{\text{INTP133}}$ pin	Pin	27	0230H	00000230H	nextPC
	Interrupt	INTCMD0	CMICD0	CMD0 match signal	RPU	28	0240H	00000240H	nextPC
	Interrupt	INTCMD1	CMICD1	CMD1 match signal	RPU	29	0250H	00000250H	nextPC

Table 7-1. Interrupt/Exception Source List (2/2)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTCMD2	CMICD2	CMD2 match signal	RPU	30	0260H	00000260H	nextPC
	Interrupt	INTCMD3	CMICD3	CMD3 match signal	RPU	31	0270H	00000270H	nextPC
	Interrupt	INTDMA0	DMAIC0	End of DMA0 transfer	DMA	32	0280H	00000280H	nextPC
	Interrupt	INTDMA1	DMAIC1	End of DMA1 transfer	DMA	33	0290H	00000290H	nextPC
	Interrupt	INTDMA2	DMAIC2	End of DMA2 transfer	DMA	34	02A0H	000002A0H	nextPC
	Interrupt	INTDMA3	DMAIC3	End of DMA3 transfer	DMA	35	02B0H	000002B0H	nextPC
	Interrupt	INTCSI0	CSIIC0	CSI0 transmission/ reception completion	SIO	36	02C0H	000002C0H	nextPC
	Interrupt	INTSER0	SEIC0	UART0 reception error	SIO	37	02D0H	000002D0H	nextPC
	Interrupt	INTSR0	SRIC0	UART0 reception completion	SIO	38	02E0H	000002E0H	nextPC
	Interrupt	INTST0	STIC0	UART0 transmission completion	SIO	39	02F0H	000002F0H	nextPC
	Interrupt	INTCSI1	CSIIC1	CSI1 transmission/ reception completion	SIO	40	0300H	00000300H	nextPC
	Interrupt	INTSER1	SEIC1	UART1 reception error	SIO	41	0310H	00000310H	nextPC
	Interrupt	INTSR1	SRIC1	UART1 reception completion	SIO	42	0320H	00000320H	nextPC
	Interrupt	INTST1	STIC1	UART1 transmission completion	SIO	43	0330H	00000330H	nextPC
	Interrupt	INTCSI2	CSIIC2	CSI2 transmission/ reception completion	SIO	44	0340H	00000340H	nextPC
	Interrupt	INTSER2	SEIC2	UART2 reception error	SIO	45	0350H	00000350H	nextPC
	Interrupt	INTSR2	SRIC2	UART2 reception completion	SIO	46	0360H	00000360H	nextPC
	Interrupt	INTST2	STIC2	UART2 transmission completion	SIO	47	0370H	00000370H	nextPC
	Interrupt	INTAD	ADIC	End of A/D conversion	ADC	48	0380H	00000380H	nextPC

Note n = 0 to FH

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

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

nextPC: The PC value that starts the processing following interrupt/exception processing.

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

7.2 Non-Maskable Interrupts

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

A non-maskable interrupt request is input from the NMI pin. When the valid edge specified by bit 0 (ESN0) of external interrupt mode register 0 (INTM0) is detected at the NMI pin, the interrupt occurs.

While the service program of the non-maskable interrupt is being executed (PSW.NP = 1), the acknowledgement of another non-maskable interrupt request is held pending. The pending NMI is acknowledged after the original service program 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 program for an NMI, the number of NMIs that will be acknowledged after PSW.NP is cleared to 0 is only one.

Remark PSW.NP: The NP bit of the PSW register.

7.2.1 Operation

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

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

The servicing configuration of a non-maskable interrupt is shown in Figure 7-1.

Figure 7-1. Servicing Configuration of Non-Maskable Interrupt

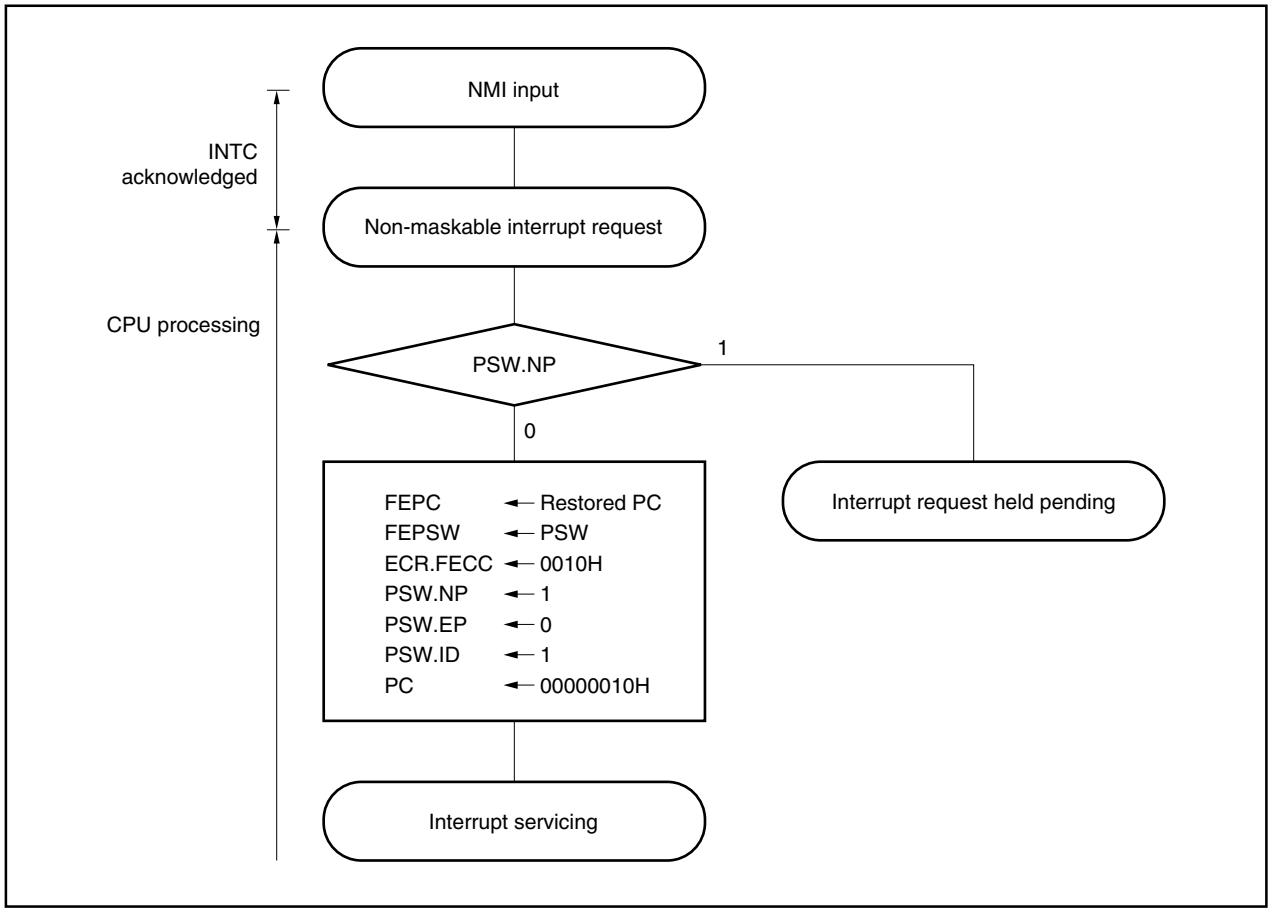
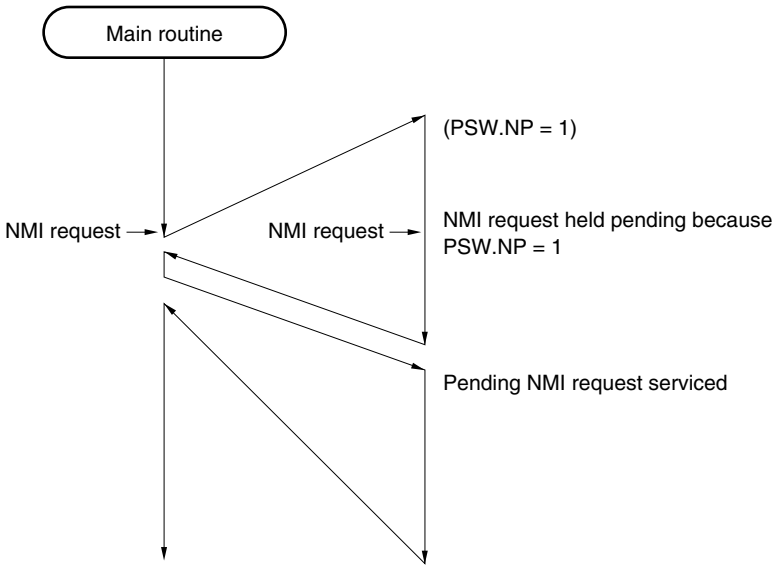
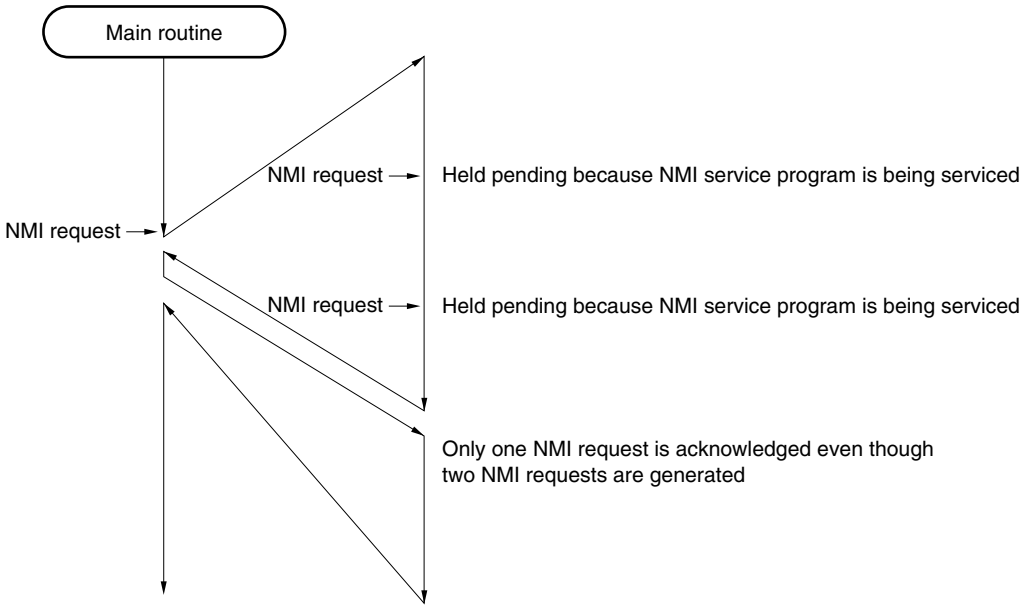


Figure 7-2. Acknowledging Non-Maskable Interrupt Request

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



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



7.2.2 Restore

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

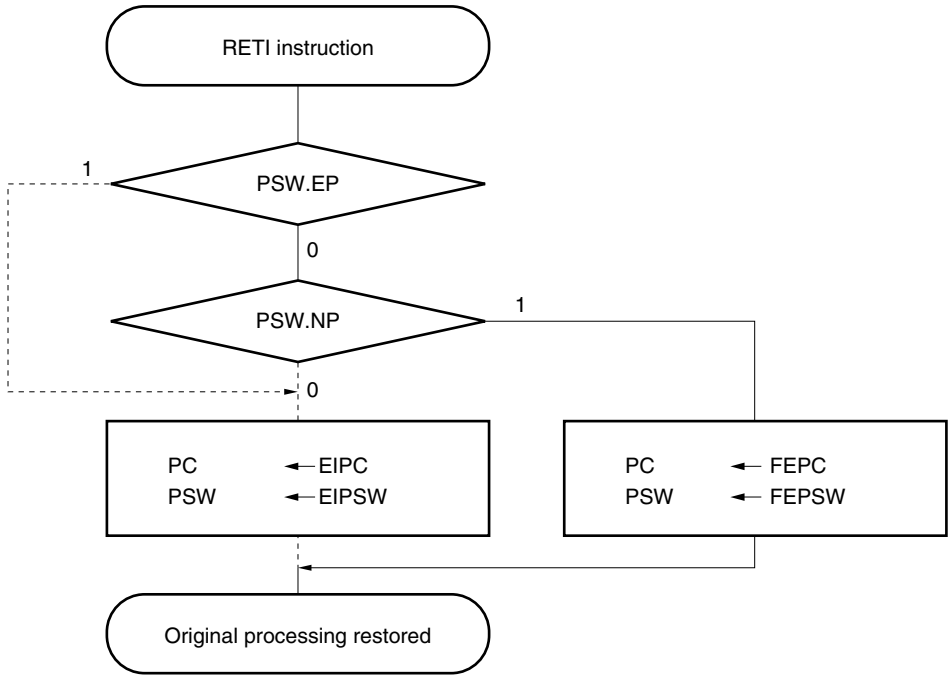
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Restores the values of the PC and the PSW from FEPC and FEPSW, respectively, because the EP bit of the PSW is 0 and the NP bit of the PSW is 1.

<2> Transfers control back to the address of the restored PC and PSW.

Figure 7-3 illustrates how the RETI instruction is processed.

Figure 7-3. RETI Instruction Processing



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

Remark The solid line shows the CPU processing flow.

7.2.3 Non-maskable interrupt status flag (NP)

The NP flag is a status flag that indicates that non-maskable interrupt (NMI) servicing is under execution.

This flag is set when an NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

PSW

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

After reset: 00000020H

Bit position	Bit name	Function
7	NP	NMI Pending Indicates whether NMI interrupt servicing is in progress. 0: No NMI interrupt servicing 1: NMI interrupt currently being serviced

7.2.4 Noise elimination

NMI pin noise is eliminated with analog delay. The delay time is 60 to 300 ns. A signal input that changes within the delay time is not internally acknowledged.

7.2.5 Edge detection function

External interrupt mode register 0 (INTM0) is a register that specifies the valid edge of a non-maskable interrupt (NMI). The NMI valid edge can be specified to be either the rising edge or the falling edge by the ESN0 bit.

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

	7	6	5	4	3	2	1	<0>
INTM0	0	0	0	0	0	0	0	ESN0

Address: FFFFF880H
After reset: 00H

Bit position	Bit name	Function
0	ESN0	Edge Select NMI Specifies the NMI pin's valid edge. 0: Falling edge 1: Rising edge

7.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850E/MA1 has 49 maskable interrupt sources.

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

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

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

However, if multiple interrupts are executed, the following processing is necessary.

<1> Save EIPC and EIPSW in memory or a general-purpose register before executing the EI instruction.
<2> Execute the DI instruction before executing the RETI instruction, then reset EIPC and EIPSW with the values saved in <1>.

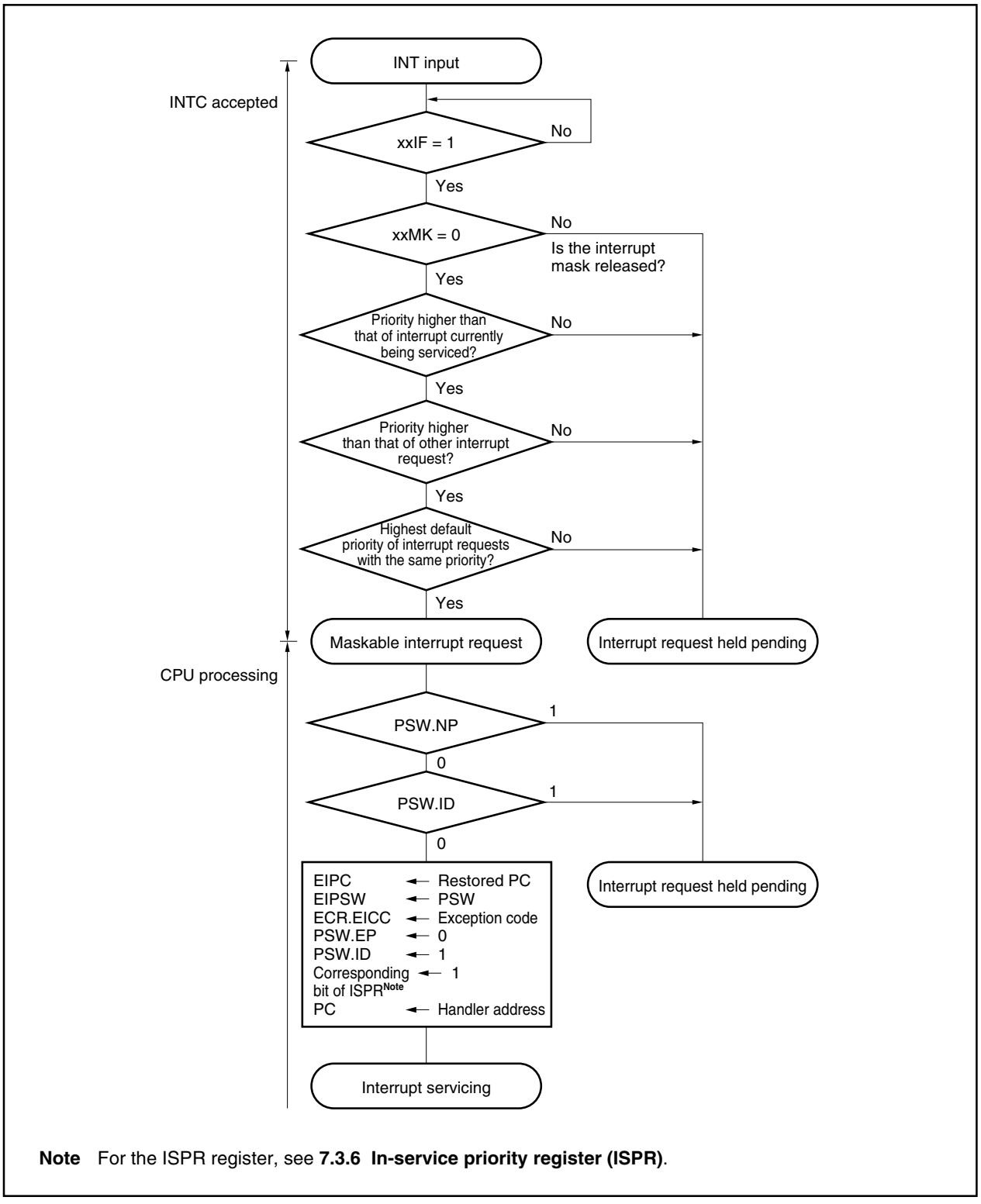
7.3.1 Operation

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

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

The servicing configuration of a maskable interrupt is shown in Figure 7-4.

Figure 7-4. Maskable Interrupt Servicing



Note For the ISPR register, see **7.3.6 In-service priority register (ISPR)**.

The INT input masked by the interrupt controllers and the INT input that occurs while another interrupt is being serviced (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the interrupt controller. In such case, if the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 as set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt servicing.

7.3.2 Restore

Recovery from maskable interrupt servicing is carried out by the RETI instruction.

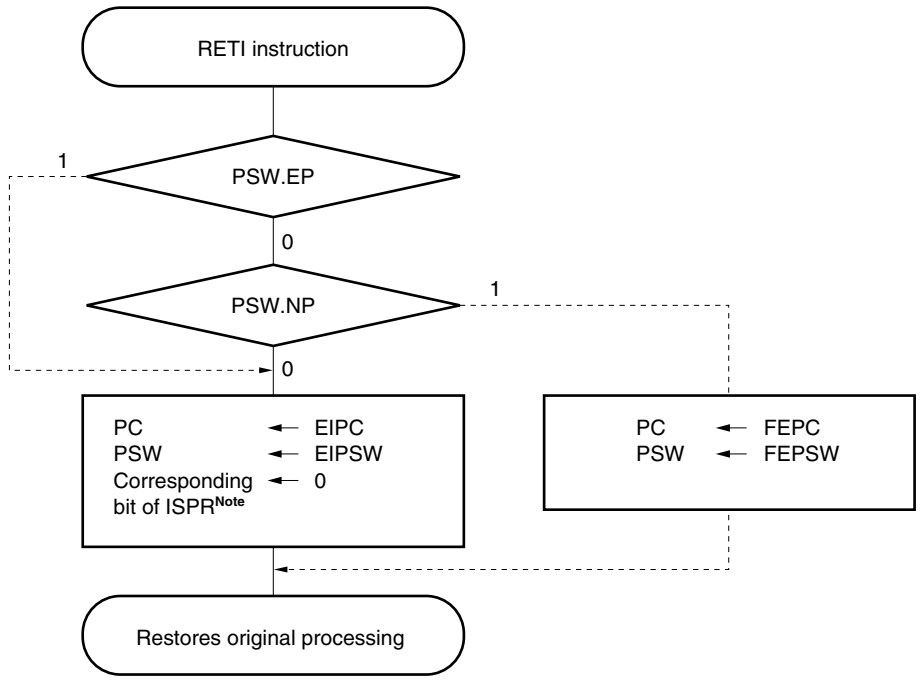
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Restores the values of the PC and the PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.

<2> Transfers control to the address of the restored PC and PSW.

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

Figure 7-5. RETI Instruction Processing



Note For the ISPR register, see **7.3.6 In-service priority register (ISPR)**.

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

Remark The solid line shows the CPU processing flow.

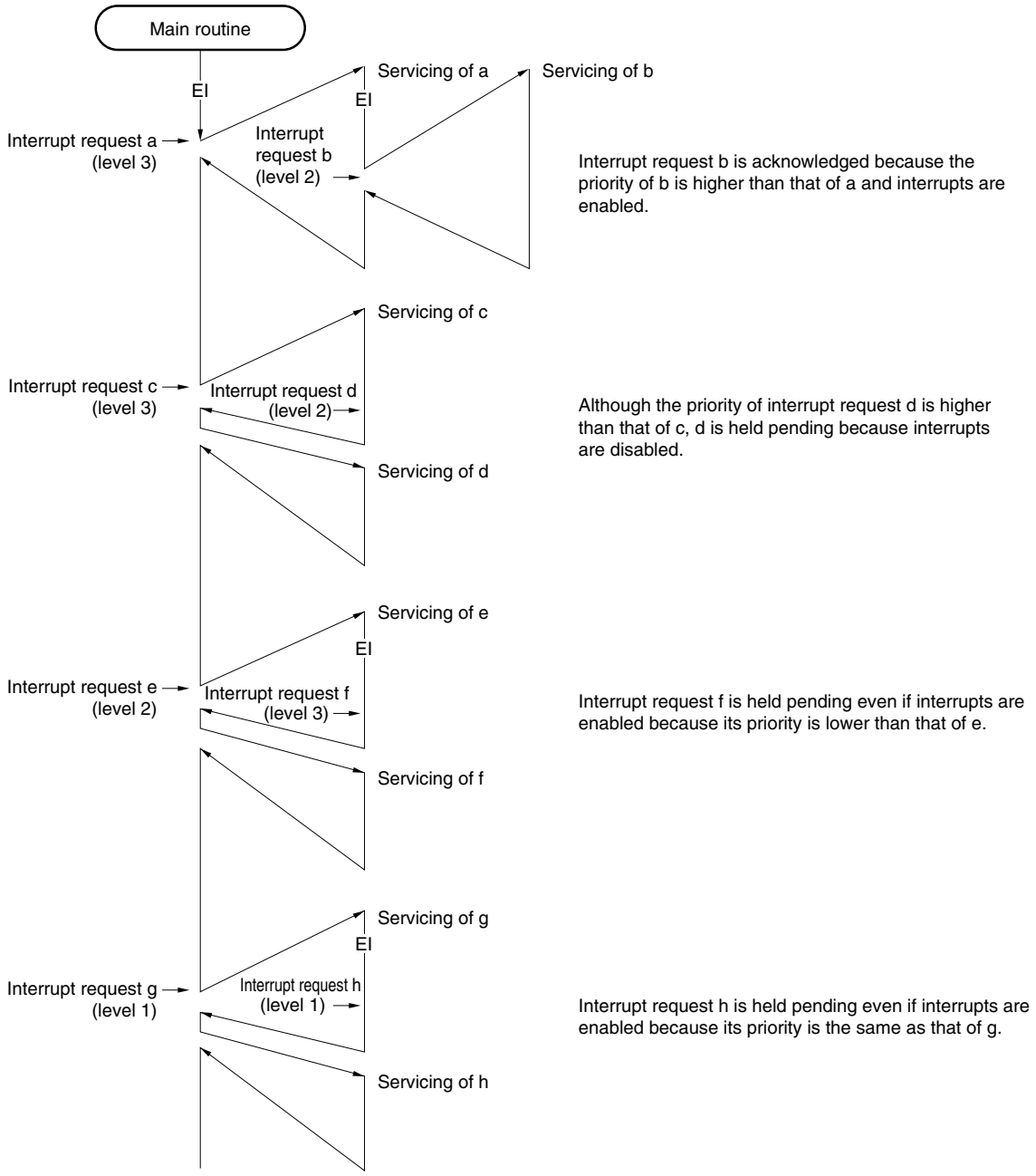
7.3.3 Priorities of maskable interrupts

The V850E/MA1 provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to **Table 7-1 Interrupt/Exception Source List**. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

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

Figure 7-6. Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (1/2)

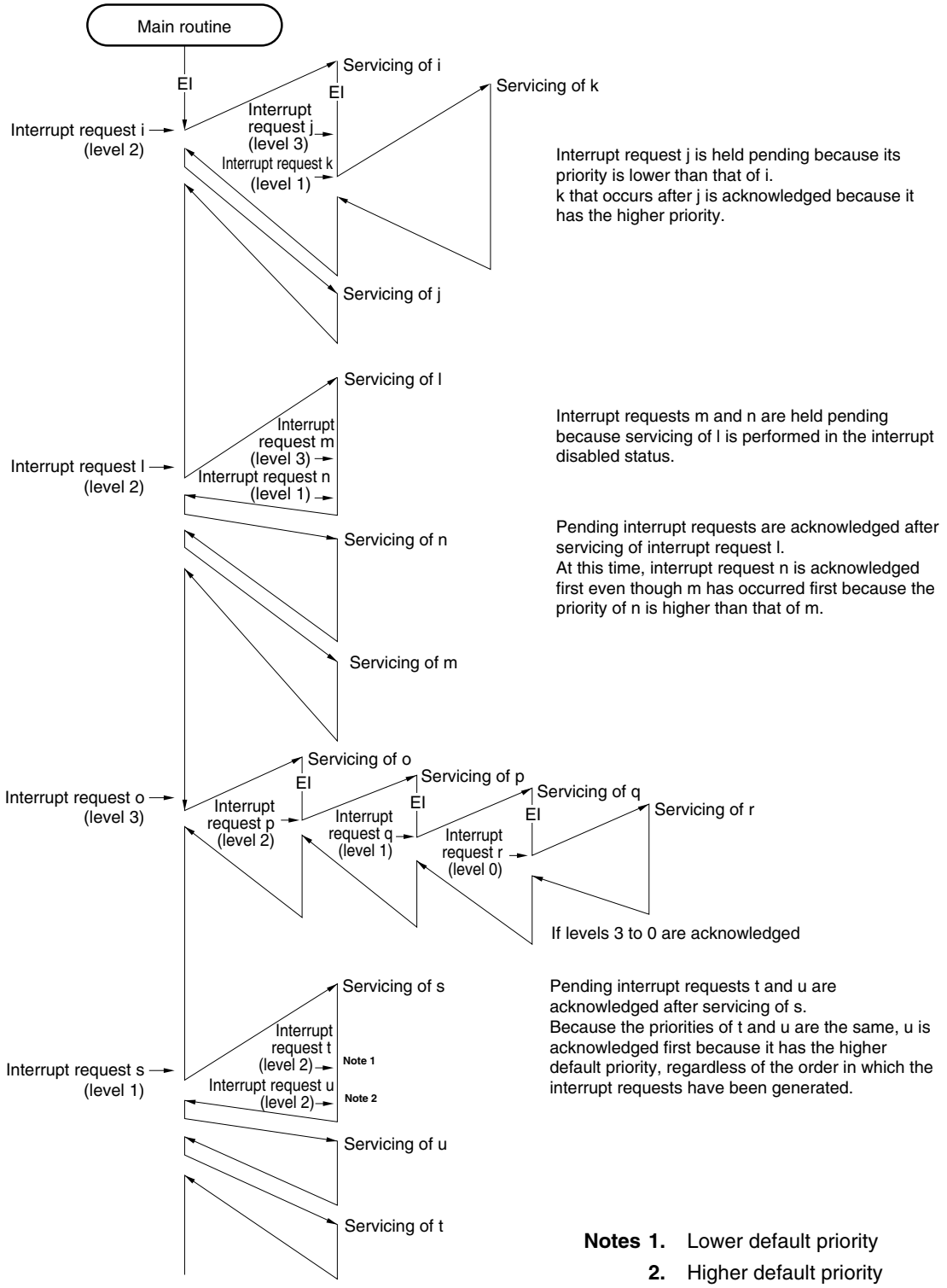


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

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

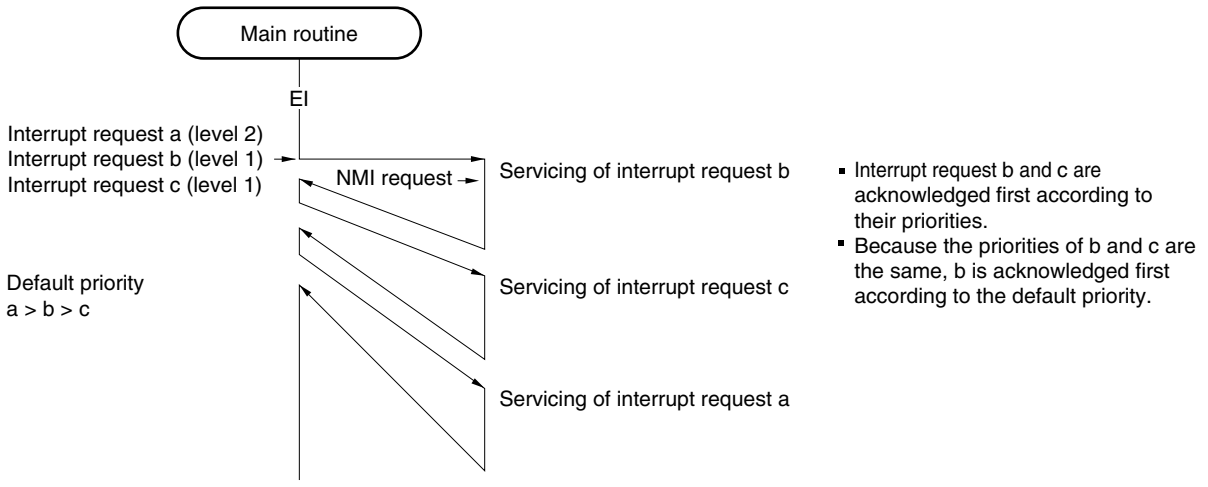
2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 7-6. Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Serviced (2/2)



Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 7-7. Example of Servicing Interrupt Requests Simultaneously Generated



Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

7.3.4 Interrupt control register (xxICn)

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

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

	<7>	<6>	5	4	3	2	1	0	Address	After reset
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0	FFFFF110H to FFFF170H	47H

Bit position	Bit name	Function
7	xxIFn	Interrupt Request Flag This is an interrupt request flag. 0: Interrupt request not issued 1: Interrupt request issued The flag xxIFn is reset automatically by the hardware if an interrupt request is acknowledged.
6	xxMKn	Mask Flag This is an interrupt mask flag. 0: Interrupt servicing enabled 1: Interrupt servicing disabled (pending)
2 to 0	xxPRn2 to xxPRn0	Priority 8 levels of priority order are specified for each interrupt.

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).

Remark xx: Identification name of each peripheral unit (OV, P00 to P03, P10 to P13, CM, DMA, CSI, SE, SR, ST, AD)

n: Peripheral unit number (None or 0 to 3).

The addresses and bits of the interrupt control registers are as follows:

(1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	OVIC00	OVIF0	OVMK0	0	0	0	OVPR02	OVPR01	OVPR00
FFFFF112H	OVIC01	OVIF1	OVMK1	0	0	0	OVPR12	OVPR11	OVPR10
FFFFF114H	OVIC02	OVIF2	OVMK2	0	0	0	OVPR22	OVPR21	OVPR20
FFFFF116H	OVIC03	OVIF3	OVMK3	0	0	0	OVPR32	OVPR31	OVPR30
FFFFF118H	P00IC0	P00IF0	P00MK0	0	0	0	P00PR02	P00PR01	P00PR00
FFFFF11AH	P00IC1	P00IF1	P00MK1	0	0	0	P00PR12	P00PR11	P00PR10
FFFFF11CH	P01IC0	P01IF0	P01MK0	0	0	0	P01PR02	P01PR01	P01PR00
FFFFF11EH	P01IC1	P01IF1	P01MK1	0	0	0	P01PR12	P01PR11	P01PR10
FFFFF120H	P02IC0	P02IF0	P02MK0	0	0	0	P02PR02	P02PR01	P02PR00
FFFFF122H	P02IC1	P02IF1	P02MK1	0	0	0	P02PR12	P02PR11	P02PR10
FFFFF124H	P03IC0	P03IF0	P03MK0	0	0	0	P03PR02	P03PR01	P03PR00
FFFFF126H	P03IC1	P03IF1	P03MK1	0	0	0	P03PR12	P03PR11	P03PR10
FFFFF128H	P10IC0	P10IF0	P10MK0	0	0	0	P10PR02	P10PR01	P10PR00
FFFFF12AH	P10IC1	P10IF1	P10MK1	0	0	0	P10PR12	P10PR11	P10PR10
FFFFF12CH	P10IC2	P10IF2	P10MK2	0	0	0	P10PR22	P10PR21	P10PR20
FFFFF12EH	P10IC3	P10IF3	P10MK3	0	0	0	P10PR32	P10PR31	P10PR30
FFFFF130H	P11IC0	P11IF0	P11MK0	0	0	0	P11PR02	P11PR01	P11PR00
FFFFF132H	P11IC1	P11IF1	P11MK1	0	0	0	P11PR12	P11PR11	P11PR10
FFFFF134H	P11IC2	P11IF2	P11MK2	0	0	0	P11PR22	P11PR21	P11PR20
FFFFF136H	P11IC3	P11IF3	P11MK3	0	0	0	P11PR32	P11PR31	P11PR30
FFFFF138H	P12IC0	P12IF0	P12MK0	0	0	0	P12PR02	P12PR01	P12PR00
FFFFF13AH	P12IC1	P12IF1	P12MK1	0	0	0	P12PR12	P12PR11	P12PR10
FFFFF13CH	P12IC2	P12IF2	P12MK2	0	0	0	P12PR22	P12PR21	P12PR20
FFFFF13EH	P12IC3	P12IF3	P12MK3	0	0	0	P12PR32	P12PR31	P12PR30
FFFFF140H	P13IC0	P13IF0	P13MK0	0	0	0	P13PR02	P13PR01	P13PR00
FFFFF142H	P13IC1	P13IF1	P13MK1	0	0	0	P13PR12	P13PR11	P13PR10
FFFFF144H	P13IC2	P13IF2	P13MK2	0	0	0	P13PR22	P13PR21	P13PR20
FFFFF146H	P13IC3	P13IF3	P13MK3	0	0	0	P13PR32	P13PR31	P13PR30
FFFFF148H	CMICD0	CMIF0	CMMK0	0	0	0	CMPR02	CMPR01	CMPR00
FFFFF14AH	CMICD1	CMIF1	CMMK1	0	0	0	CMPR12	CMPR11	CMPR10
FFFFF14CH	CMICD2	CMIF2	CMMK2	0	0	0	CMPR22	CMPR21	CMPR20
FFFFF14EH	CMICD3	CMIF3	CMMK3	0	0	0	CMPR32	CMPR31	CMPR30
FFFFF150H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF152H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF154H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF156H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF158H	CSIIC0	CSIIF0	CSIMK0	0	0	0	CSIPR02	CSIPR01	CSIPR00

(2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF15AH	SEIC0	SEIF0	SEMK0	0	0	0	SEPR02	SEPR01	SEPR00
FFFFF15CH	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF15EH	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF160H	CSIIC1	CSIIF1	CSIMK1	0	0	0	CSIPR12	CSIPR11	CSIPR10
FFFFF162H	SEIC1	SEIF1	SEMK1	0	0	0	SEPR12	SEPR11	SEPR10
FFFFF164H	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF166H	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF168H	CSIIC2	CSIIF2	CSIMK2	0	0	0	CSIPR22	CSIPR21	CSIPR20
FFFFF16AH	SEIC2	SEIF2	SEMK2	0	0	0	SEPR22	SEPR21	SEPR20
FFFFF16CH	SRIC2	SRIF2	SRMK2	0	0	0	SRPR22	SRPR21	SRPR20
FFFFF16EH	STIC2	STIF2	STMK2	0	0	0	STPR22	STPR21	STPR20
FFFFF170H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0

7.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3)

These registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR3 registers is equivalent to the xxICn register.

The IMRm register (m = 0 to 3) can be read/written in 16-bit units.

If the higher 8 bits of the IMRm register are used as an IMRmH register and the lower 8 bits as an IMRmL register, these registers can be read/written in 8-bit or 1-bit units.

Bits 15 to 1 of the IMR3 register (bits 7 to 0 of the IMR3H register and bits 7 to 1 of the IMR3L register) are fixed to 1. If these bits are not 1, the operation cannot be guaranteed.

	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>		
IMR0	P10MK3	P10MK2	P10MK1	P10MK0	P03MK1	P03MK0	P02MK1	P02MK0	Address FFFFF100H	After reset FFFFH
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>		
	P01MK1	P01MK0	P00MK1	P00MK0	OVMK3	OVMK2	OVMK1	OVMK0		
	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>		
IMR1	CMMK3	CMMK2	CMMK1	CMMK0	P13MK3	P13MK2	P13MK1	P13MK0	Address FFFFF102H	After reset FFFFH
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>		
	P12MK3	P12MK2	P12MK1	P12MK0	P11MK3	P11MK2	P11MK1	P11MK0		
	<15>	<14>	<13>	<12>	<11>	<10>	<9>	<8>		
IMR2	STMK2	SRMK2	SEMK2	CSIMK2	STMK1	SRMK1	SEMK1	CSIMK1	Address FFFFF104H	After reset FFFFH
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>		
	STMK0	SRMK0	SEMK0	CSIMK0	DMAMK3	DMAMK2	DMAMK1	DMAMK0		
	15	14	13	12	11	10	9	8		
IMR3	1	1	1	1	1	1	1	1	Address FFFFF106H	After reset FFFFH
	7	6	5	4	3	2	1	<0>		
	1	1	1	1	1	1	1	ADMK		

Bit position	Bit name	Function
15 to 0 (IMR0 to 2), 0 (IMR3)	xxMKn	Mask Flag Interrupt mask flag 0: Interrupt servicing enabled 1: Interrupt servicing disabled (pending)

Remark xx: Identification name of each peripheral unit (OV, P00 to P03, P10 to P13, CM, DMA, CSI, SE, SR, ST, and AD).

n: Peripheral unit number (None, or 0 to 3)

7.3.6 In-service priority register (ISPR)

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

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

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

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	FFFFF1FAH	00H

Bit position	Bit name	Function
7 to 0	ISPR7 to ISPR0	In-Service Priority Flag 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)

7.3.7 Maskable interrupt status flag (ID)

The ID flag is bit 5 of the PSW and controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests.

	31																				8	7	6	5	4	3	2	1	0	After reset
PSW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z	00000020H

Bit position	Bit name	Function
5	ID	Interrupt Disable Indicates whether maskable interrupt servicing is enabled or disabled. 0: Maskable interrupt request acknowledgement enabled 1: Maskable interrupt request acknowledgement disabled (pending) This bit is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW. Non-maskable interrupt requests and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware. The interrupt request generated during the acknowledgement disabled period (ID = 1) is acknowledged when the xxIFn bit of xxICn is set to 1, and the ID flag is reset to 0.

7.3.8 Noise elimination

The noise of the INTPn, $\overline{\text{INTPm}}$, and TI000 to TI030 pins is eliminated with analog delay (n = 000, 001, 010, 011, 020, 021, 030, 031, m = 103 to 100, 113 to 110, 123 to 120, and 133 to 130). The delay time is about 60 to 220 ns. A signal input that changes within the delay time is not internally acknowledged.

7.3.9 Interrupt trigger mode selection

The valid edge of pins INTP0n0, INTP0n1, $\overline{\text{INTP1nm}}$, ADTRG, and TI0n0 can be selected by program. Moreover, a level trigger can be selected for the $\overline{\text{INTP1nm}}$ pin (n = 0 to 3, m = 0 to 3). The edge that can be selected as the valid edge is one of the following.

- Rising edge
- Falling edge
- Both the rising and falling edges

When the INTP0n0, INTP0n1, INTP1nm, ADTRG, and TI0n0 pins are edge-detected, they become interrupt sources and capture trigger, A/D trigger, and timer external count inputs (n = 0 to 3, m = 0 to 3).

The valid edge is specified by external interrupt mode registers 1 to 4 (INTM1 to INTM4) and valid edge select registers (SESC0 to SESC3). The level trigger is specified by external interrupt mode registers 1 to 4 (INTM1 to INTM4).

(1) External interrupt mode registers 1 to 4 (INTM1 to INTM4)

These registers specify the trigger mode for external interrupt requests (INTP100 to INTP103, INTP110 to INTP113, INTP120 to INTP122, INTP123/ADTRG, INTP130 to INTP133), input via external pins. The correspondence between each register and the external interrupt requests that register controls is shown below.

- INTM1: INTP100 to INTP103
- INTM2: INTP110 to INTP113
- INTM3: INTP120 to INTP122, INTP123/ADTRG
- INTM4: INTP130 to INTP133

INTP123 is the alternate function pin of the A/D converter external trigger input (ADTRG). Therefore, when INTP123/ADTRG is set to the external trigger mode by the TRG0 to TRG2 bits of the A/D converter mode register (ADM), the ES1231 and ES1230 bits specify the valid edge of the external trigger input (ADTRG).

The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read/written in 8-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
INTM1	ES1031	ES1030	ES1021	ES1020	ES1011	ES1010	ES1001	ES1000	FFFFF882H	00H
	INTP103		INTP102		INTP101		INTP100			
INTM2	ES1131	ES1130	ES1121	ES1120	ES1111	ES1110	ES1101	ES1100	FFFFF884H	00H
	INTP113		INTP112		INTP111		INTP110			
INTM3	ES1231	ES1230	ES1221	ES1220	ES1211	ES1210	ES1201	ES1200	FFFFF886H	00H
	INTP123/ADTRG		INTP122		INTP121		INTP120			
INTM4	ES1331	ES1330	ES1321	ES1320	ES1311	ES1310	ES1301	ES1300	FFFFF888H	00H
	INTP133		INTP132		INTP131		INTP130			

Bit position	Bit name	Function
7 to 0	ES1nm1, ES1nm0 (n = 0 to 3, m = 0 to 3)	Edge Select Specifies the valid edge of the $\overline{\text{INTP1nm}}$ and ADTRG pins.

ES1nm1	ES1nm0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	Level detection (low-level detection)[Notes 1, 2, 3]
1	1	Both rising and falling edges

Notes 1. The level of the $\overline{\text{INTP1nm}}$ pin is sampled at the interval of the system clock divided by two, and the P1nIFm bit is latched as an interrupt request when a low level is detected. Therefore, even if the P1nIFm bit of the interrupt control register (P1nICm) is automatically cleared to 0 when the CPU acknowledges an interrupt, the P1nIFm bit is immediately set to 1, and an interrupt is generated continuously. To avoid this, forcibly clear the P1nIFm bit to 0 after making the $\overline{\text{INTP1nm}}$ pin inactive for an external device in the interrupt service routine (n = 0 to 3, m = 0 to 3).

2. When a lower priority level-detection interrupt request (INTP1nm) occurs while another interrupt is being serviced and this newly generated level-detection interrupt request becomes inactive before the current interrupt service is complete, this new interrupt request (INTP1nm) is held pending. To avoid acknowledging this INTP1nm interrupt request, clear the P1nIFm bit of the interrupt control register (n = 0 to 3, m = 0 to 3).

3. When this pin is used as the ADTRG pin, do not select this setting (level detection).

(2) Valid edge select registers C0 to C3 (SESC0 to SESC3)

These registers specify the valid edge for external interrupt requests (INTP000, INTP001, INTP010, INTP011, INTP020, INTP021, INTP030, INTP031, TI000 to TI030), input via external pins. The correspondence between each register and the external interrupt requests that register controls is shown below.

- SESC0: TI000, INTP000, INTP001
- SESC1: TI010, INTP010, INTP011
- SESC2: TI020, INTP020, INTP021
- SESC3: TI030, INTP030, INTP031

The valid edge can be specified independently for each pin (rising edge, falling edge, or both rising and falling edges).

These registers can be read/written in 8-bit units.

Caution When using the INTP0n0/TI0n0 or INTP0n1 pin as INTP0n0, INTP0n1, be sure to preset the TMCCAEn bit of timer mode control register Cn0 (TMCCn0) to 1 (n = 0 to 3).

	7	6	5	4	3	2	1	0	Address	After reset
SESC0	TES01	TES00	0	0	IES0011	IES0010	ES0001	IES0000	FFFFF609H	00H
	TI000				INTP001		INTP000			
SESC1	TES11	TES10	0	0	IES0111	IES0110	IES0101	IES0100	FFFFF619H	00H
	TI010				INTP011		INTP010			
SESC2	TES21	TES20	0	0	IES0211	IES0210	IES0201	IES0200	FFFFF629H	00H
	TI020				INTP021		INTP020			
SESC3	TES31	TES30	0	0	IES0311	IES0310	IES0301	IES0300	FFFFF639H	00H
	TI030				INTP031		INTP030			

Bit position	Bit name	Function
7, 6	TESn1, TESn0 (n = 0 to 3)	Edge Select Specifies the valid edge of the INTPn and TI000 to TI030 pins.
3, 2	IESn1, IESn0 (n = 001, 011, 021, 031)	
1, 0	IESn1, IESn0 (n = 000, 010, 020, 030)	

xESn1	xESn0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both rising and falling edges

7.4 Software Exception

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

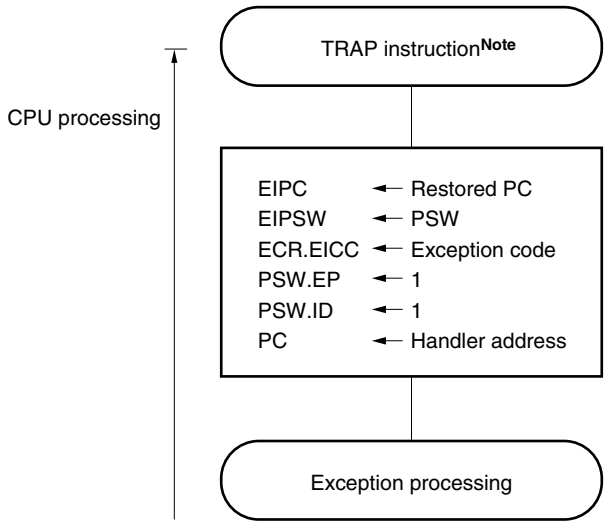
7.4.1 Operation

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

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

Figure 7-8 illustrates the processing of a software exception.

Figure 7-8. Software Exception Processing



Note TRAP instruction format: TRAP vector (the vector is a value from 0 to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 0 to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

7.4.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.

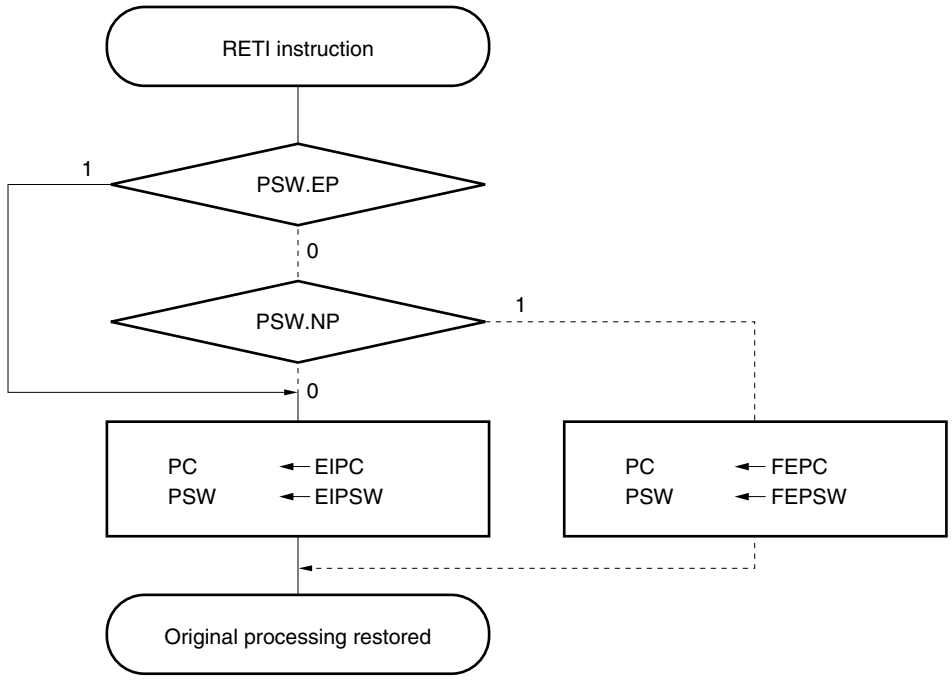
By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 1.

<2> Transfers control to the address of the restored PC and PSW.

Figure 7-9 illustrates the processing of the RETI instruction.

Figure 7-9. RETI Instruction Processing



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

Remark The solid line shows the CPU processing flow.

7.4.3 Exception status flag (EP)

The EP flag is bit 6 of the PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

PSW

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

After reset
00000020H

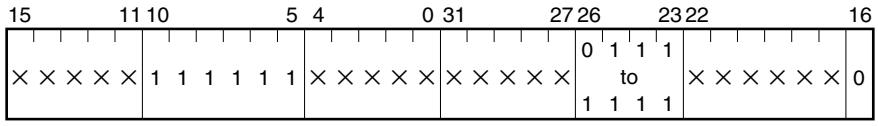
Bit Position	Bit Name	Function
6	EP	Exception Pending Shows that exception processing is in progress. 0: Exception processing not in progress. 1: Exception processing in progress.

7.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/MA1, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

7.5.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.



×: Arbitrary

Caution Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used.

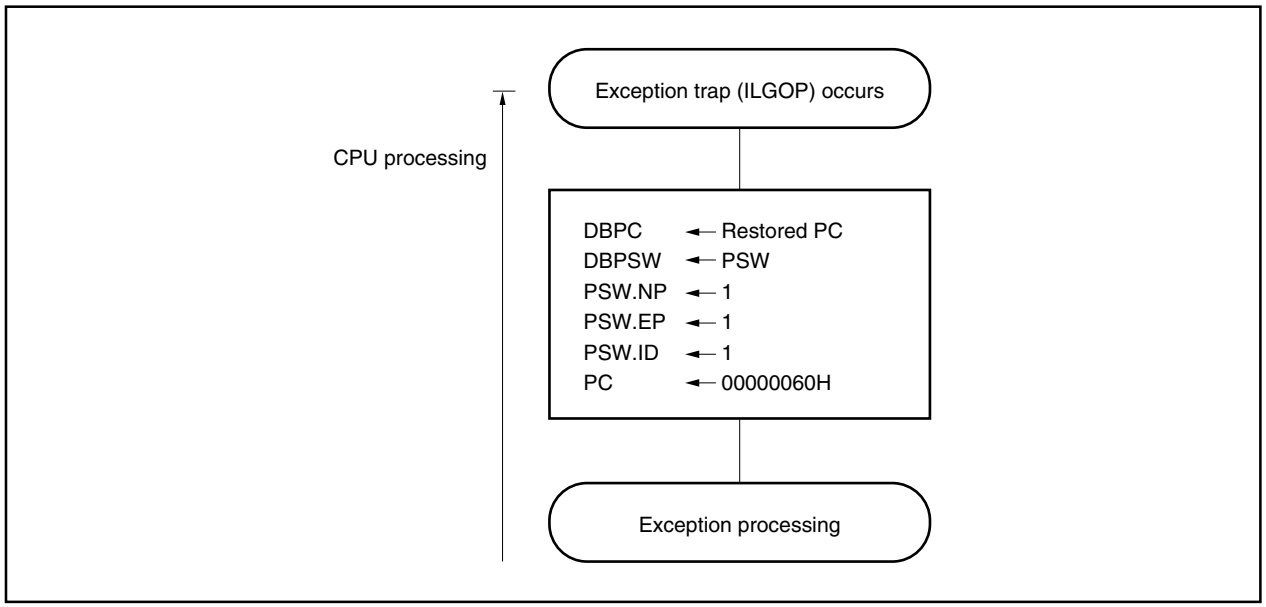
(1) Operation

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

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the NP, EP, and ID bits of the PSW.
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 7-10 illustrates the processing of the exception trap.

Figure 7-10. Exception Trap Processing



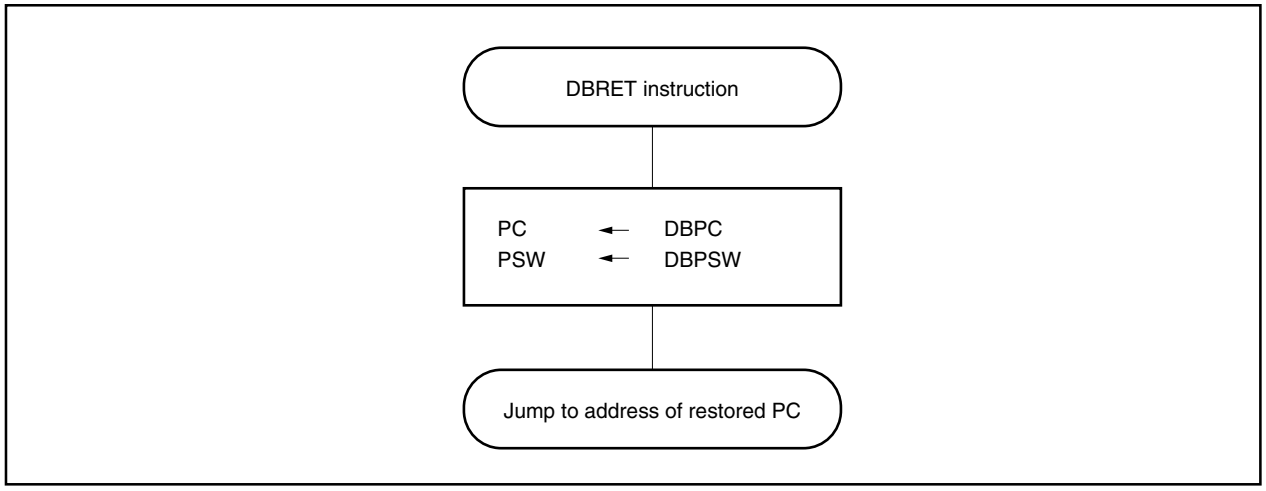
(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 7-11 illustrates the restore processing from an exception trap.

Figure 7-11. Restore Processing from Exception Trap



7.5.2 Debug trap

The debug trap is an exception that can be acknowledged every time and is generated by execution of the DBTRAP instruction.

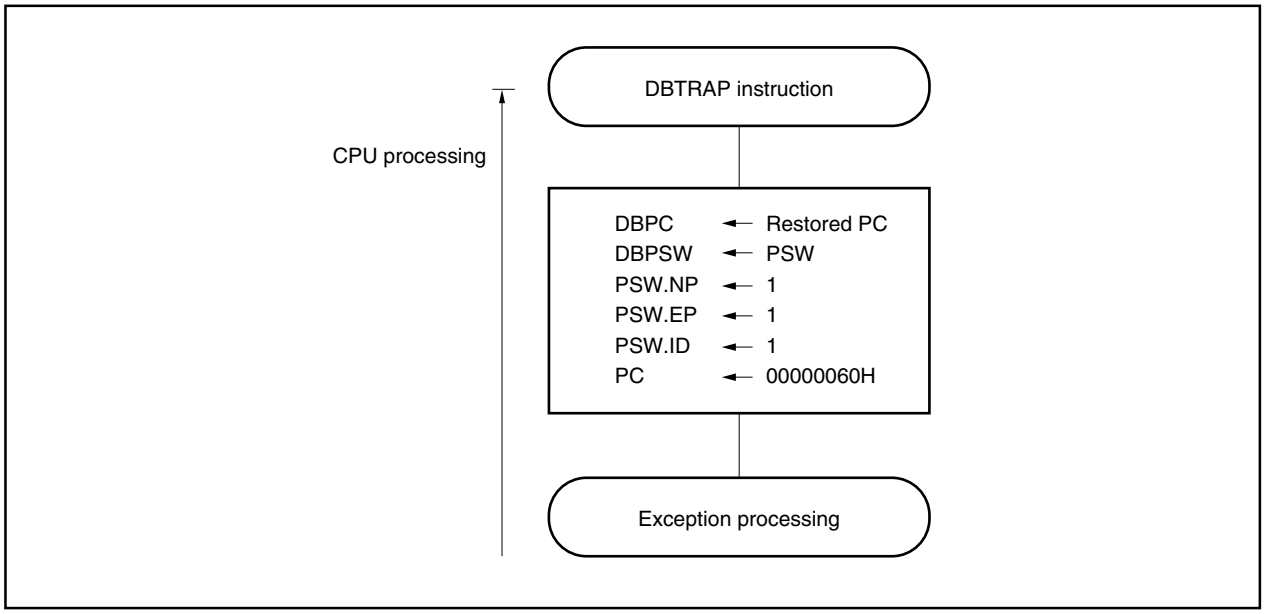
When the debug trap is generated, the CPU performs the following processing.

(1) Operation

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the NP, EP and ID bits of the PSW.
<4> Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

Figure 7-12 illustrates the processing of the debug trap.

Figure 7-12. Debug Trap Processing



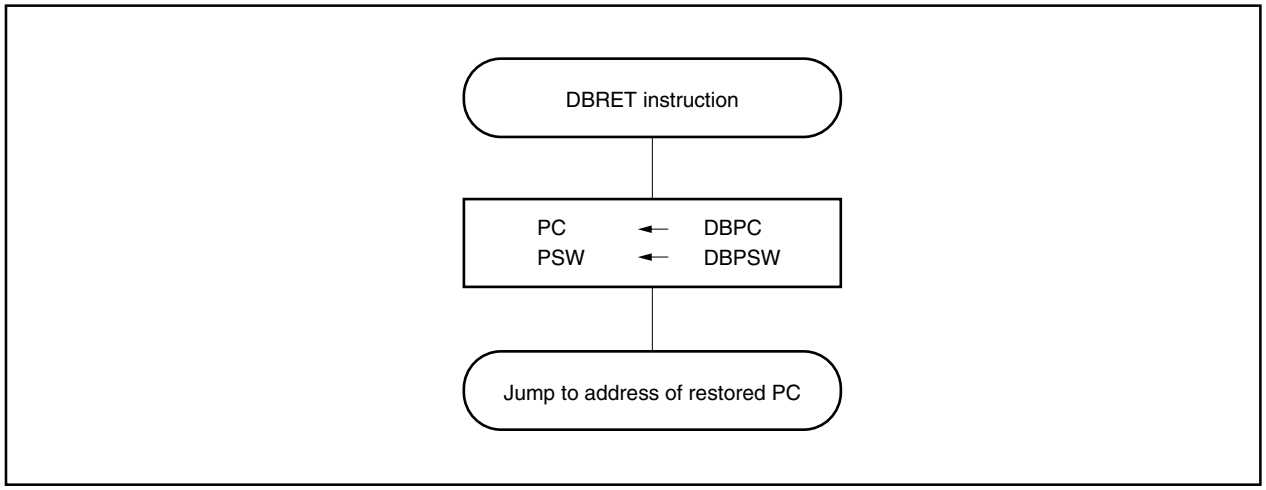
(2) Restore

Recovery from a debug trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 7-13 illustrates the restore processing from a debug trap.

Figure 7-13. Restore Processing from Debug Trap



7.6 Multiple Interrupt Servicing Control

Multiple interrupt servicing control is a process by which an interrupt request that is currently being serviced can be interrupted during servicing if there is an interrupt request with a higher priority level, and the higher priority interrupt request is acknowledged and serviced first.

If there is an interrupt request with a lower priority level than the interrupt request currently being serviced, that interrupt request is held pending.

Multiple interrupt servicing control of maskable interrupts is executed when interrupts are enabled (ID = 0). Thus, to execute multiple interrupts, it is necessary to set the interrupt enabled state (ID = 0) even for an interrupt service routine.

If maskable interrupts are enabled or a software exception is generated in a maskable interrupt or software exception service program, it is necessary to save EIPC and EIPSW.

This is accomplished by the following procedure.

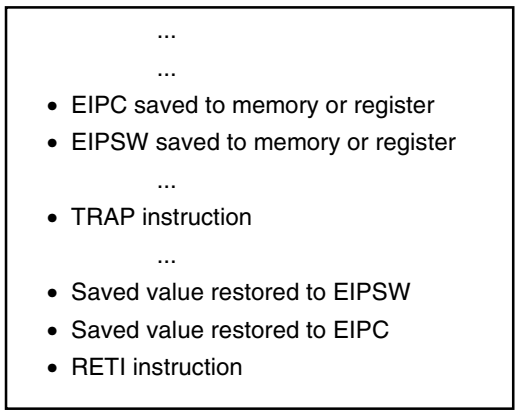
(1) Acknowledgement of maskable interrupts in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• EIPC saved to memory or register
• EIPSW saved to memory or register
• EI instruction (interrupt acknowledgement enabled)
        ...
        ...                                  ← Maskable interrupt acknowledgement
        ...
        ...
• DI instruction (interrupt acknowledgement disabled)
• Saved value restored to EIPSW
• Saved value restored to EIPC
• RETI instruction
```

(2) Generation of exception in service program

Service program of maskable interrupt or exception



The priority order for multiple interrupt servicing control has 8 levels, from 0 to 7 for each maskable interrupt request (0 is the highest priority), but it can be set as desired via software. The priority order is set using the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxICn), provided for each maskable interrupt request. After system reset, an interrupt request is masked by the xxMKn bit and the priority order is set to level 7 by the xxPRn0 to xxPRn2 bits.

The priority order of maskable interrupts is as follows.

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

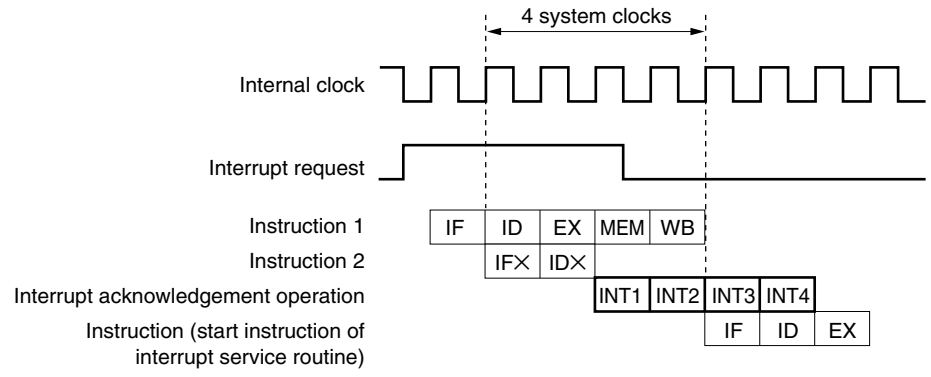
Interrupt servicing that has been suspended as a result of multiple servicing control is resumed after the servicing of the higher priority interrupt has been completed and the RETI instruction has been executed.
A pending interrupt request is acknowledged after the current interrupt servicing has been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

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

7.7 Interrupt Latency Time

The V850E/MA1 interrupt latency time (from interrupt generation to start of interrupt servicing) is described below.

★ **Figure 7-14. Pipeline Operation at Interrupt Request Acknowledgement (Outline)**



INT1 to INT4: Interrupt acknowledgement processing
IF×: Invalid instruction fetch
ID×: Invalid instruction decode

	Interrupt latency time (internal system clock)			Condition
	Internal interrupt	External interrupt		
		INTP0nm	INTP1nm	
Minimum	4	7 + Analog delay time	4 + Analog delay time	The following cases are exceptions. • In IDLE/software STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to interrupt control register
Maximum	10	13 + Analog delay time	10 + Analog delay time	

Remark n = 0 to 3, m = 0, 1

7.8 Periods in Which Interrupts Are Not Acknowledged

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

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- ★ The store instruction for the following registers.
  - Command register (PRCMD)
  - Interrupt-related registers:
    Interrupt control register (xxICn), interrupt mask registers 0 to 3 (IMR0 to IMR3), in-service priority register (ISPR)
  - CSI-related registers:
    Clocked serial interface clock selection registers 0 to 2 (CSIC0 to CSIC2),
    clocked serial interface mode registers 0 to 2 (CSIM0 to CSIM2),
    serial I/O shift registers 0 to 2 (SIO0 to SIO2),
    receive-only serial I/O shift registers 0 to 2 (SIOE0 to SIOE2),
    clocked serial interface transmit buffer registers 0 to 2 (SOTB0 to SOTB2)

CHAPTER 8 PRESCALER UNIT (PRS)

The prescaler divides the internal system clock and supplies the divided clock to internal peripheral units. The divided clock differs depending on the unit.

For the timer units and A/D converter, a 2-division clock is input.

For other units, the input clock is selected using that unit's control register.

The CPU operates with the internal system clock.

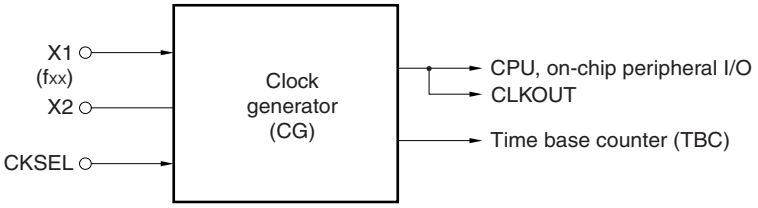
CHAPTER 9 CLOCK GENERATION FUNCTION

The clock generator (CG) generates and controls the internal system clock (ϕ) that is supplied to each internal unit, such as the CPU.

9.1 Features

- Multiplication function using phase locked loop (PLL) synthesizer
- Clock sources
  - Oscillation by connecting a resonator
  - External clock
- Power-save control
  - HALT mode
  - IDLE mode
  - Software STOP mode
- Internal system clock output function

9.2 Configuration



Remark fxx: External resonator or external clock frequency

9.3 Input Clock Selection

The clock generator consists of an oscillator and a PLL synthesizer. For example, connecting a 5.0 MHz crystal resonator or ceramic resonator to pins X1 and X2 enables a 50 MHz internal system clock (ϕ) to be generated when multiplied by 10.

Also, an external clock can be input directly to the oscillator. In this case, the clock signal should be input only to the X1 pin (the X2 pin should be left open).

Two basic operation modes are provided for the clock generator. These are PLL mode and direct mode. The operation mode is selected by the CKSEL pin. The input to this pin is latched on reset.

CKSEL	Operating Mode
0	PLL mode
1	Direct mode

Caution The input level for the CKSEL pin must be fixed. If it is switched during operation, malfunction may occur.

9.3.1 Direct mode

★ In direct mode, an external clock with twice the frequency of the internal system clock is input. The maximum frequency that can be input in direct mode is 50 MHz. The V850E/MA1 is mainly used in application systems in which it is operated at relatively low frequencies. Taking EMI prevention into consideration, if the external clock frequency (f_{XX}) is 32 MHz (internal system clock (ϕ) = 16 MHz) or greater, PLL mode is recommended.

Caution In direct mode, an external clock must be input (an external resonator should not be connected).

9.3.2 PLL mode

In PLL mode, an external resonator is connected or an external clock is input and multiplied by the PLL synthesizer. The multiplied PLL output is divided by the division ratio specified by the clock control register (CKC) to generate a system clock that is 10, 5, 2.5, or 1 times the frequency of the external resonator or external clock (fxx).

After reset, an internal system clock (ϕ) that is the same frequency as the internal clock frequency (fxx) (1 × fxx) is generated.

When a frequency that is 10 times the input clock frequency (fxx) (10 × fxx) is generated, a system with low noise and low power consumption can be realized because a frequency of up to 50 MHz is obtained based on a 5 MHz external resonator or external clock.

In PLL mode, if the clock supply from an external resonator or external clock source stops, operation of the internal system clock (ϕ) based on the free-running frequency of the clock generator's internal voltage controlled oscillator (VCO) continues. However, do not devise an application method expecting to use this free-running frequency.

Example: Clock when PLL mode (ϕ = 10 × fxx) is used

System Clock Frequency (ϕ)	External Resonator or External Clock Frequency (fxx)
50.000 MHz	5.0000 MHz
40.000 MHz	4.0000 MHz

Caution When in PLL mode, only an fxx (4 to 5 MHz) value for which 10 × fxx does not exceed the system clock maximum frequency (50 MHz) can be used for the oscillation frequency or external clock frequency.

★ **However, if any of 5 × fxx, 2.5 × fxx, or 1 × fxx is used, a frequency of 4 to 6.6 MHz can be used.**

Remark If the V850E/MA1 does not need to be operated at high frequency, when PLL mode is selected a power consumption can be reduced by lowering the system clock frequency using software (ϕ = 5 × fxx, ϕ = 2.5 × fxx, or ϕ = 1 × fxx).

9.3.3 Peripheral command register (PHCMD)

This is an 8-bit register that is used to set protection for writing to registers that can significantly affect the system so that the application system is not halted unexpectedly due to an inadvertent program loop. This register is write-only in 8-bit units (when it is read, undefined data is read out).

Writing to the first specific register (CKC or FLPMC register) is only valid after first writing to the PHCMD register. Because of this, the register value can be overwritten only with the specified sequence, preventing an illegal write operation from being performed.

	7	6	5	4	3	2	1	0	Address	After reset
PHCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	FFFFF800H	Undefined

Bit position	Bit name	Function
7 to 0	REG7 to REG0	Registration Code (arbitrary 8-bit data) The specific registers targeted are as follows. • Clock control register (CKC) • Flash programming mode control register (FLPMC)

The generation of an illegal store operation can be checked with the PRERR bit of the peripheral status register (PHS).

9.3.4 Clock control register (CKC)

The clock control register is an 8-bit register that controls the internal system clock (ϕ) in PLL mode. It can be written to only by a specific sequence combination so that it cannot easily be overwritten by mistake due to an inadvertent program loop.

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

Caution Do not change bits CKDIV2 to CKDIV0 in direct mode.

	7	6	5	4	3	2	1	0	Address	After reset
CKC	0	0	TBCS	CESEL	0	CKDIV2	CKDIV1	CKDIV0	FFFFF822H	00H

Bit position	Bit name	Function
5	TBCS	Time Base Count Select Selects the time base counter clock. 0: $f_{XX}/2^8$ 1: $f_{XX}/2^9$ For details, see **9.6.2 Time base counter (TBC)**.
4	CESEL	Crystal/External Select Specifies the functions of the X1 and X2 pins. 0: A resonator is connected to the X1 and X2 pins 1: An external clock is connected to the X1 pin When CESEL = 1, the oscillator feedback loop is disconnected to prevent current leak in software STOP mode.
2 to 0	CKDIV2 to CKDIV0	Clock Divide Sets the internal system clock frequency (ϕ) when PLL mode is used. CKDIV2 / CKDIV1 / CKDIV0 / Internal system clock (ϕ) 0 / 0 / 0 / f_{XX} 0 / 0 / 1 / 2.5 × f_{XX} 0 / 1 / 1 / 5 × f_{XX} 1 / 1 / 1 / 10 × f_{XX} Other than above / Setting prohibited To change the internal system clock frequency in the middle of an operation, be sure to set it to f_{XX} first, and then change the frequency as desired.

Example Clock generator settings

Operation Mode	CKSEL Pin	CKC Register			Input Clock (f_{XX})	Internal System Clock (ϕ)
		CKDIV2	CKDIV0	CKDIV0		
Direct mode	High-level input	0	0	0	16 MHz	8 MHz
PLL mode	Low-level input	0	0	0	5 MHz	5 MHz
		0	0	1	5 MHz	12.5 MHz
		0	1	1	5 MHz	25 MHz
		1	1	1	5 MHz	50 MHz
Other than above					Setting prohibited	Setting prohibited

Set data in the clock control register (CKC) in the following sequence.

<1> Disable interrupts (set the NP bit of PSW to 1)
<2> Prepare data in any one of the general-purpose registers to set in the specific register.
<3> Write data to the peripheral command register (PHCMD)
<4> Set the clock control register (CKC) (with the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<5> Assert the NOP instructions (5 instructions (<5> to <9>))
<10> Release the interrupt disabled state (set the NP bit of PSW to 0).

[Sample coding]

```
<1> LDSR   rX, 5
<2> MOV    0X07, r10
<3> ST.B   r10, PHCMD [r0]
<4> ST.B   r10, CKC [r0]
<5> NOP
<6> NOP
<7> NOP
<8> NOP
<9> NOP
<10> LDSR  rY, 5
```

Remark rX: Value written to PSW
rY: Value returned to PSW

No special sequence is required to read the specific register.

Cautions 1. If an interrupt is acknowledged between the issuance of data to the PHCMD (<3>) and writing to the specific register immediately after (<4>), the write operation to the specific register is not performed and a protection error (the PRERR bit of the PHS register = 1) may occur. Therefore, set the NP bit of the PSW to 1 (<1>) to disable interrupt acknowledgement. Also disable interrupt acknowledgement when selecting a bit manipulation instruction for the specific register setting.

2. Although the data written to the PHCMD register is dummy data, use the same register as the general-purpose register used in specific register setting (<4>) for writing to the PHCMD register (<3>). The same method should be applied when using a general-purpose register for addressing.

★ **3. Be sure to terminate all DMA transfers prior to the execution of the above sequence.**

9.3.5 Peripheral status register (PHS)

If a write operation to the protection-targeted internal registers is not performed in the correct sequence, including access to the command register, writing is not performed and a protection error is generated, setting the status flag (PRERR) to 1. This flag is a cumulative flag. After checking the PRERR flag, it is cleared to 0 by an instruction.

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

	7	6	5	4	3	2	1	<0>	Address	After reset
PHS	0	0	0	0	0	0	0	PRERR	FFFFF802H	00H

Bit position	Bit name	Function
0	PRERR	Protection Error 0: Protection error has not occurred 1: Protection error occurred

The operating conditions of the PRERR flag are as follows.

Set conditions: <1> If the operation of the relevant store instruction for the peripheral I/O is not a write operation for the PHCMD register, but the peripheral specific register is written to.
<2> If the first store instruction operation after the write operation to the PHCMD register is for memory other than the specific registers and peripheral I/O.

Reset conditions: <1> If the PRERR flag of the PHS register is set to 0.
<2> If the system is reset

9.4 PLL Lockup

The lockup time (frequency stabilization time) is the time from when the power is turned on or software STOP mode is released until the phase locks at the prescribed frequency. The state until this stabilization occurs is called the unlocked state, and the stabilized state is called the locked state.

The lock register (LOCKR) has a lock flag that reflects the stabilized state of the PLL frequency.

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

Caution If the phase is locked, the LOCK flag is cleared to 0. If it is unlocked later because of a standby status, the LOCK flag is set to 1. If the phase is unlocked by a cause other than the standby status, however, the LOCK flag is not affected (LOCK = 0).

	7	6	5	4	3	2	1	<0>	Address	After reset
LOCKR	0	0	0	0	0	0	0	LOCK	FFFFF824H	0000000xB

Bit position	Bit name	Function
0	LOCK	Lock Status Flag This is a read-only flag that indicates the PLL lock state. This flag holds the value 0 as long as a lockup state is maintained and is not initialized by a system reset. 0: Indicates that the PLL is locked. 1: Indicates that the PLL is not locked (unlock state).

If the clock stops, the power fails, or some other factor operates to cause an unlock state to occur, for control processing that depends on software execution speed, such as real-time processing, be sure to judge the LOCK flag by software immediately after operation begins so that processing does not begin until after the clock stabilizes.

On the other hand, static processing such as the setting of internal hardware or the initialization of register data or memory data can be executed without waiting for the LOCK flag to be reset.

The relationship between the oscillation stabilization time (the time from when the resonator starts to oscillate until the input waveform stabilizes) when a resonator is used, and the PLL lockup time (the time until frequency stabilizes) is shown below.

Oscillation stabilization time < PLL lockup time.

9.5 Power-Save Control

9.5.1 Overview

The power-save function has the following three modes.

(1) HALT mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the CPU's operation clock stops. Since the supply of clocks to on-chip peripheral functions other than the CPU continues, operation continues. The power consumption of the overall system can be reduced by intermittent operation using a combination of the HALT mode and the normal operation mode.

The system is switched to HALT mode by a specific instruction (the HALT instruction).

(2) IDLE mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the supply of internal system clocks is stopped, which causes the overall system to stop.

When the system is released from IDLE mode, it can be switched to normal operation mode quickly because the oscillator's oscillation stabilization time does not need to be secured.

The system is switched to IDLE mode by a PSMR register setting.

IDLE mode is located midway between software STOP mode and HALT mode in relation to the clock stabilization time and current consumption. It is used for situations in which a low-current-consumption mode is to be used and the clock stabilization time is to be eliminated after the mode is released.

(3) Software STOP mode

In this mode, the overall system is stopped by stopping the clock generator (oscillator and PLL synthesizer). The system enters an ultra-low-power-consumption state in which only leakage current is lost.

The system is switched to software STOP mode by a PSMR register setting.

(a) PLL mode

The system is switched to software STOP mode by setting the register using software. The PLL synthesizer's clock output is stopped at the same time the oscillator is stopped. After software STOP mode is released, the oscillator's oscillation stabilization time must be secured until the system clock stabilizes. Also, PLL lockup time may be required depending on the program. When a resonator or external clock is connected, following the release of the software STOP mode, execution of the program is started after the count time of the time base counter has elapsed.

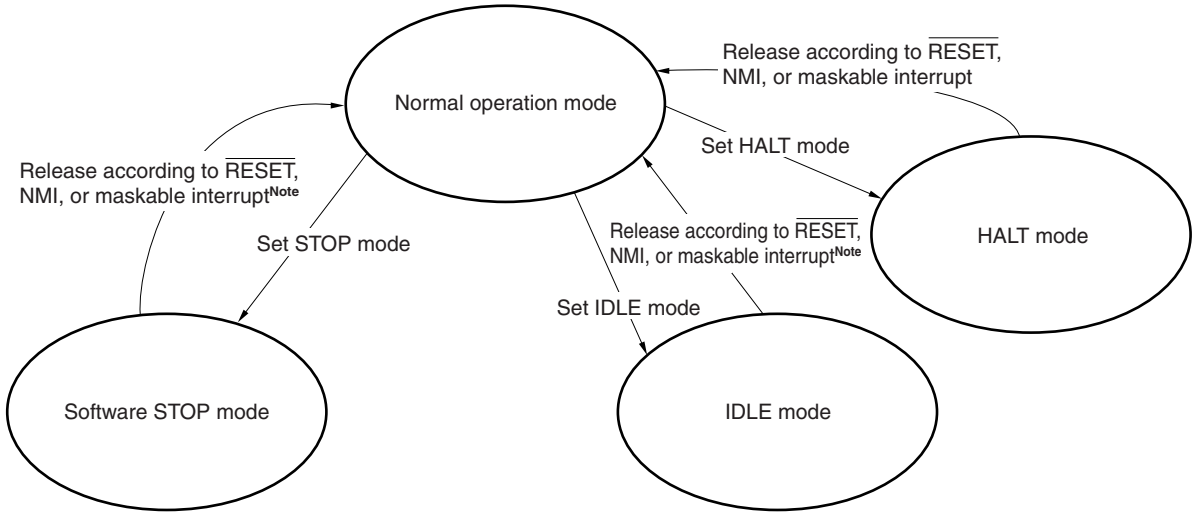
(b) Direct mode

To stop the clock, set the X1 pin to low level. After the release of software STOP mode, execution of the program is started after the count time of the time base counter has elapsed.

Figure 9-1 shows the operation of the clock generator in normal operation mode, HALT mode, IDLE mode, and software STOP mode.

An effective low power consumption system can be realized by combining these modes and switching modes according to the required use.

Figure 9-1. Power-Save Mode State Transition Diagram



Note INTP1nn (n = 0 to 3)
When level detection is specified for the $\overline{\text{INTP1nn}}$ pin, software STOP mode and IDLE mode cannot be released.

Table 9-1. Clock Generator Operation Using Power-Save Control

Clock Source		Power-Save Mode	Oscillator	PLL Synthesizer	Clock Supply to Peripheral I/O	Clock Supply to CPU
PLL mode	Oscillation with resonator	Normal operation	√	√	√	√
		HALT mode	√	√	√	–
		IDLE mode	√	√	–	–
		Software STOP mode	–	–	–	–
	External clock	Normal operation	–	√	√	√
		HALT mode	–	√	√	–
		IDLE mode	–	√	–	–
		Software STOP mode	–	–	–	–
Direct mode	External clock	Normal operation	–	–	√	√
		HALT mode	–	–	√	–
		IDLE mode	–	–	–	–
		Software STOP mode	–	–	–	–

Remark √: Operating
–: Stopped

9.5.2 Control registers

(1) Power-save mode register (PSMR)

This is an 8-bit register that controls power-save mode. It is effective only when the STB bit of the PSC register is set to 1.

Writing to the PSMR register is executed by the store instruction (ST/SST instruction) and a bit manipulation instruction (SET1/CLR1/NOT1 instruction).

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

	7	6	5	4	3	2	1	<0>	Address	After reset
PSMR	0	0	0	0	0	0	0	PSM	FFFFF820H	00H

Bit position	Bit name	Function
0	PSM	Power Save Mode Specifies IDLE mode or software STOP mode. 0: Switches the system to IDLE mode 1: Switches the system to software STOP mode

(2) Command register (PRCMD)

This is an 8-bit register that is used to set protection for write operations to registers that can significantly affect the system so that the application system is not halted unexpectedly due to an inadvertent program loop.

Writing to the first specific register (power-save control register (PSC)) is only valid after first writing to the PRCMD register. Because of this, the register value can be overwritten only by the specified sequence, preventing an illegal write operation from being performed.

This register is write-only in 8-bit units. When it is read, undefined data is read out.

	7	6	5	4	3	2	1	0	Address	After reset
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	FFFFF1FCH	Undefined

Bit position	Bit name	Function
7 to 0	REG7 to REG0	Registration Code (arbitrary 8-bit data) The specific register targeted is the power-save control register (PSC).

(3) Power-save control register (PSC)

This is an 8-bit register that controls the power-save function. This register, which is one of the specific registers, is valid only when accessed in a specific sequence during a write operation.

This register can be read or written in 8-bit or 1-bit units. If bit 7 or 6 is set to 1, operation cannot be guaranteed.

Caution It is impossible to set the STB bit and the NMIM or INTM bit at the same time. Be sure to set the STB bit after setting the NMIM or INTM bit.

	7	6	<5>	<4>	3	2	<1>	0	Address	After reset
PSC	0	0	NMIM	INTM	0	0	STB	0	FFFFF1FEH	00H

Bit position	Bit name	Function
5	NMIM	NMI Mode This is the enable/disable setting bit for standby mode release using the valid edge input of NMI. 0: Release by NMI enabled 1: Release by NMI disabled
4	INTM	INT Mode This is the enable/disable setting for standby mode release using an unmasked maskable interrupt (INTP1nn) (n = 0 to 3). 0: Release by maskable interrupt enabled 1: Release by maskable interrupt disabled
1	STB	Standby Mode Indicates the standby mode status. If 1 is written to this bit, the system enters IDLE or software STOP mode (set by the PSM bit of the PSMR register). When standby mode is released, this bit is automatically reset to 0. 0: Standby mode is released 1: Standby mode is in effect

Set data in the power-save control register (PSC) in the following sequence.

<1> Set the power-save mode register (PSMR) (with the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<2> Prepare data in any one of the general-purpose registers to set to the specific register.

<3> Write data to the command register (PRCMD).

<4> Set the power-save control register (PSC) (with the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<5> Assert the NOP instructions (5 instructions (<5> to <9>).

Sample coding

```
<1> ST.B   r11, PSMR [r0]    ; Set PSMR register
<2> MOV    0×02, r10
<3> ST.B   r10, PRCMD [r0]   ; Write PRCMD register
<4> ST.B   r10, PSC [r0]     ; Set PSC register
<5> NOP                      ; Dummy instruction
<6> NOP                      ; Dummy instruction
<7> NOP                      ; Dummy instruction
<8> NOP                      ; Dummy instruction
<9> NOP                      ; Dummy instruction
(next instruction)           ; Execution routine after software STOP mode and IDLE mode release
```

No special sequence is required to read the specific register.

Cautions 1. A store instruction for the command register does not acknowledge interrupts. This coding is made on assumption that <3> and <4> above are executed by the program with consecutive store instructions. If another instruction is set between <3> and <4>, the above sequence may become ineffective when the interrupt is acknowledged by that instruction, and a malfunction of the program may result.

2. Although the data written to the PRCMD register is dummy data, use the same register as the general-purpose register used in specific register setting (<4>) for writing to the PRCMD register (<3>). The same method should be applied when using a general-purpose register for addressing.

3. At least 5 NOP instructions must be inserted after executing a store instruction to the PSC register to set software STOP or IDLE mode.

★ **4. Be sure to terminate all DMA transfers prior to the execution of the above sequence.**

9.5.3 HALT mode

(1) Setting and operation status

In HALT mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the operation clock of the CPU is stopped. Since the supply of clocks to on-chip peripheral I/O units other than the CPU continues, operation continues. The power consumption of the overall system can be reduced by setting the system to HALT mode while the CPU is idle.

The system is switched to HALT mode by the HALT instruction.

Although program execution stops in HALT mode, the contents of all registers, internal RAM, and ports are maintained in the state they were in immediately before HALT mode began. Also, operation continues for all on-chip peripheral I/O units (other than ports) that do not depend on CPU instruction processing. Table 9-2 shows the status of each hardware unit in HALT mode.

Caution If the HALT instruction is executed while an interrupt is being held pending, the HALT mode is set once but it is immediately released by the pending interrupt request.

Table 9-2. Operation Status in HALT Mode

Function	Operation Status
Clock generator	Operating
Internal system clock	Operating
CPU	Stopped
Ports	Maintained
On-chip peripheral I/O (excluding ports)	Operating
Internal data	All internal data such as CPU registers, statuses, data, and the contents of internal RAM are maintained in the state they were in immediately before HALT mode began.
D0 to D15	Operating
A0 to A25	
$\overline{RD}$, $\overline{WE}$, $\overline{OE}$, $\overline{BCYST}$	
$\overline{UWR}$, $\overline{LWR}$, $\overline{IORD}$, $\overline{IOWR}$	
LDQM, UDQM	
$\overline{CS0}$ to $\overline{CS7}$	
$\overline{LCAS}$, $\overline{UCAS}$	
$\overline{RAS1}$, $\overline{RAS3}$, $\overline{RAS4}$, $\overline{RAS6}$	
SDRAS	
SDCAS	
$\overline{REFRQ}$	
$\overline{HLDAK}$	
$\overline{HLDRQ}$	
$\overline{WAIT}$	
SELFREF	
SDCKE	
SDCLK	Clock output
CLKOUT	

(2) Release of HALT mode

HALT mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request, or $\overline{\text{RESET}}$ pin input.

(a) Release according to a non-maskable interrupt request or an unmasked maskable interrupt request

HALT mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request regardless of the priority. However, if the system is set to HALT mode during an interrupt servicing routine, operation will differ as follows.

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being serviced, HALT mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, HALT mode is released and the newly generated interrupt request is acknowledged.

Table 9-3. Operation After HALT Mode Is Released by Interrupt Request

Release Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

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

This is the same as a normal reset operation.

9.5.4 IDLE mode

(1) Setting and operation status

In IDLE mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the supply of internal system clocks is stopped which causes the overall system to stop.

When IDLE mode is released, the system can be switched to normal operation mode quickly because the oscillator's oscillation stabilization time or the PLL lockup time does not need to be secured.

The system is switched to IDLE mode by setting the PSC or PSMR register using a store instruction (ST or SST instruction) or a bit manipulation instruction (SET1, CLR1, or NOT1 instruction) (see **9.5.2 Control registers**).

In IDLE mode, program execution is stopped, and the contents of all registers, internal RAM, and ports are maintained in the state they were in immediately before execution stopped. The operation of on-chip peripheral I/O units (excluding ports) also is stopped.

Table 9-4 shows the status of each hardware unit in IDLE mode.

Table 9-4. Operation Status in IDLE Mode

Function	Operation Status
Clock generator	Operating
Internal system clock	Stopped
CPU	Stopped
Ports	Maintained
On-chip peripheral I/O (excluding ports)	Stopped
Internal data	All internal data such as CPU registers, statuses, data, and the contents of internal RAM are maintained in the state they were in immediately before IDLE mode began.
D0 to D15	High impedance
A0 to A25	
$\overline{RD}$, $\overline{WE}$, $\overline{OE}$, $\overline{BCYST}$	High-level output
$\overline{UWR}$, $\overline{LWR}$, $\overline{IORD}$, $\overline{IOWR}$	
LDQM, UDQM	
$\overline{CS0}$ to $\overline{CS7}$	
$\overline{LCAS}$, $\overline{UCAS}$	Operating
$\overline{RAS1}$, $\overline{RAS3}$, $\overline{RAS4}$, $\overline{RAS6}$	
SDRAS	
SDCAS	
$\overline{REFRQ}$	
$\overline{HLDAK}$	High-level output
$\overline{HLDRQ}$	Input (no sampling)
$\overline{WAIT}$	
SELFREF	
SDCKE	Low-level output
SDCLK	
CLKOUT	

(2) Release of IDLE mode

IDLE mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request (INTP1nn), or $\overline{\text{RESET}}$ pin input.

(a) Release according to a non-maskable interrupt request or an unmasked maskable interrupt request

IDLE mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request (INTP1nn) regardless of the priority. However, if the system is set to IDLE mode during a maskable interrupt servicing routine, operation will differ as follows (n = 0 to 3).

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being serviced, IDLE mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, IDLE mode is released and the newly generated interrupt request is acknowledged.

Table 9-5. Operation After IDLE Mode Is Released by Interrupt Request

Release Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

If the system is set to IDLE mode during an NMI servicing routine, IDLE mode is released, but the interrupt is not acknowledged (interrupt is held pending).

Interrupt servicing that is started when IDLE mode is released by NMI pin input is handled in the same way as normal NMI interrupt servicing that occurs during an emergency (because the NMI interrupt handler address is unique). Therefore, when a program must be able to distinguish between these two situations, a software status must be prepared in advance and that status must be set before setting the PSMR register using a store instruction or a bit manipulation instruction. By checking for this status during NMI interrupt servicing, an ordinary NMI can be distinguished from the processing that is started when IDLE mode is released by NMI pin input.

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

This is the same as a normal reset operation.

9.5.5 Software STOP mode

(1) Setting and operation status

In software STOP mode, the clock generator (oscillator and PLL synthesizer) is stopped. The overall system is stopped, and ultra-low power consumption is achieved in which only leakage current is lost.

The system is switched to software STOP mode by using a store instruction (ST or SST instruction) or bit manipulation instruction (SET1, CLR1, or NOT1 instruction) to set the PSC and PSMR registers (see **9.5.2 Control registers**).

When PLL mode and resonator connection mode (CESEL bit of CKC register = 1) are used, the oscillator's oscillation stabilization time must be secured after software STOP mode is released.

In both PLL and direct mode, following the release of software STOP mode, execution of the program is started after the count time of the time base counter has elapsed.

Although program execution stops in software STOP mode, the contents of all registers, internal RAM, and ports are maintained in the state they were in immediately before software STOP mode began. The operation of all on-chip peripheral I/O units (excluding ports) is also stopped.

Table 9-6 shows the status of each hardware unit in software STOP mode.

Table 9-6. Operation Status in Software STOP Mode

Function	Operation Status
Clock generator	Stopped
Internal system clock	Stopped
CPU	Stopped
Ports	Maintained[Note]
On-chip peripheral I/O (excluding ports)	Stopped
Internal data	All internal data such as CPU registers, statuses, data, and the contents of internal RAM are maintained in the state they were in immediately before software STOP mode began.
D0 to D15	High impedance
A0 to A25	
$\overline{RD}$, $\overline{WE}$, $\overline{OE}$, $\overline{BCYST}$	High-level output
$\overline{UWR}$, $\overline{LWR}$, $\overline{IORD}$, $\overline{IOWR}$	
LDQM, UDQM	
$\overline{CS0}$ to $\overline{CS7}$	
$\overline{LCAS}$, $\overline{UCAS}$	Operating
$\overline{RAS1}$, $\overline{RAS3}$, $\overline{RAS4}$, $\overline{RAS6}$	
SDRAS	
SDCAS	
$\overline{REFRQ}$	
$\overline{HLDAK}$	High-level output
$\overline{HLDRQ}$	Input (no sampling)
$\overline{WAIT}$	
SELFREF	
SDCKE	Low-level output
SDCLK	
CLKOUT	

Note When the V_{DD} value is within the operable range. However, even if it drops below the minimum operable voltage, as long as the data retention voltage V_{DDDR} is maintained, the contents of only the internal RAM will be maintained.

(2) Release of software STOP mode

Software STOP mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request (INTP1nn), or $\overline{\text{RESET}}$ pin input. Also, to release software STOP mode when PLL mode (CKSEL pin = low level) and resonator connection mode (CESEL bit of CKC register = 0) are used, the oscillator's oscillation stabilization time must be secured.

Moreover, the oscillation stabilization time must be secured even when an external clock is connected (CESEL bit = 1). See **9.4 PLL Lockup** for details.

(a) Release according to a non-maskable interrupt request or an unmasked maskable interrupt request

Software STOP mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request (INTP1nn) regardless of the priority. However, if the system is set to software STOP mode during an interrupt servicing routine, operation will differ as follows (n = 0 to 3).

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being servicing, software STOP mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, software STOP mode is released and the newly generated interrupt request is acknowledged.

Table 9-7. Operation After Software STOP Mode Is Released by Interrupt Request

Cancellation Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

If the system is set to software STOP mode during an NMI servicing routine, software STOP mode is released, but the interrupt is not acknowledged (interrupt is held pending).

Interrupt servicing that is started when software STOP mode is released by NMI pin input is handled in the same way as normal NMI interrupt servicing that occurs during an emergency (because the NMI interrupt handler address is unique). Therefore, when a program must be able to distinguish between these two situations, a software status must be prepared in advance and that status must be set before setting the PSMR register using a store instruction or a bit manipulation instruction.

By checking for this status during NMI interrupt servicing, an ordinary NMI can be distinguished from the servicing that is started when software STOP mode is released by NMI pin input.

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

This is the same as a normal reset operation.

9.6 Securing Oscillation Stabilization Time

9.6.1 Oscillation stabilization time security specification

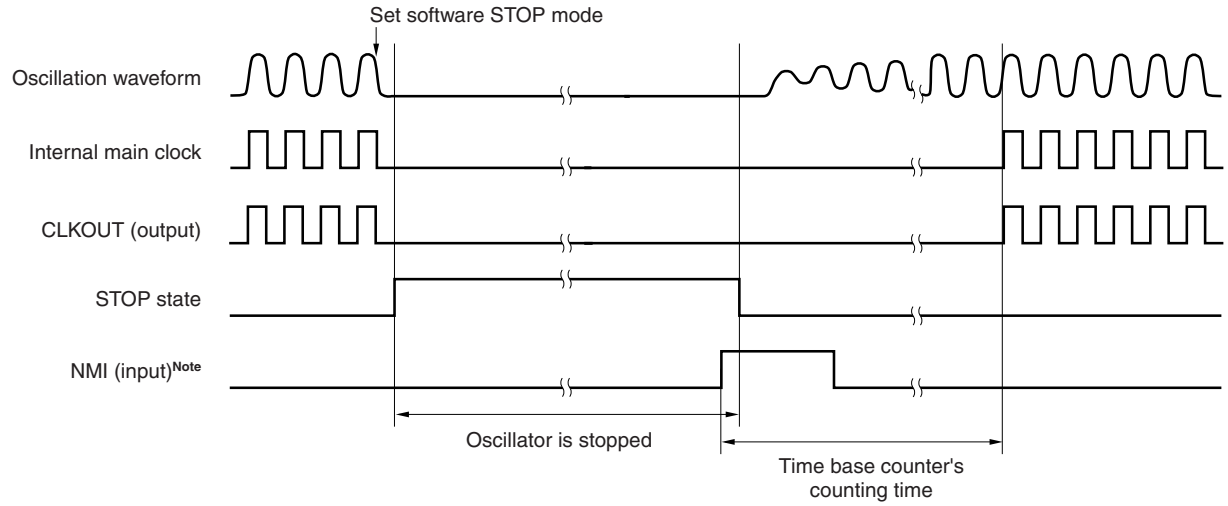
Two specification methods can be used to secure the time from when software STOP mode is released until the stopped oscillator stabilizes.

(1) Securing the time using an on-chip time base counter

Software STOP mode is released when a valid edge is input to the NMI pin or a maskable interrupt request is input (INTP1nn). Valid edge input to the pin causes the time base counter (TBC) to start counting, and the time until the clock output from the oscillator stabilizes is secured during that counting time.

Oscillation stabilization time = TBC counting time

After a fixed time, internal system clock output begins, and processing branches to the NMI interrupt or maskable interrupt handler address.



Note Valid edge: When specified as the rising edge.

The NMI pin should usually be set to an inactive level (for example, high level when the valid edge is specified as the falling edge) in advance.

Software STOP mode is immediately released if STOP mode is set by NMI valid edge input or maskable interrupt request input (INTP1nn) before the CPU acknowledges the interrupt.

If direct mode or external clock connection mode (CESEL bit of CKC register = 1) is used, program execution begins after the count time of the time base counter has elapsed.

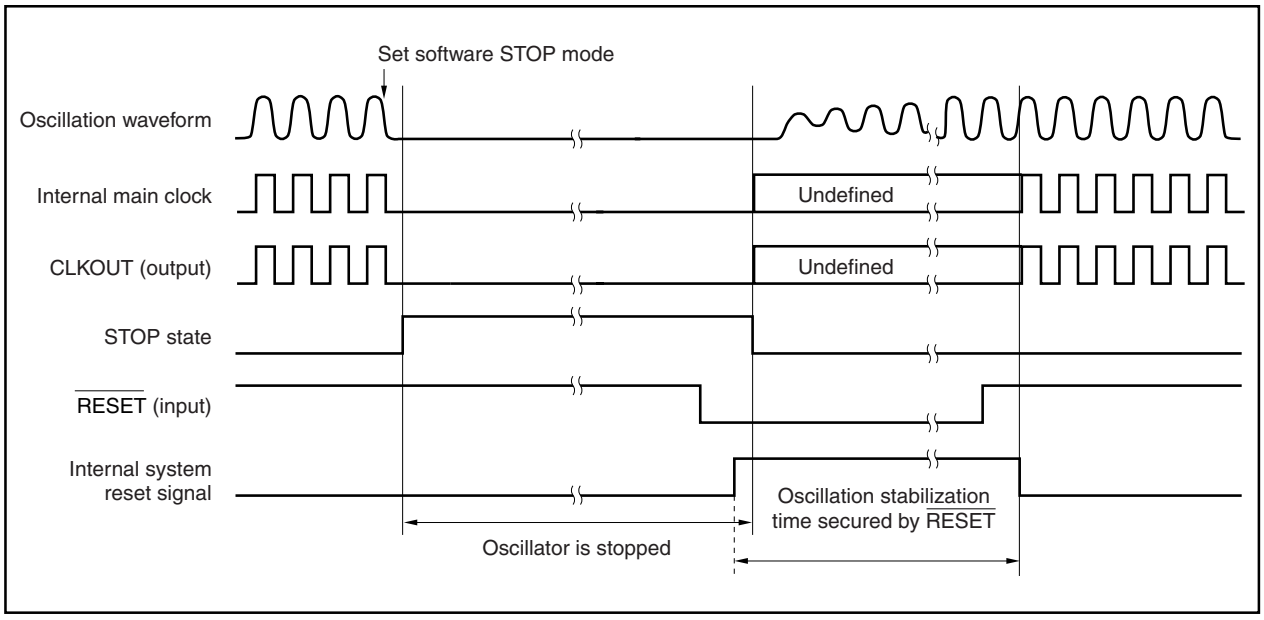
Also, even if PLL mode and resonator connection mode (CESEL bit of CKC register = 0) are used, program execution begins after the oscillation stabilization time is secured according to the time base counter.

(2) Securing the time according to the signal level width ($\overline{\text{RESET}}$ pin input)

Software STOP mode is released due to falling edge input to the $\overline{\text{RESET}}$ pin.

The time until the clock output from the oscillator stabilizes is secured according to the low-level width of the signal that is input to the pin.

The supply of internal system clocks begins after a rising edge is input to the $\overline{\text{RESET}}$ pin, and processing branches to the handler address used for a system reset.



9.6.2 Time base counter (TBC)

The time base counter (TBC) is used to secure the oscillator's oscillation stabilization time when software STOP mode is released.

When an external clock is connected (CESEL bit of CKC register = 1) or a resonator is connected (PLL mode and CESEL bit of CKC register = 0), the TBC counts the oscillation stabilization time after software STOP mode is released, and program execution begins after the count is completed.

The TBC count clock is selected according to the TBCS bit of the CKC register, and the next counting time can be set (reference).

Table 9-8. Counting Time Examples (ϕ = 10 × fxx)

TBCS Bit	Count Clock	Counting Time	
		fxx = 4.0000 MHz	fxx = 5.0000 MHz
		ϕ = 40.000 MHz	ϕ = 50.000 MHz
0	$f_{XX}/2^8$	16.3 ms	13.1 ms
1	$f_{XX}/2^9$	32.6 ms	26.2 ms

fxx: External oscillation frequency

ϕ: Internal system clock frequency

CHAPTER 10 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)

10.1 Timer C

10.1.1 Features (timer C)

Timer C is a 16-bit timer/counter that can perform the following operations.

- Interval timer function
- PWM output
- External signal cycle measurement

10.1.2 Function overview (timer C)

- 16-bit timer/counter
- Capture/compare common registers: 8
- Interrupt request sources
  - Capture/match interrupt requests: 8
  - Overflow interrupt requests: 4
- Timer/counter count clock sources: 2
  (Selection of external pulse input or internal system clock division)
- Either free-running mode or overflow stop mode can be selected as the operation mode when the timer/counter overflows
- Timer/counter can be cleared by a match of the timer/counter and a compare register
- External pulse outputs: 4

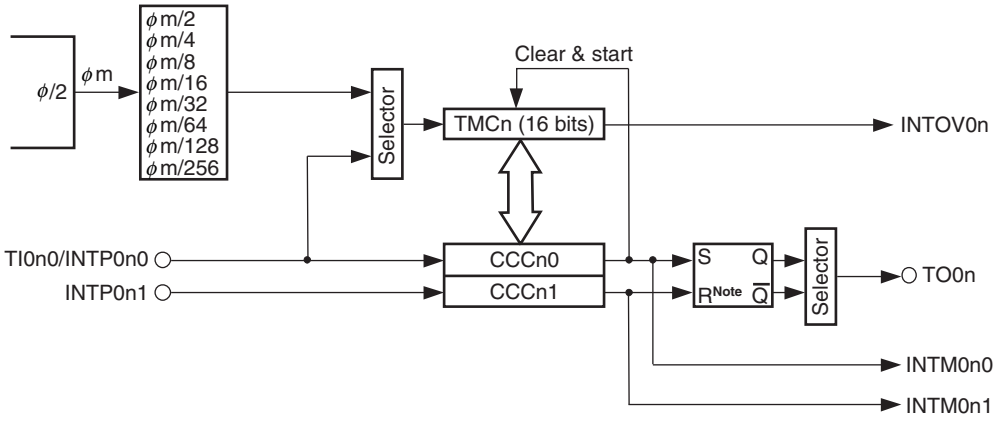
10.1.3 Basic configuration of timer C

Table 10-1. Timer C Configuration

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R	Other Functions
Timer C	ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, ϕ/256, ϕ/512,	TMC0	Read	INTOV00	–	–	–
		CCC00	Read/write	INTM000	INTP000	TO00 (S)	A/D conversion start trigger
		CCC01	Read/write	INTM001	INTP001	TO00 (R)	A/D conversion start trigger
		TMC1	Read	INTOV01	–	–	–
		CCC10	Read/write	INTM010	INTP010	TO01 (S)	A/D conversion start trigger
		CCC11	Read/write	INTM011	INTP011	TO01 (R)	A/D conversion start trigger
		TMC2	Read	INTOV02	–	–	–
		CCC20	Read/write	INTM020	INTP020	TO02 (S)	–
		CCC21	Read/write	INTM021	INTP021	TO02 (R)	–
		TMC3	Read	INTOV03	–	–	–
		CCC30	Read/write	INTM030	INTP030	TO03 (S)	–
		CCC31	Read/write	INTM031	INTP031	TO03 (R)	–

Remarks ϕ: Internal system clock
S/R: Set/reset

(1) Timer C (16-bit timer/counter)



Note Reset priority

Remark n = 0 to 3

10.1.4 Timer C

(1) Timers C0 to C3 (TMC0 to TMC3)

TMCn functions as a 16-bit free-running timer or as an event counter for an external signal. Besides being mainly used for cycle measurement, TMCn can be used as pulse output (n = 0 to 3).

TMCn is read-only in 16-bit units.

Cautions 1. The TMCn register can only be read. If the TMCn register is written, the subsequent operation is undefined.

2. If the TMCCAEn bit of the TMCCn register is cleared (0), a reset is performed asynchronously.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TMC0																	FFFFF600H	0000H
TMC1																	FFFFF610H	0000H
TMC2																	FFFFF620H	0000H
TMC3																	FFFFF630H	0000H

TMCn performs the count-up operations of an internal count clock or external count clock. Timer start and stop are controlled by the TMCCEn bit of timer mode control register Cn0 (TMCCn0) (n = 0 to 3).

The internal or external count clock is selected by the ETIn bit of timer mode control register Cn1 (TMCCn1) (n = 0 to 3).

(a) Selection of the external count clock

TMCn operates as an event counter.

When the ETIn bit of timer mode control register Cn1 (TMCCn1) is set (1), TMCn counts the valid edges of the external clock input (TI0n0), synchronized with the internal count clock. The valid edge is specified by valid edge select register Cn (SESCn) (n = 0 to 3).

Caution When the INTP0n0/TI0n0 pin is used as TI0n0 (external clock input pin), disable the INTP0n0 interrupt or set CCCn0 to compare mode (n = 0 to 3).

(b) Selection of the internal count clock

TMCn operates as a free-running timer.

When an internal clock is specified as the count clock by timer mode control register Cn1 (TMCCn1), TMCn is counted up for each input clock cycle specified by the CSn0 to CSn2 bits of the TMCCn0 register (n = 0 to 3).

Division by the prescaler can be selected for the count clock from among ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, ϕ/256, and ϕ/512 by the TMCCn0 register.

An overflow interrupt can be generated if the timer overflows. Also, the timer can be stopped following an overflow by setting the OSTn bit of the TMCCn1 register to 1.

Caution The count clock cannot be changed while the timer is operating.

The conditions when the TMCn register becomes 0000H are shown below.

(a) Asynchronous reset

- TMCCAEn bit of TMCCn0 register = 0
- Reset input

(b) Synchronous reset

- TMCCEn bit of TMCCn0 register = 0
- The CCCn0 register is used as a compare register, and the TMCn and CCCn0 registers match when clearing the TMCn register is enabled (CCLRn bit of the TMCCn1 register = 1)

(2) Capture/compare registers Cn0 and Cn1 (CCCn0 and CCCn1) (n = 0 to 3)

These capture/compare registers are 16-bit registers.

They can be used as capture registers or compare registers according to the CMSn0 and CMSn1 bit specifications of timer mode control register Cn1 (TMCCn1) (n = 0 to 3).

These registers can be read or written in 16-bit units. (However, write operations can only be performed in compare mode.)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CCC0n																	FFFFF602H, FFFFF604H	0000H
CCC1n																	FFFFF612H, FFFFF614H	0000H
CCC2n																	FFFFF622H, FFFFF624H	0000H
CCC3n																	FFFFF632H, FFFFF634H	0000H

Remark n = 0 and 1

(a) Setting these registers as capture registers (CMSn0 and CMSn1 of TMCCn1 = 0)

When these registers are set as capture registers, the valid edges of the corresponding external interrupt signals INTP0n0 and INTP0n1 are detected as capture triggers. The timer TMCn is synchronized with the capture trigger, and the value of TMCn is latched in the CCCn0 and CCCn1 registers (capture operation).

The valid edge of the INTP0n0 pin is specified (rising, falling, or both rising and falling edges) according to the IES0n01 and IES0n00 bits of the SESCn register, and the valid edge of the INTP0n1 pin is specified according to the IES0n11 and IES0n10 bits of the SESCn register (n = 0 to 3).

The capture operation is performed asynchronously to the count clock. The latched value is held in the capture register until another capture operation is performed (n = 0 to 3).

When the TMCCAEn bit of timer mode control register Cn0 (TMCCn0) is 0, 0000H is read (n = 0 to 3).

If these registers are specified as capture registers, an interrupt is generated by detecting the valid edge of signals INTP0n0 and INTP0n1 (n = 0 to 3).

(b) Setting these registers as compare registers (CMSn0 and CMSn1 of TMCCn1 = 1)

When these registers are set as compare registers, the TMCn and register values are compared for each timer count clock, and an interrupt is generated by a match. If the CCLRn bit of timer mode control register Cn1 (TMCCn1) is set (1), the TMCn value is cleared (0) at the same time as a match with the CCCn0 register (it is not cleared (0) by a match with the CCCn1 register) (n = 0 to 3).

A compare register is equipped with a set/reset function. The corresponding timer output (TO0n) is set or reset, in synchronization with the generation of a match signal (n = 0 to 3).

The interrupt selection source differs according to the function of the selected register.

Cautions 1. To write to capture/compare registers Cn0 and Cn1, always set the TMCCAEn bit to 1 first. If the TMCCAEn bit is 0, the data that is written will be invalid.

2. Write to capture/compare registers Cn0 and Cn1 after setting them as compare registers via TMCCn0 and TMCCn1 register settings. If they are set as capture registers (CMSn0 and CMSn1 bits of TMCCn1 register = 0), no data is written even if a write operation is performed to CCCn0 and CCCn1.

3. When these registers are set as compare registers, INTP0n0 and INTP0n1 cannot be used (n = 0 to 3).

10.1.5 Timer C control registers

(1) Timer mode control registers C00 to C30 (TMCC00 to TMCC30)

The TMCCn0 registers control the operation of TMCn (n = 0 to 3). These registers can be read or written in 8-bit or 1-bit units.

Cautions 1. The TMCCAEn and other bits cannot be set at the same time. The other bits and the registers of the other TMCn unit should always be set after the TMCCAEn bit has been set. Also, to use external pins related to the timer function when timer C is used, be sure to set (1) the TMCCAEn bit after setting the external pins to control mode.

★ **2. When conflict occurs between an overflow and a TMCCn0 register write, the OVFn bit value becomes the value written during the TMCCn0 register write (n = 0 to 3).**

3. The operation of the system is not guaranteed when bits 2 and 3 are set to a value other than 0.

(1/2)

	<7>	6	5	4	3	2	<1>	<0>	Address	After reset
TMCC00	OVF0	CS02	CS01	CS00	0	0	TMCCE0	TMCCAE0	FFFFF606H	00H
TMCC10	OVF1	CS12	CS11	CS10	0	0	TMCCE1	TMCCAE1	FFFFF616H	00H
TMCC20	OVF2	CS22	CS21	CS20	0	0	TMCCE2	TMCCAE2	FFFFF626H	00H
TMCC30	OVF3	CS32	CS31	CS30	0	0	TMCCE3	TMCCAE3	FFFFF636H	00H

Bit position	Bit name	Function
7	OVFn (n = 0 to 3)	Overflow This is a flag that indicates TMCn overflow (n = 0 to 3). 0: No overflow occurs 1: Overflow occurs When TMCn has counted up from FFFFH to 0000H, the OVFn bit becomes 1 and an overflow interrupt request (INTOV0n) is generated at the same time. However, if TMCn is cleared to 0000H after a match at FFFFH when the CCCn0 register is set to compare mode (CMSn0 bit of TMCCn1 register = 1) and clearing is enabled for a match when TMCn and CCCn0 are compared (CCLRn bit of TMCCn1 register = 1), then TMCn is considered to be cleared and the OVFn bit does not become 1. Also, no INTOV0n interrupt is generated. The OVFn bit retains the value 1 until 0 is written directly or until an asynchronous reset is performed because the TMCCAEn bit is 0. An interrupt operation due to an overflow is independent of the OVFn bit, and the interrupt request flag (OVIFn) for INTOV0n is not affected even if the OVFn bit is manipulated. If an overflow occurs while the OVFn bit is being read, the flag value changes, and the change is reflected when the next read operation occurs.

(2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>CSn2 to
CSn0
(n = 0 to 3)</td><td>Count Enable Select
Selects the TMCn internal count clock (n = 0 to 3).
<table>
<tr><th>CSn2</th><th>CSn1</th><th>CSn0</th><th>Count cycle</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>ϕ/4</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>ϕ/8</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>ϕ/16</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ϕ/32</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ϕ/64</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>ϕ/128</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>ϕ/256</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>ϕ/512</td></tr>
</table>
Caution The CSn2 to CSn0 bits must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMCCEn bit to 0. If these bits are overwritten during timer operation, operation cannot be guaranteed.</td></tr>
<tr><td>1</td><td>TMCCEn
(n = 0 to 3)</td><td>Count Enable
Controls the operation of TMCn (n = 0 to 3).
0: Count disabled (stops at 0000H and does not operate)
1: Counting operation is performed

Caution When TMCCEn = 0, the external pulse output (TO0n) becomes inactive (the active level of TO0n output is set by the ALVn bit of the TMCCn1 register).</td></tr>
<tr><td>0</td><td>TMCCAEn
(n = 0 to 3)</td><td>Clock Action Enable
Controls the internal count clock (n = 0 to 3).
0: The entire TMCn unit is asynchronously reset. The supply of clocks to the TMCn unit stops.
1: Clocks are supplied to the TMCn unit

Cautions 1. When the TMCCAEn bit is set to 0, the TMCn unit can be asynchronously reset.
2. When TMCCAEn = 0, the TMCn unit is in a reset state. Therefore, to operate TMCn, the TMCCAEn bit must be set to 1.
3. When the TMCCAEn bit is changed from 1 to 0, all registers of the TMCn unit are initialized. When TMCCAEn is set to 1 again, the TMCn unit registers must be set again.</td></tr>
</table>

(2) Timer mode control registers C01 to C31 (TMCC01 to TMCC31)

The TMCCn1 registers control the operation of TMCn (n = 0 to 3).

These registers can be read or written in 8-bit units.

Cautions 1. The various bits of the TMCCn1 register must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMCCEn bit of the TMCCn0 register to 0. If these bits are overwritten during timer operation, operation cannot be guaranteed (n = 0 to 3).

2. If the ENTn1 and ALVn bits are changed at the same time, a glitch (spike shaped noise) may be generated in the TO0n pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTn1 and ALVn bits do not change at the same time (n = 0 to 3).

3. TO0n output is not changed by an external interrupt signal (INTP0n0 or INTP0n1). To use the TO0n signal, specify that the capture/compare registers are compare registers (CMSn0 and CMSn1 bits of TMCCn1 register = 1) (n = 0 to 3).

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
TMCC01	OST0	ENT01	ALV0	ETI0	CCLR0	0	CMS01	CMS00	FFFFF608H	20H
TMCC11	OST1	ENT11	ALV1	ETI1	CCLR1	0	CMS11	CMS10	FFFFF618H	20H
TMCC21	OST2	ENT21	ALV2	ETI2	CCLR2	0	CMS21	CMS20	FFFFF628H	20H
TMCC31	OST3	ENT31	ALV3	ETI3	CCLR3	0	CMS31	CMS30	FFFFF638H	20H

Bit position	Bit name	Function
7	OSTn (n = 0 to 3)	Overflow Stop Sets the operation when TMCn has overflowed (n = 0 to 3). 0: After the overflow, counting continues (free-running mode) 1: After the overflow, the timer maintains the value 0000H, and counting stops (overflow stop mode). At this time, the TMCCEn bit of TMCCn0 remains at 1. Counting is restarted by writing 1 to the TMCCEn bit.
6	ENTn1 (n = 0 to 3)	Enable To Pin External pulse output is enabled/disabled (TO0n) (n = 0 to 3). 0: External pulse output is disabled. Output of the ALVn bit inactive level to the TO0n pin is fixed. The TO0n pin level is not changed even if a match signal from the corresponding compare register is generated. 1: External pulse output is enabled. A compare register match causes TO0n output to change. However, if capture mode is set, TO0n output does not change. The ALVn bit inactive level is output from the time when timer output is enabled until a match signal is first generated. **Caution If either CCCn0 or CCCn1 is specified as a capture register, the ENTn1 bit must be set to 0.**

(2/2)

Bit position	Bit name	Function
5	ALVn (n = 0 to 3)	Active Level Specifies the active level for external pulse output (TO0n) (n = 0 to 3). 0: Active level is low level 1: Active level is high level **Caution The initial value of the ALVn bit is 1.**
4	ETIn (n = 0 to 3)	External Input Specifies a switch between the external and internal count clock. 0: Specifies the input clock (internal). The count clock can be selected according to the CSn2 to CSn0 bits of TMCCn0 (n = 0 to 3). 1: Specifies the external clock (TI0n0). The valid edge can be selected according to the TESn1 and TESn0 bit specifications of SESCn (n = 0 to 3).
3	CCLRn (n = 0 to 3)	Compare Clear Enable Sets whether the clearing of TMCn is enabled or disabled during a compare operation (n = 0 to 3). 0: Clearing is disabled 1: Clearing is enabled (if CCCn0 and TMCn match during a compare operation, TMCn is cleared).
1	CMSn1 (n = 0 to 3)	Capture/Compare Mode Select Selects the operation mode of the capture/compare register (CCCn1) (n = 0 to 3). 0: The register operates as a capture register 1: The register operates as a compare register
0	CMSn0 (n = 0 to 3)	Capture/Compare Mode Select Selects the operation mode of the capture/compare register (CCCn0) (n = 0 to 3). 0: The register operates as a capture register 1: The register operates as a compare register

Remarks 1. A reset takes precedence for the flip-flop of the TO0n output (n = 0 to 3).

2. When the A/D converter is set to timer trigger mode, the match interrupt of the compare registers becomes a start trigger for A/D conversion, and the conversion operation begins. At this time, the compare register match interrupt also functions as a compare register match interrupt for the CPU. To prevent the generation of a compare register match interrupt for the CPU, disable interrupts using the interrupt mask bits (P00MK0, P00MK1, P01MK0, and P01MK1) of the interrupt control registers (P00IC0, P00IC1, P01IC0, and P01IC1).

(3) Valid edge select registers C0 to C3 (SESC0 to SESC3)

These registers specify the valid edge of an external interrupt request (INTP000, INTP001, INTP010, INTP011, INTP020, INTP021, INTP030, INTP031, and TI000 to TI030) from an external pin.

The rising edge, the falling edge, or both rising and falling edges can be specified as the valid edge independently for each pin.

Each of these registers can be read or written in 8-bit units.

Cautions 1. The various bits of the SESCn register must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMCCEn bit of the TMCCn0 register to 0. If the SESCn register is overwritten during timer operation, operation cannot be guaranteed.

2. The operation of the system is not guaranteed if bits 5 and 4 are set to a value other than 0.

	7	6	5	4	3	2	1	0	Address	After reset
SESC0	TES01	TES00	0	0	IES0011	IES0010	IES0001	IES0000	FFFFF609H	00H
	TI000				INTP001		INTP000			
SESC1	TES11	TES10	0	0	IES0111	IES0110	IES0101	IES0100	FFFFF619H	00H
	TI010				INTP011		INTP010			
SESC2	TES21	TES20	0	0	IES0211	IES0210	IES0201	IES0200	FFFFF629H	00H
	TI020				INTP021		INTP020			
SESC3	TES31	TES30	0	0	IES0311	IES0310	IES0301	IES0300	FFFFF639H	00H
	TI030				INTP031		INTP030			

Bit position	Bit name	Function
7, 6	TESn1, TESn0 (n = 0 to 3)	Edge Select Specifies the valid edge of the INTPn and TI000 to TI030 pins.
3, 2	IESn1, IESn0 (n = 001, 011, 021, 031)	
1, 0	IESn1, IESn0 (n = 000, 010, 020, 030)	

xESn1	xESn0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both edges

10.1.6 Timer C operation

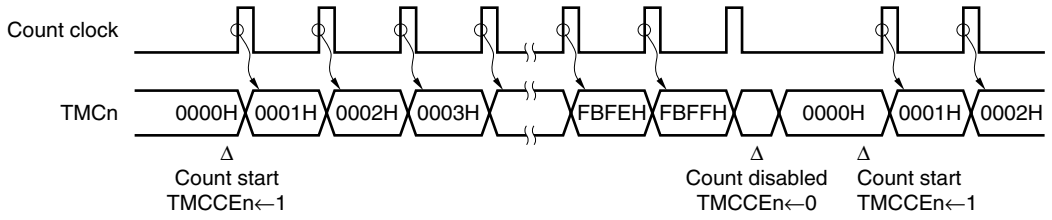
(1) Count operation

Timer C can function as a 16-bit free-running timer or as an external signal event counter. The setting for the type of operation is specified by timer mode control registers Cn0 and Cn1 (TMCCn0 and TMCCn1) (n = 0 to 3).

When it operates as a free-running timer, if the CCCn0 or CCCn1 register and the TMCn count value match, an interrupt signal is generated and the timer output signal (TO0n) can be set or reset. Also, a capture operation that holds the TMCn count value in the CCCn0 or CCCn1 register is performed, in synchronization with the valid edge that was detected from the external interrupt request input pin as an external trigger. The capture value is held until the next capture trigger is generated.

Caution When using the INTP0n0/TI0n0 pin as an external clock input pin (TI0n0), be sure to disable the INTP0n0 interrupt or set CCCn0 to compare mode (n = 0 to 3).

Figure 10-1. Basic Operation of Timer C



Remark n = 0 to 3

(2) Overflow

When the TMCn register has counted the count clock from FFFFH to 0000H, the OVFn bit of the TMCCn0 register is set (1), and an overflow interrupt (INTOV0n) is generated at the same time (n = 0 to 3). However, if the CCCn0 register is set to compare mode (CMSn0 bit = 1) and to the value FFFFH when match clearing is enabled (CCLRn bit = 1), then the TMCn register is considered to be cleared and the OVFn bit is not set (1) when the TMCn register changes from FFFFH to 0000H. Also, the overflow interrupt (INTOV0n) is not generated .

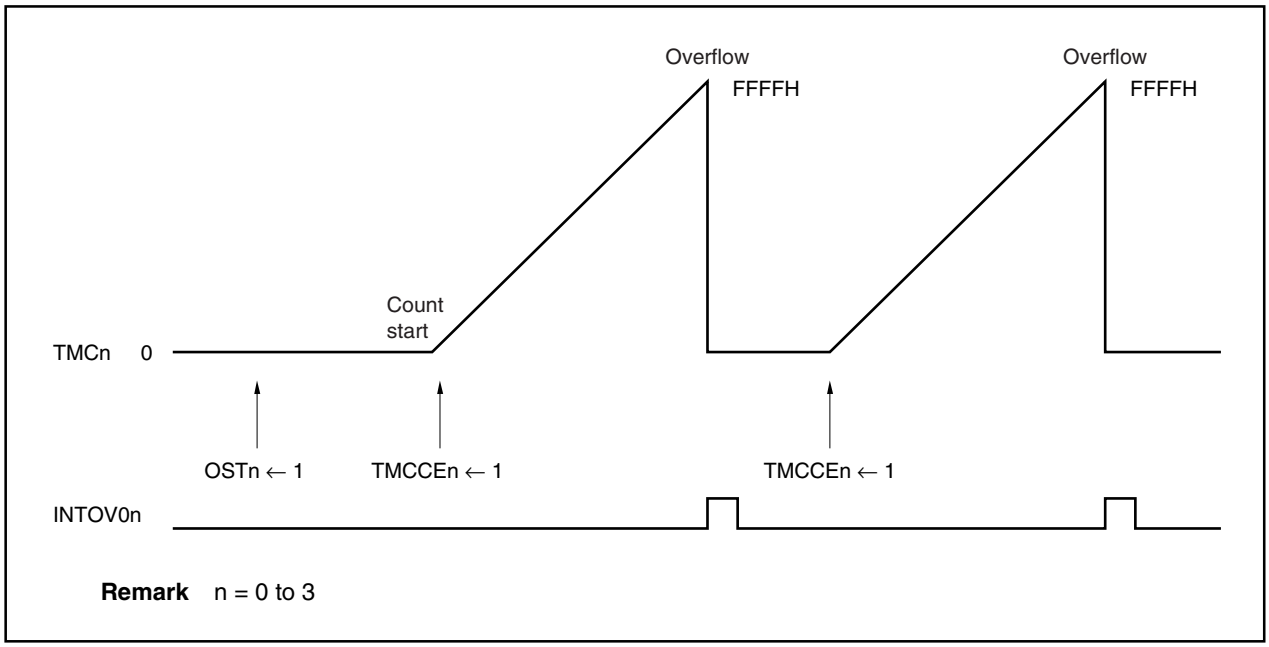
When the TMCn register is changed from FFFFH to 0000H because the TMCCEn bit changes from 1 to 0, the TMCn register is considered to be cleared, but the OVFn bit is not set (1) and no INTOV0n interrupt is generated.

Also, timer operation can be stopped after an overflow by setting the OSTn bit of the TMCCn1 register to 1. When the timer is stopped due to an overflow, the count operation is not restarted until the TMCCEn bit of the TMCCn0 register is set (1).

Operation is not affected even if the TMCCEn bit is set (1) during a count operation.

Remark n = 0 to 3

Figure 10-2. Operation After Overflow (When OSTn = 1)



Remark n = 0 to 3

(3) Capture operation

The TMCn register has two capture/compare registers. These are the CCCn0 register and the CCCn1 register. A capture operation or a compare operation is performed according to the settings of both the CMSn1 and CMSn0 bits of the TMCCn1 register. If the CMSn1 and CMSn0 bits of the TMCCn1 register are set to 0, the register operates as a capture register.

A capture operation that captures and holds the TMCn count value asynchronously relative to the count clock is performed in synchronization with an external trigger. The valid edge that is detected from an external interrupt request input pin (INTP0n0 or INTP0n1) is used as an external trigger (capture trigger). The TMCn count value during counting is captured and held in the capture register, in synchronization with that capture trigger signal. The capture register value is held until the next capture trigger is generated.

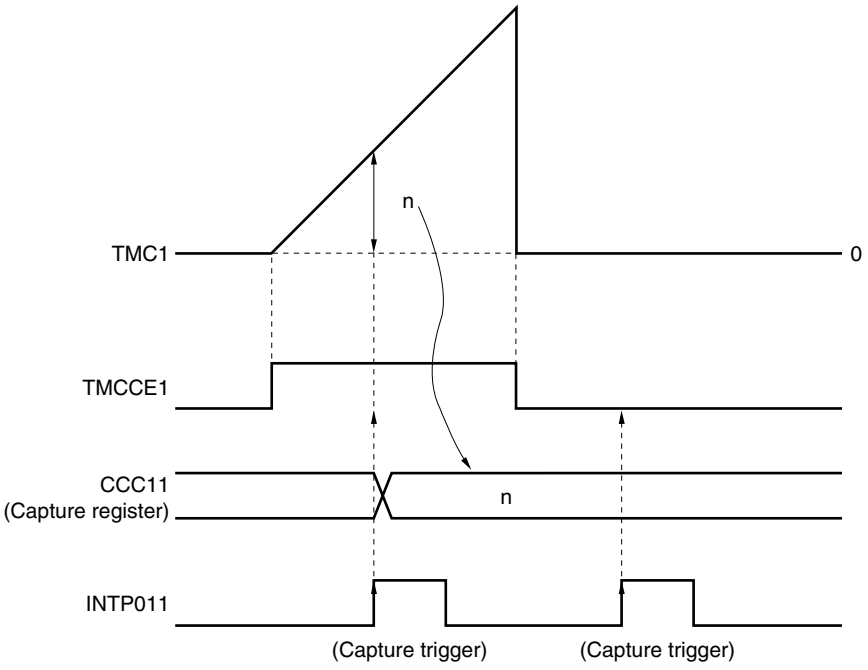
Also, an interrupt request (INTM0n0 or INTM0n1) is generated by INTP0n0 or INTP0n1 signal input.

The valid edge of the capture trigger is set by valid edge select register Cn (SESCn).

If both the rising and falling edges are set as capture triggers, the input pulse width from an external source can be measured. Also, if only one of the edges is set as the capture trigger, the input pulse cycle can be measured.

Remark n = 0 to 3

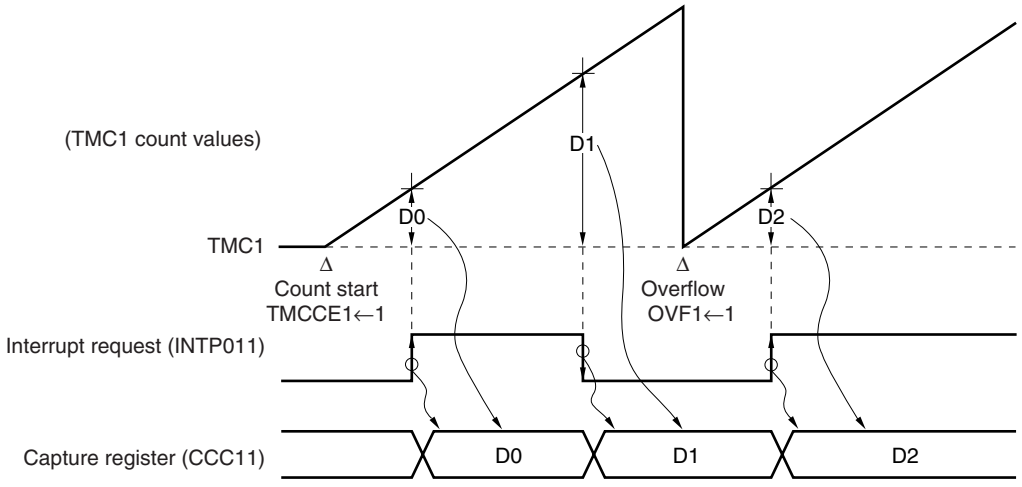
Figure 10-3. Capture Operation Example



Remarks 1. When the TMCCE1 bit is 0, no capture operation is performed even if INTP011 is input.

2. Valid edge of INTP011: Rising edge

Figure 10-4. TMC1 Capture Operation Example (When Both Edges Are Specified)



Remark D0 to D2: TMC1 count values

(4) Compare operation

The TMCn register has two capture/compare registers. These are the CCCn0 register and the CCCn1 register. A capture operation or a compare operation is performed according to the settings of both the CMSn1 and CMSn0 bits of the TMCCn1 register. If the CMSn1 and CMSn0 bits of the TMCCn1 register are set to 1, the register operates as a compare register.

A compare operation that compares the value that was set in the compare register and the TMCn count value is performed.

If the TMCn count value matches the value of the compare register, which had been set in advance, a match signal is sent to the output controller. The match signal causes the timer output pin (TO0n) to change and an interrupt request signal (INTCCn) to be generated at the same time.

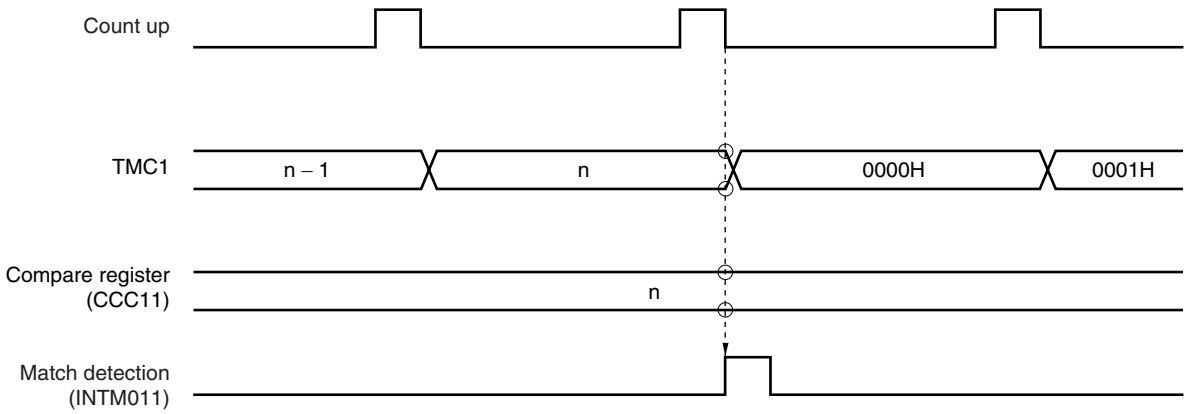
If the CCCn0 register is set to 0000H, the 0000H after the TMCn register counts up from FFFFH to 0000H is ★ judged as a match. In this case, the TMCn register value is cleared (0) at the next count timing, however, this 0000H is not judged as a match. Also, the 0000H when the TMCn register begins counting is not judged as a match.

If match clearing is enabled (CCLRn bit = 1) for the CCCn0 register, the TMCn register is cleared when a match with the TMCn register occurs during a compare operation.

Remark n = 0 to 3

★ **Figure 10-5. Compare Operation Example (1/2)**

(a) When CCLR1 = 1 and CCC11 is other than 0000H

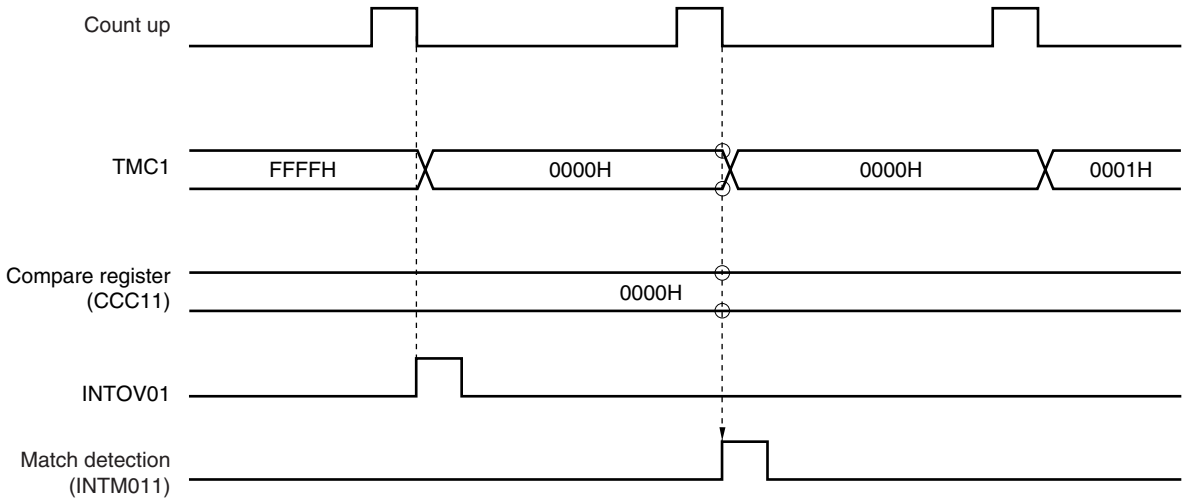


Remarks 1. The match is detected immediately after the count-up, and the match detection signal is generated.

2. n ≠ 0000H

★ **Figure 10-5. Compare Operation Example (2/2)**

(b) When CCLR1 = 1 and CCC11 is 0000H



Remark The match is detected immediately after the count-up, and the match detection signal is generated.

(5) External pulse output

Timer C has four timer output pins (TO0n).

An external pulse output (TO0n) is generated when a match of the two compare registers (CCCn0 and CCCn1) and the TMCn register is detected.

If a match is detected when the TMCn count value and the CCC0n0 value are compared, the output level of the TO0n pin is set. Also, if a match is detected when the TMCn count value and the CCC0n1 value are compared, the output level of the TO0n pin is reset.

The output level of the TO0n pin can be specified by the TMCCn1 register.

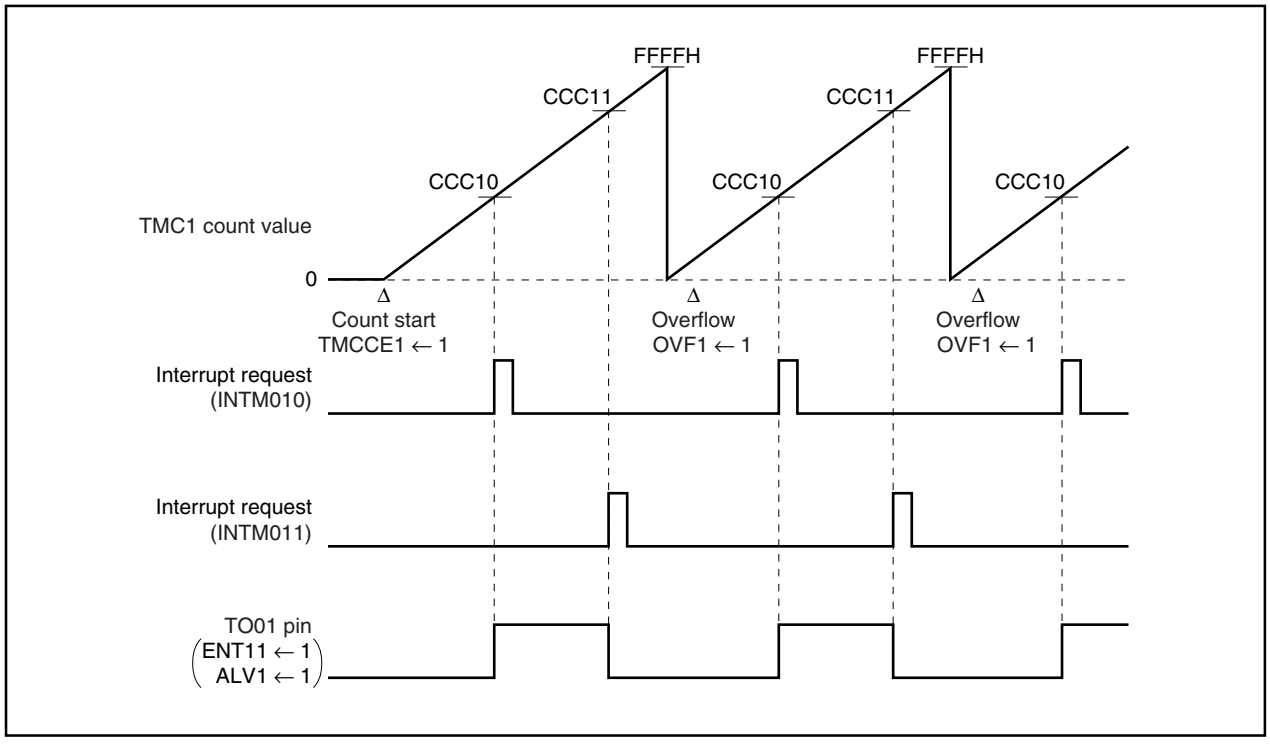
Remark n = 0 to 3

Table 10-2. TO0n Output Control

ENTn1	ALVn	TO0n Output
0	0	1
0	1	0
1	0	When the CCCn0 register is matched: 0 When the CCCn1 register is matched: 1
1	1	When the CCCn0 register is matched: 1 When the CCCn1 register is matched: 0

Remark n = 0 to 3

Figure 10-6. TMC1 Compare Operation Example (Set/Reset Output Mode)



10.1.7 Application examples (timer C)

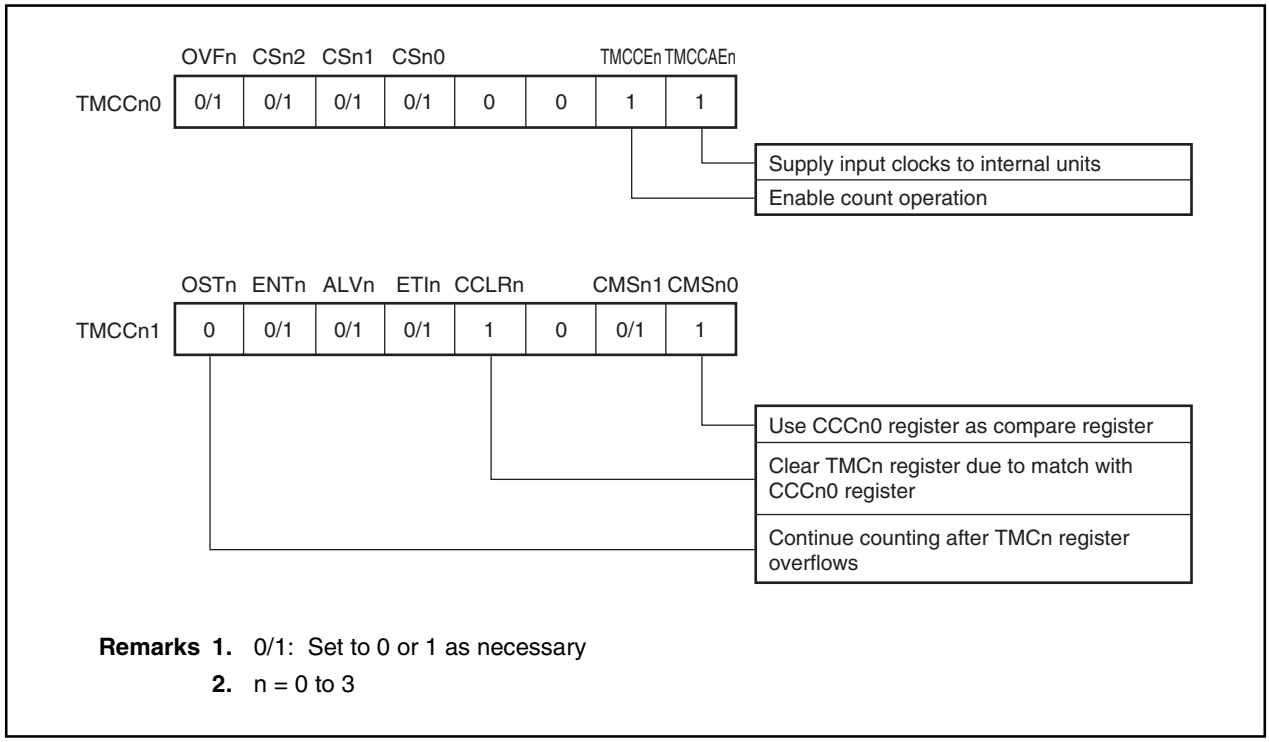
(1) Interval timer

By setting the TMCCn0 and TMCCn1 registers as shown in Figure 10-7, timer C operates as an interval timer that repeatedly generates interrupt requests with the value that was preset in the CCCn0 register as the interval.

When the counter value of the TMCn register matches the setting value of the CCCn0 register, the TMCn register is cleared (0000H) and an interrupt request signal (INTM0n0) is generated at the same time that the count operation resumes.

Remark n = 0 to 3

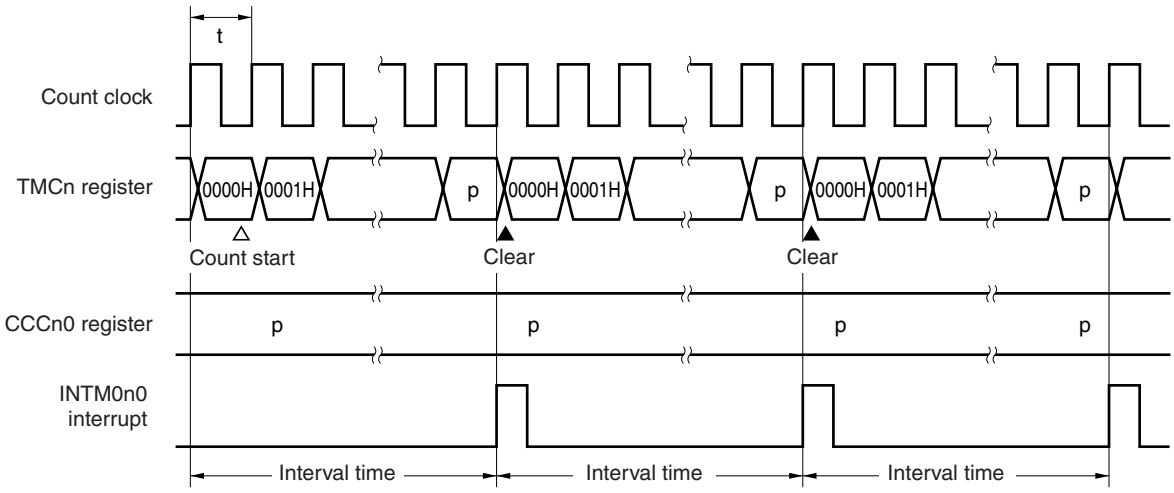
Figure 10-7. Contents of Register Settings When Timer C Is Used as Interval Timer



Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0 to 3

Figure 10-8. Interval Timer Operation Timing Example



Remarks 1. p: Setting value of CCCn0 register (0000H to FFFFH)

t: Count clock cycle

Interval time = (p + 1) × t

2. n = 0 to 3

(2) PWM output

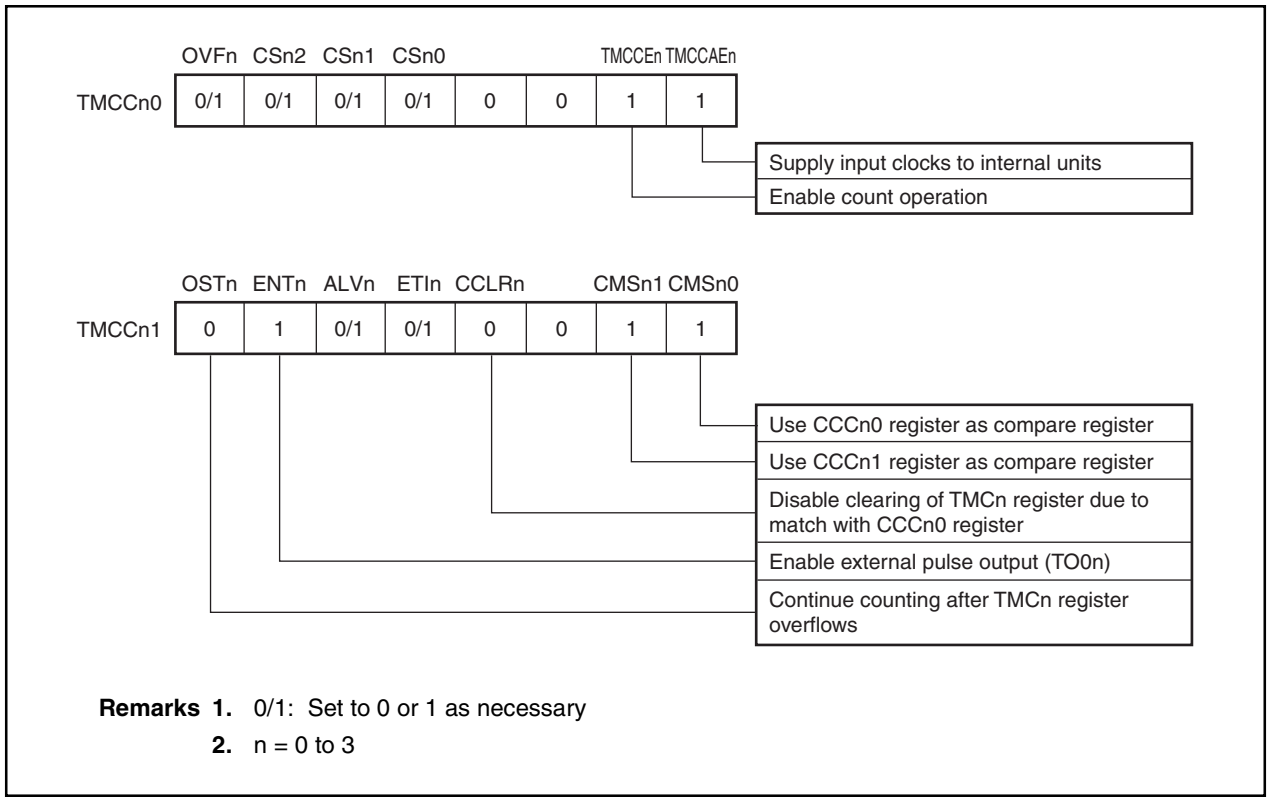
By setting the TMCCn0 and TMCCn1 registers as shown in Figure 10-9, timer C can output PWM of an arbitrary frequency with the values that were preset in the CCCn0 and CCCn1 registers determining the intervals.

When the counter value of the TMCn register matches the setting value of the CCCn0 register, the TO0n output becomes active. Then, when the counter value of the TMCn register matches the setting value of the CCCn1 register, the TO0n output becomes inactive. This enables PWM of an arbitrary frequency to be output. When the setting value of the CCCn0 register and the setting value of the CCCn1 register are the same, the TO0n output remains inactive and does not change.

The active level of the TO0n output can be set by the ALVn bit of the TMCCn1 register.

Remark n = 0 to 3

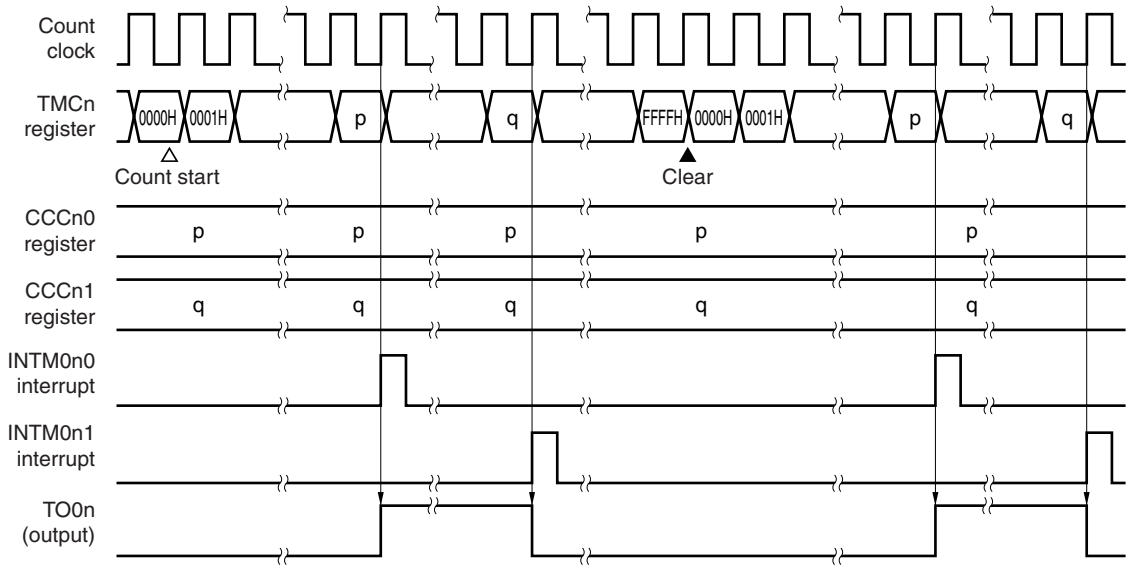
Figure 10-9. Contents of Register Settings When Timer C Is Used for PWM Output



Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0 to 3

Figure 10-10. PWM Output Timing Example



Remarks 1. p: Setting value of CCCn0 register (0000H to FFFFH)
q: Setting value of CCCn1 register (0000H to FFFFH)
$p \neq q$

2. In this example, the active level of the TO0n output is set to high level.

3. n = 0 to 3

(3) Cycle measurement

By setting the TMCCn0 and TMCCn1 registers as shown in Figure 10-11, timer C can measure the cycle of signals input to the INTP0n0 or INTP0n1 pin.

The valid edge of the INTP0n0 pin is selected according to the IES0n01 and IES0n00 bits of the SESCn register, and the valid edge of the INTP0n1 pin is selected according to the IES0n11 and IES0n10 bits of the SESCn register. Either the rising edge, the falling edge, or both edges can be selected as the valid edges of both pins.

If the CCCn0 register is set as a capture register, the valid edge input of the INTP0n0 pin is set as the trigger for capturing the TMCn register value in the CCCn0 register. When this value is captured, an INTM0n0 interrupt is generated.

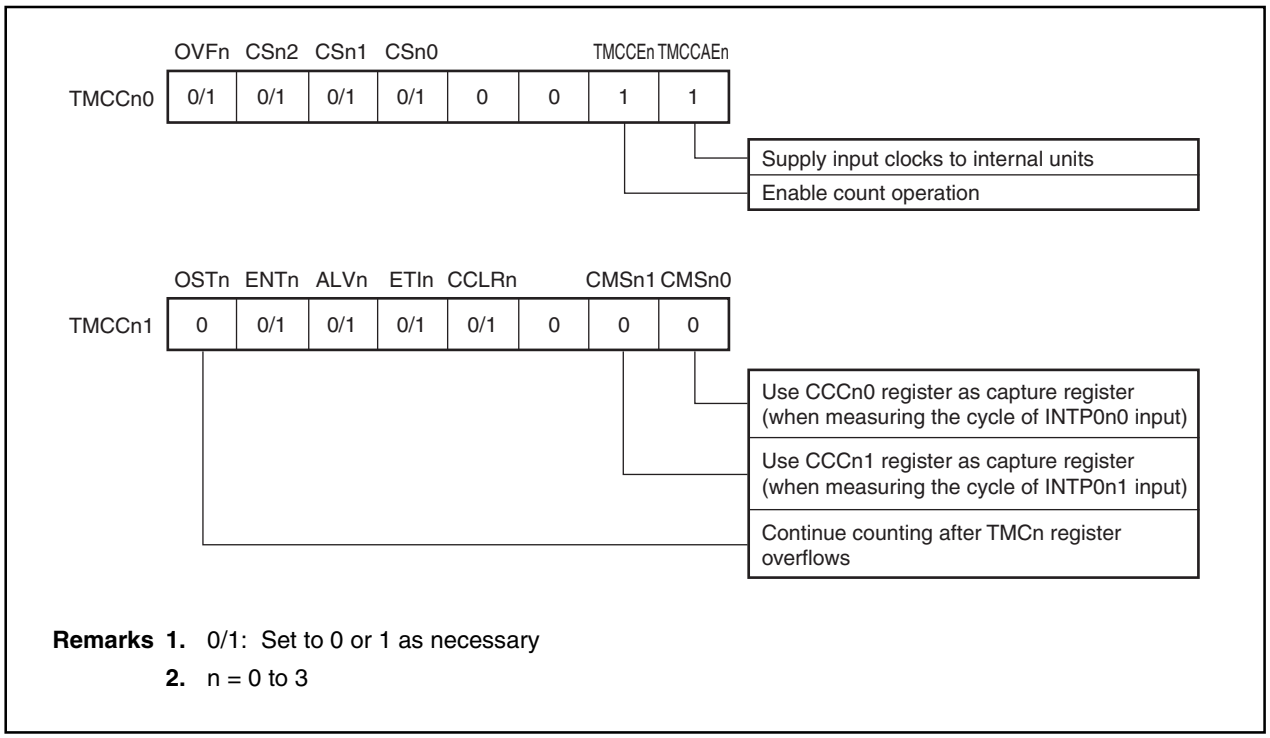
Similarly, if the CCCn1 register is set as a capture register, the valid edge input of the INTP0n1 pin is set as the trigger for capturing the TMCn register value in the CCCn1 register. When this value is captured, an INTM0n1 interrupt is generated.

The cycle of signals input to the INTP0n0 pin is calculated by obtaining the difference between the TMCn register's count value (Dx) that was captured in the CCCn0 register according to the x-th valid edge input of the INTP0n0 pin and the TMCn register's count value (D(x+1)) that was captured in the CCCn0 register according to the (x+1)-th valid edge input of the INTP0n0 pin and multiplying the value of this difference by the cycle of the clock control signal.

The cycle of signals input to the INTP0n1 pin is calculated by obtaining the difference between the TMCn register's count value (Dx) that was captured in the CCCn1 register according to the x-th valid edge input of the INTP0n1 pin and the TMCn register's count value (D(x+1)) that was captured in the CCCn1 register according to the (x+1)-th valid edge input of the INTP0n1 pin and multiplying the value of this difference by the cycle of the clock control signal.

Remark n = 0 to 3

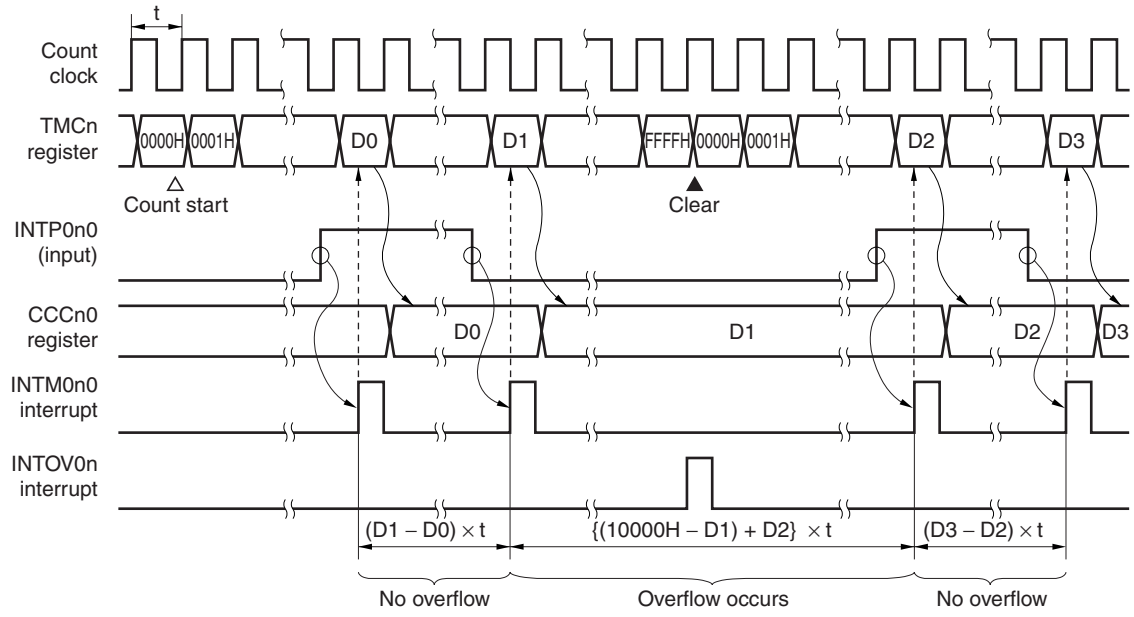
Figure 10-11. Contents of Register Settings When Timer C Is Used for Cycle Measurement



Remarks 1. 0/1: Set to 0 or 1 as necessary

2. n = 0 to 3

Figure 10-12. Cycle Measurement Operation Timing Example



Caution An overflow must not be generated more than once between the 1st and 2nd INTM0n0 interrupts.

Remarks 1. D0 to D3: TMCn register count values
t: Count clock cycle

2. In this example, the valid edge of the INTP0n0 input has been set to both edges (rising and falling).

3. n = 0 to 3

10.1.8 Cautions (timer C)

Various cautions concerning timer C are shown below.

(1) If a conflict occurs between the reading of the CCCn0 register and a capture operation when the CCCn0 register is used in capture mode, an external trigger (INTP0n0) valid edge is detected and an external interrupt request signal (INTM0n0) is generated, however, the timer value is not stored in the CCCn0 register.

(2) If a conflict occurs between the reading of the CCCn1 register and a capture operation when the CCCn1 register is used in capture mode, an external trigger (INTP0n1) valid edge is detected and an external interrupt request signal (INTM0n1) is generated, however, the timer value is not stored in the CCCn1 register.

(3) The following bits and registers must not be rewritten during operation (TMCCEn = 1).
- CSn2 to CSn0 bits of TMCCn0 register
- TMCCn1 register
- SESCn register

(4) The TMCCAEn bit of the TMCCn0 register is a TMCn reset signal. To use TMCn, first set (1) the TMCCAEn bit.

(5) The analog noise elimination time + two cycles of the input clock are required to detect the valid edge of the external interrupt request signal (INTP0n0 or INTP0n1). Therefore, edge detection will not be performed normally for changes that are less than the analog noise elimination time + two cycles of the input clock. For details of analog noise elimination, refer to **7.3.8 Noise elimination**.

(6) The operation of an external interrupt request signal (INTM0n0 or INTM0n1) is automatically determined according to the operating state of the capture/compare register. When the capture/compare register is used for a capture operation, the external interrupt request signal is used for valid edge detection. When the capture/compare register is used for a compare operation, the external interrupt request signal is used for an interrupt indicating a match with the TMCn register.

(7) If the ENTn1 and ALVn bits are changed at the same time, a glitch (spike shaped noise) may be generated in the TO0n pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTn1 and ALVn bits are not changed at the same time.

Remark n = 0 to 3

10.2 Timer D

10.2.1 Features (timer D)

Timer D functions as a 16-bit interval timer.

10.2.2 Function overview (timer D)

- 16-bit interval timer
- Compare registers: 4
- Interrupt request sources: 4 sources
- Count clock selected from divisions of internal system clock

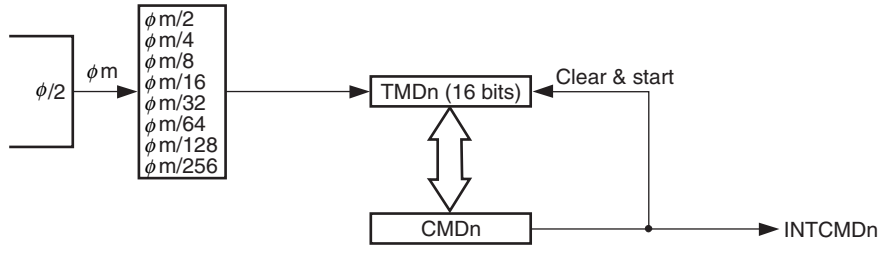
10.2.3 Basic configuration of timer D

Table 10-3. Timer D Configuration

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R	Other Functions
Timer D	ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, ϕ/256, ϕ/512	TMD0	Read	–	–	–	–
		CMD0	Read/write	INTCMD0	–	–	–
		TMD1	Read	–	–	–	–
		CMD1	Read/write	INTCMD1	–	–	–
		TMD2	Read	–	–	–	–
		CMD2	Read/write	INTCMD2	–	–	–
		TMD3	Read	–	–	–	–
		CMD3	Read/write	INTCMD3	–	–	–

Remark ϕ: Internal system clock
S/R: Set/reset

(1) Timer D (16-bit timer/counter)



Remark n = 0 to 3

10.2.4 Timer D

(1) Timers D0 to D3 (TMD0 to TMD3)

TMDn is a 16-bit timer. It is mainly used as an interval timer for software (n = 0 to 3).

Starting and stopping TMDn is controlled by the TMDCEn bit of the timer mode control register Dn (TMCDn) (n = 0 to 3).

Division by the prescaler can be selected for the count clock from among ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, ϕ/256, and ϕ/512 by the CSn0 to CSn2 bits of the TMCDn register.

TMDn is read-only in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TMD0																	FFFFF540H	0000H
TMD1																	FFFFF550H	0000H
TMD2																	FFFFF560H	0000H
TMD3																	FFFFF570H	0000H

The conditions for which the TMDn register becomes 0000H are shown below (n = 0 to 3).

- Reset input
- TMDCAEn bit = 0
- TMDCEn bit = 0
- Match of TMDn register and CMDn register
- Overflow

Cautions 1. If the TMDCAEn bit of the TMCDn register is cleared (0), a reset is performed asynchronously.

2. If the TMDCEn bit of the TMCDn register is cleared (0), a reset is performed, in synchronization with the internal clock. Similarly, a synchronized reset is performed after a match with the CMDn register and after an overflow.

3. The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the TMDCEn bit is cleared (0).

4. Up to ϕ/4 clocks are required after a value is set in the TMDCEn bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent cycles.

5. After a compare match is generated, the timer is cleared at the next count clock. Therefore, if the division ratio is large, the timer value may not be zero even if the timer value is read immediately after a match interrupt is generated.

(2) Compare registers D0 to D3 (CMD0 to CMD3)

CMDn and the TMDn register count value are compared, and an interrupt request signal (INTCMDn) is generated when a match occurs. TMDn is cleared, in synchronization with this match. If the TMDCAEn bit of the TMCDn register is set to 0, a reset is performed asynchronously, and the registers are initialized (n = 0 to 3).

The CMDn registers are configured with a master/slave configuration. When a CMDn register is written, data is first written to the master register and then the master register data is transferred to the slave register. In a compare operation, the slave register value is compared with the count value of the TMDn register. When a CMDn register is read, data in the master side is read out.

CMDn can be read or written in 16-bit units.

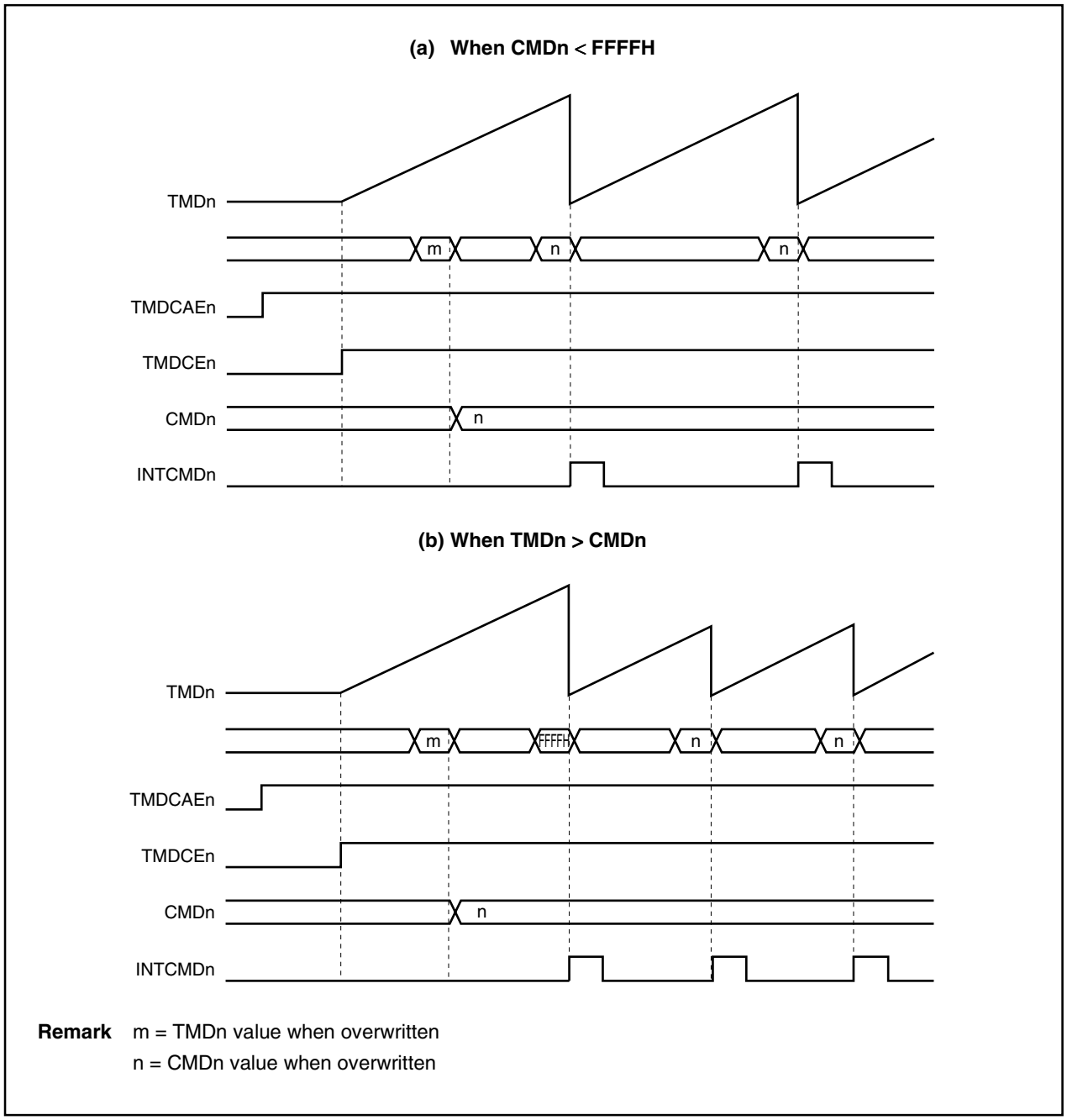
Cautions 1. A write operation to a CMDn register requires ϕ/4 clocks until the value that was set in the CMDn register is transferred to internal units. When writing continuously to the CMDn register, be sure to reserve a time interval of at least ϕ/4 clocks.

2. The CMDn register can be overwritten only once in a single TMDn register cycle (from 0000H until an INTCMDn interrupt is generated due to a match of the TMDn register and CMDn register). If this cannot be secured by the application, make sure that the CMDn register is not overwritten during timer operation.

3. Note that a match signal will be generated after an overflow if a value less than the counter value is written in the CMDn register during TMDn register operation (Figure 10-13).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
CMD0																	FFFFF542H	0000H
CMD1																	FFFFF552H	0000H
CMD2																	FFFFF562H	0000H
CMD3																	FFFFF572H	0000H

Figure 10-13. Example of Timing During TMDn Operation



Remark m = TMDn value when overwritten
n = CMDn value when overwritten

10.2.5 Timer D control registers

(1) Timer mode control registers D0 to D3 (TMCD0 to TMCD3)

The TMCDn registers control the operation of timer Dn (n = 0 to 3).

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

Caution The TMDCAEn and other bits cannot be set at the same time. The other bits and the registers of the other TMDn units should always be set after the TMDCAEn bit has been set.

	7	6	5	4	3	2	<1>	<0>	Address	After reset
TMCD0	0	CS02	CS01	CS00	0	0	TMDCE0	TMDCAE0	FFFFF544H	00H
TMCD1	0	CS12	CS11	CS10	0	0	TMDCE1	TMDCAE1	FFFFF554H	00H
TMCD2	0	CS22	CS21	CS20	0	0	TMDCE2	TMDCAE2	FFFFF564H	00H
TMCD3	0	CS32	CS31	CS30	0	0	TMDCE3	TMDCAE3	FFFFF574H	00H

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>CSn2 to
CSn0
(n = 0 to 3)</td><td>Count Enable Select
Selects the TMDn internal count clock cycle (n = 0 to 3).
<table>
<tr><th>CS2</th><th>CS1</th><th>CS0</th><th>Count cycle</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>ϕ/4</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>ϕ/8</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>ϕ/16</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ϕ/32</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ϕ/64</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>ϕ/128</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>ϕ/256</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>ϕ/512</td></tr>
</table>
Caution The CSn2 to CSn0 bits must not be changed during timer operation. If they are to be changed, they must be changed after setting the TMDCEn bit to 0. If these bits are overwritten during timer operation, operation cannot be guaranteed.</td></tr>
<tr><td>1</td><td>TMDCEn
(n = 0 to 3)</td><td>Count Enable
Controls the operation of TMDn (n = 0 to 3).
0: Count disabled (stops at 0000H and does not operate)
1: Counting operation is performed

Caution TMDCEn bit is not cleared even if a match is detected by the compare operation. To stop the count operation, clear the TMDCEn bit.</td></tr>
<tr><td>0</td><td>TMDCAEn
(n = 0 to 3)</td><td>Clock Action Enable
Controls the internal count clock (n = 0 to 3).
0: The entire TMDn unit is reset asynchronously. The supply of input clocks to the TMDn unit stops.
1: Input clocks are supplied to the TMDn unit

Cautions 1. When the TMDCAEn bit is set to 0, the TMDn unit can be asynchronously reset.
2. When TMDCAEn = 0, the TMDn unit is in a reset state. Therefore, to operate TMDn, the TMDCAEn bit must be set to 1.</td></tr>
</table>

10.2.6 Timer D operation

(1) Compare operation

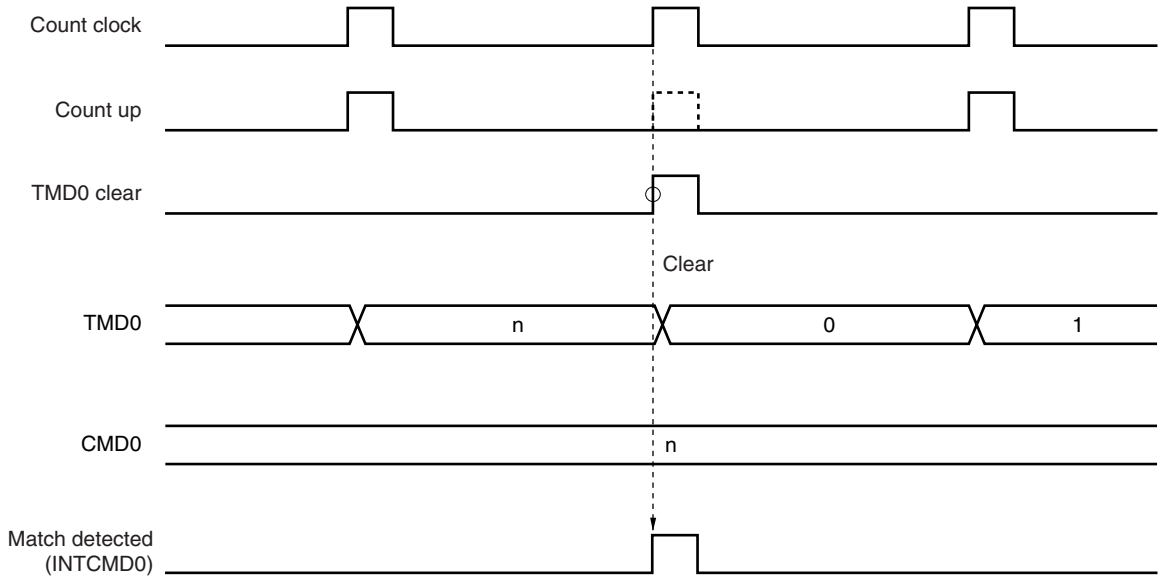
TMDn can be used for a compare operation in which the value that was set in a compare register (CMDn) is compared with the TMDn count value (n = 0 to 3).

If a match is detected by the compare operation, an interrupt (INTCMDn) is generated. The generation of the interrupt causes TMDn to be cleared (0) at the next count timing. This function enables timer D to be used as an interval timer.

CMDn can also be set to 0. In this case, when an overflow occurs and TMDn becomes 0, a match is detected and INTCMDn is generated. Although the TMDn value is cleared (0) at the next count timing, INTCMDn is not generated by this match.

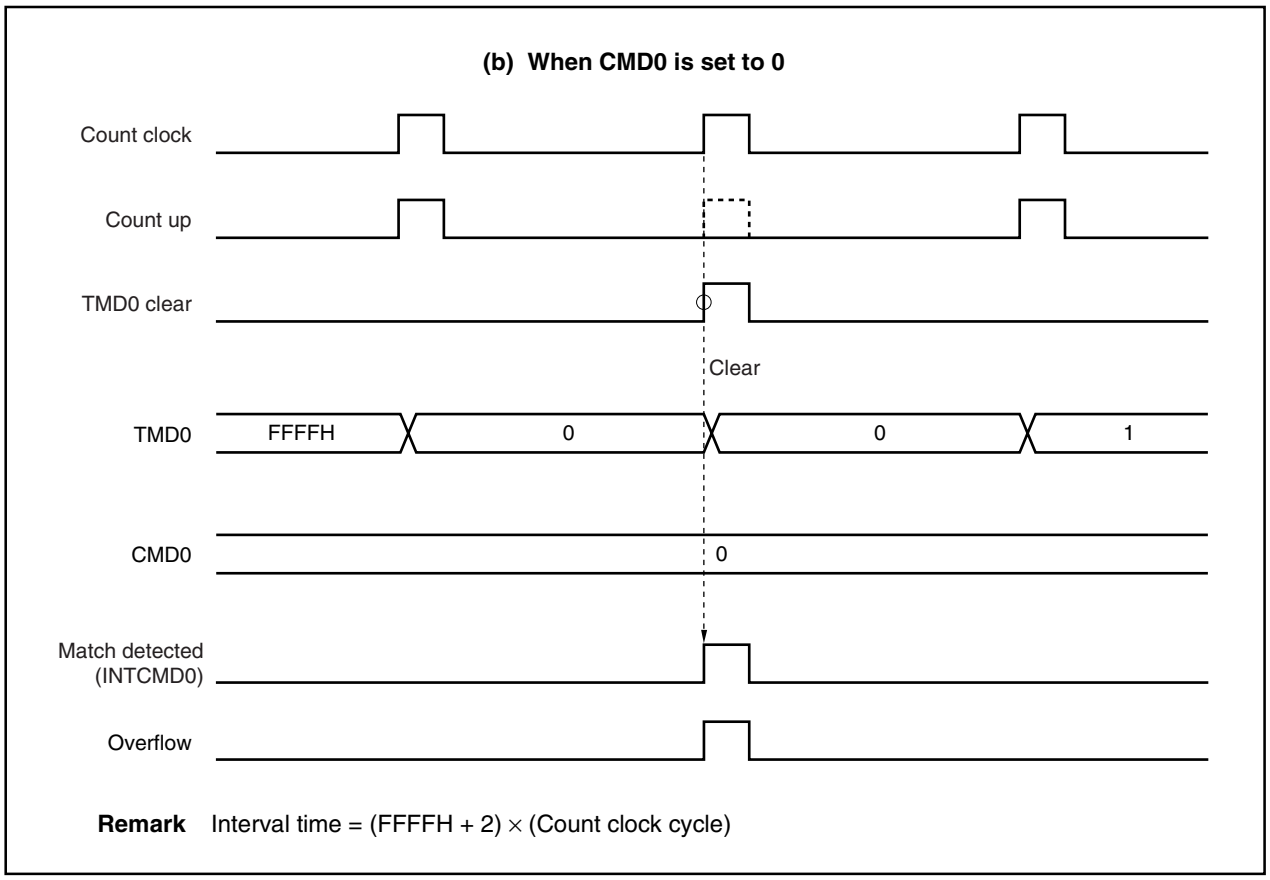
★ **Figure 10-14. TMD0 Compare Operation Example (1/2)**

(a) When CMD0 is set to n (non-zero)



Remark Interval time = (n + 1) × (Count clock cycle)
n = 1 to 65,536 (FFFFH)

★ **Figure 10-14. TMD0 Compare Operation Example (2/2)**



Remark Interval time = (FFFFH + 2) × (Count clock cycle)

10.2.7 Application examples (timer D)

(1) Interval timer

This section explains an example in which timer D is used as an interval timer with 16-bit precision.

Interrupt requests (INTCMDn) are output at equal intervals (see **Figure 10-14 TMD0 Compare Operation Example**). The setup procedure is shown below (n = 0 to 3).

<1> Set (1) the TMDCAEn bit.
<2> Set each register.
- Select the count clock using the CSn0 to CSn2 bits of the TMCDn register.
- Set the compare value in the CMDn register.

<3> Start counting by setting (1) the TMDCEn bit.
<4> If the TMDn register and CMDn register values match, an INTCMDn interrupt is generated.
<5> INTCMDn interrupts are generated thereafter at equal intervals.

Remark n = 0 to 3

10.2.8 Cautions (timer D)

Various cautions concerning timer D are shown below.

(1) To operate TMDn, first set (1) the TMDCAEn bit.

(2) Up to 4 clocks are required after a value is set in the TMDCEn bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent cycles.

(3) To initialize the TMDn register status and start counting again, clear (0) the TMDCEn bit and then set (1) the TMDCEn bit after an interval of 4 clocks has elapsed.

(4) Up to 4 clocks are required until the value that was set in the CMDn register is transferred to internal units. When writing continuously to the CMDn register, be sure to secure a time interval of at least 4 clocks.

(5) The CMDn register can be overwritten only once during a timer/counter operation (from 0000H until an INTCMDn interrupt is generated due to a match of the TMDn register and CMDn register). If this cannot be secured, make sure that the CMDn register is not overwritten during a timer/counter operation.

(6) The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the TMDCEn bit is cleared (0). If the count clock is overwritten during a timer operation, operation cannot be guaranteed.

(7) A match signal will be generated after an overflow if a value less than the counter value is written in the CMDn register during TMDn register operation.

Remark n = 0 to 3

CHAPTER 11 SERIAL INTERFACE FUNCTION

11.1 Features

The serial interface function provides two types of serial interfaces equipped with six transmit/receive channels of which four channels can be used simultaneously.

The following two interface formats are available.

(1) Asynchronous serial interface (UART0 to UART2): 3 channels
(2) Clocked serial interface (CSI0 to CSI2): 3 channels

UART0 to UART2, which use the method of transmitting/receiving one byte of serial data following a start bit, enable full-duplex communication to be performed.

CSI0 to CSI2 transfer data according to three types of signals (3-wire serial I/O). These signals are the serial clock ($\overline{SCK0}$ to $\overline{SCK2}$), serial input (SI0 to SI2), and serial output (SO0 to SO2) signals.

11.1.1 Switching between UART and CSI modes

In the V850E/MA1, since UART0 and CSI0 pin and the UART1 and CSI1 pin are alternate function pins, they cannot be used at the same time. The PMC4 and PFC4 registers must be set in advance (see **14.3.5 Port 4**).

Also, since UART2 and CSI2 have alternate functions as external interrupt request input pins ($\overline{INTP120}$ and $\overline{INTP130}$ to $\overline{INTP133}$), the PMC3 and PFC3 registers must be set in advance (see **14.3.4 Port 3**).

If the mode is switched during a transmit or receive operation in UARTn or CSIn, operation cannot be guaranteed.

11.2 Asynchronous Serial Interfaces 0 to 2 (UART0 to UART2)

11.2.1 Features

- Transfer rate: 300 bps to 1,562.5 Kbps (using a dedicated baud rate generator and an internal system clock of 50 MHz)
- Full-duplex communications
  On-chip receive buffer (RXBn)
  On-chip transmit buffer (TXBn)
- Two-pin configuration
  TXDn: Transmit data output pin
  RXDn: Receive data input pin
- Reception error detection function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3 types
  - Reception error interrupt (INTSERn): Interrupt is generated according to the logical OR of the three types of reception errors
  - Reception completion interrupt (INTSRn): Interrupt is generated when receive data is transferred from the shift register to the receive buffer after serial transfer is completed during a reception enabled state
  - Transmission completion interrupt (INTSTn): Interrupt is generated when the serial transmission of transmit data (8 or 7 bits) from the shift register is completed
- The character length of transmit/receive data is specified according to the ASIM0 to ASIM2 registers
- Character length: 7 or 8 bits
- Parity functions: Odd, even, 0, or none
- Transmission stop bits: 1 or 2 bits
- On-chip dedicated baud rate generator

Remark n = 0 to 2

11.2.2 Configuration

UARTn is controlled by the asynchronous serial interface mode register (ASIMn), asynchronous serial interface status register (ASISn), and asynchronous serial interface transmission status register (ASIFn) (n = 0 to 2). Receive data is held in the receive buffer (RXBn), and transmit data is written to the transmit buffer (TXBn).

Figure 11-1 shows the configuration of the asynchronous serial interface.

(1) Asynchronous serial interface mode registers 0 to 2 (ASIM0 to ASIM2)

The ASIMn register is an 8-bit register for specifying the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status registers 0 to 2 (ASIS0 to ASIS2)

The ASISn register consists of a set of flags that indicate the error contents when a reception error occurs. The various reception error flags are set (1) when a reception error occurs and are reset (0) when the ASISn register is read.

(3) Asynchronous serial interface transmission status registers 0 to 2 (ASIF0 to ASIF2)

The ASIFn register is an 8-bit register that indicates the status when a transmit operation is performed.

This register consists of a transmit buffer data flag, which indicates the hold status of TXBn data, and the transmit shift register data flag, which indicates whether transmission is in progress.

(4) Reception control parity check

Receive operations are controlled according to the contents set in the ASIMn register. A check for parity errors is also performed during a receive operation, and if an error is detected, the value corresponding to the error contents is set in the ASISn register.

(5) Receive shift register

This is a shift register that converts the serial data that was input to the RXDn pin to parallel data. One byte of data is received, and if a stop bit is detected, the receive data is transferred to the receive buffer.

This register cannot be directly manipulated.

(6) Receive buffer (RXBn)

RXBn is an 8-bit buffer register for holding receive data. When 7 characters are received, 0 is stored in the MSB.

During a reception enabled state, receive data is transferred from the receive shift register to the receive buffer, in synchronization with the end of the shift-in processing of one frame.

Also, the reception completion interrupt request (INTSRn) is generated by the transfer of data to the receive buffer.

(7) Transmit shift register

This is a shift register that converts the parallel data that was transferred from the transmit buffer to serial data.

When one byte of data is transferred from the transmit buffer, the shift register data is output from the TXDn pin.

This register cannot be directly manipulated.

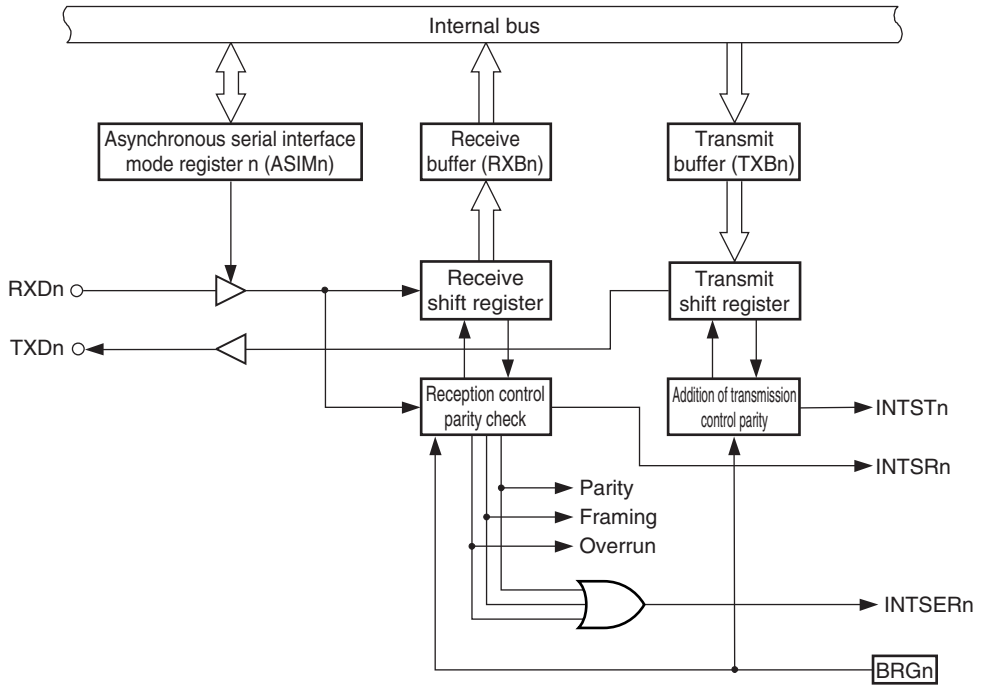
(8) Transmit buffer (TXBn)

TXBn is an 8-bit buffer for transmit data. A transmit operation is started by writing transmit data to TXBn. The transmission completion interrupt request (INTSTn) is generated in synchronization with the completion of transmission of one frame.

(9) Addition of transmission control parity

Transmit operations are controlled by adding a start bit, parity bit, or stop bit to the data that is written to the TXBn register, according to the contents that were set in the ASIMn register.

Figure 11-1. Asynchronous Serial Interface Block Diagram



Remark n = 0 to 2

11.2.3 Control registers

(1) Asynchronous serial interface mode registers 0 to 2 (ASIM0 to ASIM2)

These are 8-bit registers for controlling the transfer operations of UART0 to UART2.

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

★ **Caution To use UARTn, set clock select register n (CKSRn) and baud rate generator control register n (BRGCn). Then set the UARTCAEn bit to 1 before setting the other bits.**

(1/3)

	<7>	<6>	<5>	4	3	2	1	0	Address	After reset
ASIM0	UARTCAE0	TXE0	RXE0	PS01	PS00	CL0	SL0	ISRM0	FFFFFA00H	01H
ASIM1	UARTCAE1	TXE1	RXE1	PS11	PS10	CL1	SL1	ISRM1	FFFFFA10H	01H
ASIM2	UARTCAE2	TXE2	RXE2	PS21	PS20	CL2	SL2	ISRM2	FFFFFA20H	01H

Bit position	Bit name	Function
7	UARTCAEn (n = 0 to 2)	Clock Enable Controls the operation clock (n = 0 to 2). 0: Stops supply of clocks to UARTn unit 1: Supplies clocks to UARTn unit **Cautions 1. When the UARTCAEn bit is set to 0, the UARTn unit can be asynchronously reset.** **2. When UARTCAEn = 0, the UARTn unit is in a reset state. Therefore, to operate UARTn, the UARTCAEn bit must be set to 1.** **3. When the UARTCAEn bit is changed from 1 to 0, all registers of the UARTn unit are initialized. When the UARTCAEn is set to 1 again, the UARTn unit registers must be set again.** The TXDn pin output is always high level in the transmission disable state, irrespective of the setting of the UARTCAEn bit.
6	TXEn (n = 0 to 2)	Transmit Enable Specifies whether transmission is enabled or disabled. 0: Transmission is disabled 1: Transmission is enabled **Cautions 1. On startup, set UARTCAEn to 1 and then set TXEn to 1. Also, always set UARTCAEn to 0 after setting TXEn to 0.** **2. When the transmission unit status is to be initialized, the transmission status may not be able to be initialized unless the TXEn bit is set (1) again after an interval of two cycles of the basic clock has elapsed since the TXEn bit was cleared (0) (For the basic clock, see 11.2.6 (1) (a) Basic clock (Clock)).**

(2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>RXEn
(n = 0 to 2)</td><td>Receive Enable
Specifies whether reception is enabled or disabled.
0: Reception is disabled
1: Reception is enabled

Cautions 1. On startup, set UARTCAEn to 1 and then set RXEn to 1. Also, always set UARTCAEn to 0 after setting RXEn to 0.
2. When the reception unit status is to be initialized, the reception status may not be able to be initialized unless the RXEn bit is set (1) again after an interval of two cycles of the basic clock has elapsed since the RXEn bit was cleared (0) (For the basic clock, see 11.2.6 (1) (a) Basic clock (Clock)) .</td></tr>
<tr><td>4, 3</td><td>PSn1, PSn0
(n = 0 to 2)</td><td>Parity Select
Controls the parity bit.
<table><tr><th>PSn1</th><th>PSn0</th><th>Transmit operation</th><th>Receive operation</th></tr><tr><td>0</td><td>0</td><td>Do not output a parity bit</td><td>Receive with no parity</td></tr><tr><td>0</td><td>1</td><td>Output 0 parity</td><td>Receive as 0 parity</td></tr><tr><td>1</td><td>0</td><td>Output odd parity</td><td>Judge as odd parity</td></tr><tr><td>1</td><td>1</td><td>Output even parity</td><td>Judge as even parity</td></tr></table>
Cautions 1. To overwrite the PSn1 and PSn0 bits, first clear (0) the TXEn and RXEn bits.
2. If “0 parity” is selected for reception, no parity judgement is made. Therefore, no error interrupt is generated because the PEn bit of the ASISn register is not set.

• Even parity
If the transmit data contains an odd number of bits with the value “1”, the parity bit is set (1). If it contains an even number of bits with the value “1”, the parity bit is cleared (0). This controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an even number.
During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is odd, a parity error is generated.
• Odd parity
In contrast to even parity, odd parity controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an odd number.
During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is even, a parity error is generated.
• 0 parity
During transmission, the parity bit is cleared (0) regardless of the transmit data.
During reception, no parity error is generated because no parity bit is checked.
• No parity
No parity bit is added to transmit data.
During reception, the receive data is considered to have no parity bit. No parity error is generated because there is no parity bit.</td></tr>
</table>

(3/3)

Bit position	Bit name	Function
2	CLn (n = 0 to 2)	Character Length Specifies the character length of the transmit/receive data. 0: 7 bits 1: 8 bits **Caution To overwrite the CLn bit, first clear (0) the TXEn and RXEn bits.**
1	SLn (n = 0 to 2)	Stop Bit Length Specifies the stop bit length of the transmit data. 0: 1 bit 1: 2 bits **Cautions 1. To overwrite the SLn bit, first clear (0) the TXEn bit.** **2. Since reception always operates by using a single stop bit, the SLn bit setting does not affect receive operations.**
0	ISRMn (n = 0 to 2)	Interrupt Serial Receive Mode Specifies whether the generation of reception completion interrupt requests when an error occurs is enable or disabled. 0: An error interrupt request (INTSERn) is generated when an error occurs. In this case, no reception completion interrupt request (INTSRn) is generated. 1: A reception completion interrupt request (INTSRn) is generated when an error occurs. In this case, no error interrupt request (INTSERn) is generated. **Caution To overwrite the ISRMn bit, first clear (0) the RXEn bit.**

Remark When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to the receive buffer is performed, and the contents of the receive buffer are retained.

When reception is enabled, the receive shift operation starts, in synchronization with the detection of the start bit, and when the reception of one frame is completed, the contents of the receive shift register are transferred to the receive buffer. A reception completion interrupt (INTSRn) is also generated, in synchronization with the transfer to the receive buffer.

(2) Asynchronous serial interface status registers 0 to 2 (ASIS0 to ASIS2)

These registers, which consist of 3-bit error flags (PEn, FEn, and OVEn), indicate the error status when UARTn reception is completed (n = 0 to 2).

The status flag, which indicates a reception error, always indicates the status of the error that occurred most recently. That is, if the same error occurred several times before the receive data was read, this flag would hold only the status of the error that occurred last.

The ASISn register is cleared to 00H by a read operation. When a reception error occurs, the receive buffer (RXBn) should be read after the ASISn register is read.

These registers are read-only in 8-bit units.

Caution When the UARTCAEn bit or RXEn bit of the ASIMn register is set to 0, or when the ASISn register is read, the PEn, FEn, and OVEn bits of the ASISn register are cleared (0).

	7	6	5	4	3	2	1	0	Address	After reset
ASIS0	0	0	0	0	0	PE0	FE0	OVE0	FFFFFA03H	00H
ASIS1	0	0	0	0	0	PE1	FE1	OVE1	FFFFFA13H	00H
ASIS2	0	0	0	0	0	PE2	FE2	OVE2	FFFFFA23H	00H

Bit position	Bit name	Function
2	PEn (n = 0 to 2)	Parity Error This is a status flag that indicates a parity error. 0: When reception was completed, the transmit data parity matched the parity bit 1: When reception was completed, the transmit data parity did not match the parity bit **Caution The operation of the PEn bit differs according to the settings of the PSn1 and PSn0 bits of the ASIMn register.**
1	FEn (n = 0 to 2)	Framing Error This is a status flag that indicates a framing error. 0: When reception was completed, a stop bit was detected 1: When reception was completed, no stop bit was detected **Caution For receive data stop bits, only the first bit is checked regardless of the number of stop bits.**
0	OVEn (n = 0 to 2)	Overrun Error This is a status flag that indicates an overrun error. 0: The receive operation of one receive data value was completed or the receive operation of the next receive data value was in progress. 1: UARTn completed the next receive operation before reading the RXBn receive data. **Caution When an overrun error occurs, the next receive data value is not written to the RXBn register and the data is discarded.**

(3) Asynchronous serial interface transmission status registers 0 to 2 (ASIF0 to ASIF2)

These registers, which consist of 2-bit status flags, indicate the status during transmission.

By writing the next data to the TXBn register after data is transferred from the TXBn register to the TXSn register, transmit operations can be performed continuously without suspension even during an interrupt interval. When transmission is performed continuously, data should be written after referencing the ASIFn register to prevent writing to the TXBn register by mistake.

These registers are read-only in 8-bit or 1-bit units.

Remark n = 0 to 2

	7	6	5	4	3	2	<1>	<0>	Address	After reset
ASIF0	0	0	0	0	0	0	TXBF0	TXSF0	FFFFFA05H	00H
ASIF1	0	0	0	0	0	0	TXBF1	TXSF1	FFFFFA15H	00H
ASIF2	0	0	0	0	0	0	TXBF2	TXSF2	FFFFFA25H	00H

Bit position	Bit name	Function
1	TXBFn (n = 0 to 2)	Transmit Buffer Flag This is a transmit buffer data flag. 0: No data exists in TXBn 1: Data exists in TXBn **Caution When UARTCAEn = 0 and TXEn = 0 in the ASIMn register, the TXBFn bit is cleared (0).**
0	TXSFn (n = 0 to 2)	Transmit Shift Flag This is a transmit shift register data flag. It indicates the transmission status of UARTn. 0: Transmission completed or awaiting transmission 1: Transmission in progress **Caution When UARTCAEn = 0 and TXEn = 0 in the ASIMn register, the TXSFn bit is cleared (0).**

The following table shows relationship between the transmission status and write operations to TXBn.

TXBFn	TXSFn	Transmission Status	Write Operation to TXBn
0	0	Initial status or transmission completed	Writing is permitted
0	1	Transmission in progress (no data is in TXBn)	Writing is permitted
1	0	Awaiting transmission (data is in TXBn)	Writing is not permitted
1	1	Transmission in progress (data is in TXBn)	Writing is not permitted

Caution When transmission is performed continuously, data should be written to TXBn after confirming the TXBFn value. If writing is not enabled, operation cannot be guaranteed when data is written to TXBn.

(4) Receive buffer registers 0 to 2 (RXB0 to RXB2)

These are 8-bit buffer registers for storing parallel data that had been converted by the receive shift register.

When reception is enabled (RXEn = 1 in the ASIMn register), receive data is transferred from the receive shift register to the receive buffer, in synchronization with the completion of the shift-in processing of one frame. Also, a reception completion interrupt request (INTSRn) is generated by the transfer to the receive buffer. For information about the timing for generating these interrupt requests, see **11.2.5 (4) Receive operation**.

If reception is disabled (RXEn = 0 in the ASIMn register), the contents of the receive buffer are retained, and no processing is performed for transferring data to the receive buffer even when the shift-in processing of one frame is completed. Also, no reception completion interrupt is generated.

When 7 bits is specified for the data length, bits 6 to 0 of the RXBn register are transferred for the receive data and the MSB (bit 7) is always 0. However, if an overrun error occurs, the receive data at that time is not transferred to the RXBn register.

Except when a reset is input, the RXBn register becomes FFH even when UARTCAEn = 0 in the ASIMn register.

These registers are read-only in 8-bit units.

Remark n = 0 to 2

	7	6	5	4	3	2	1	0	Address	After reset
RXB0	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00	FFFFFA02H	FFH
RXB1	RXB17	RXB16	RXB15	RXB14	RXB13	RXB12	RXB11	RXB10	FFFFFA12H	FFH
RXB2	RXB27	RXB26	RXB25	RXB24	RXB23	RXB22	RXB21	RXB20	FFFFFA22H	FFH

Bit position	Bit name	Function
7 to 0	RXBn7 to RXBn0 (n = 0 to 2)	Receive Buffer Stores receive data. 0 can be read for RXBn7 when 7-bit character data is received.

(5) Transmit buffer registers 0 to 2 (TXB0 to TXB2)

These are 8-bit buffer registers for setting transmit data.

When transmission is enabled (TXEn = 1 in the ASIMn register), the transmit operation is started by writing data to TXBn.

When transmission is disabled (TXEn = 0 in the ASIMn register), even if data is written to TXBn, the value is ignored.

The TXBn data is transferred to the transmit shift register, and a transmission completion interrupt request (INTSTn) is generated, in synchronization with the completion of the transmission of one frame from the transmit shift register. For information about the timing for generating these interrupt requests, see **11.2.5 (2) Transmit operation**.

When TXBFn = 1 in the ASIFn register, writing must not be performed to TXBn.

These registers can be read or written in 8-bit units.

Remark n = 0 to 2

	7	6	5	4	3	2	1	0	Address	After reset
TXB0	TXB07	TXB06	TXB05	TXB04	TXB03	TXB02	TXB01	TXB00	FFFFFA04H	FFH
TXB1	TXB17	TXB16	TXB15	TXB14	TXB13	TXB12	TXB11	TXB10	FFFFFA14H	FFH
TXB2	TXB27	TXB26	TXB25	TXB24	TXB23	TXB22	TXB21	TXB20	FFFFFA24H	FFH

Bit position	Bit name	Function
7 to 0	TXBn7 to TXBn0 (n = 0 to 2)	Transmit Buffer Writes transmit data.

11.2.4 Interrupt requests

The following three types of interrupt requests are generated from UARTn (n = 0 to 2).

- Reception error interrupt (INTSERn)
- Reception completion interrupt (INTSRn)
- Transmission completion interrupt (INTSTn)

The default priorities among these three types of interrupt requests is, from high to low, reception error interrupt, reception completion interrupt, and transmission completion interrupt.

Table 11-1. Generated Interrupts and Default Priorities

Interrupt	Priority
Reception error	1
Reception completion	2
Transmission completion	3

(1) Reception error interrupt (INTSERn)

When reception is enabled, a reception error interrupt is generated according to the logical OR of the three types of reception errors explained for the ASISn register. Whether a reception error interrupt (INTSERn) or a reception completion interrupt (INTSRn) is generated when an error occurs can be specified using the ISRMn bit of the ASIMn register.

When reception is disabled, no reception error interrupt is generated.

(2) Reception completion interrupt (INTSRn)

When reception is enabled, a reception completion interrupt is generated when data is shifted in to the receive shift register and transferred to the receive buffer.

A reception completion interrupt request can be generated in place of a reception error interrupt according to the ISRMn bit of the ASIMn register even when a reception error has occurred.

When reception is disabled, no reception completion interrupt is generated.

(3) Transmission completion interrupt (INTSTn)

A transmission completion interrupt is generated when one frame of transmit data containing 7-bit or 8-bit characters is shifted out from the transmit shift register.

11.2.5 Operation

(1) Data format

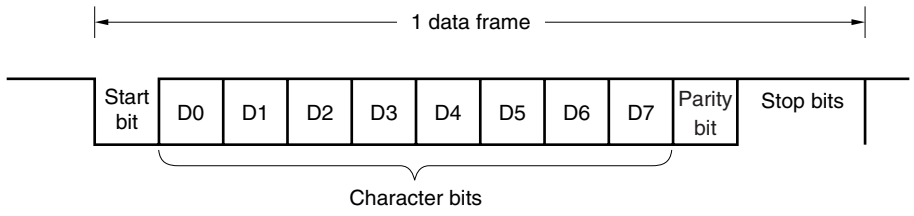
Full-duplex serial data transmission and reception can be performed.

The transmit/receive data format consists of one data frame containing a start bit, character bits, a parity bit, and stop bits as shown in Figure 11-2.

The character bit length within one data frame, the type of parity, and the stop bit length are specified by the asynchronous serial interface mode register n (ASIMn) (n = 0 to 2).

Also, data is transferred with the least significant bit (LSB) first.

Figure 11-2. Asynchronous Serial Interface Transmit/Receive Data Format



- Start bit ··· 1 bit
- Character bits ··· 7 bits or 8 bits
- Parity bit ··· Even parity, odd parity, 0 parity, or no parity
- Stop bits ··· 1 bit or 2 bits

(2) Transmit operation

When UARTCAEn is set to 1 in the ASIMn register, a high level is output to the TXDn pin.

Then, when TXEn is set to 1 in the ASIMn register, transmission is enabled, and the transmit operation is started by writing transmit data to transmit buffer register n (TXBn) (n = 0 to 2).

(a) Transmission enabled state

This state is set by the TXEn bit in the ASIMn register (n = 0 to 2).

- TXEn = 1: Transmission enabled state
- TXEn = 0: Transmission disabled state

However, when the transmission enabled state is set, to use UART0 and UART1, which share pins with clocked serial interfaces 0 and 1 (CSI0 and CSI1), the CSICAEn bit of clocked serial interface mode registers 0 and 1 (CSIM0 and CSIM1) should be set to 0.

Since UARTn does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in a reception enabled state.

(b) Starting a transmit operation

In the transmission enabled state, a transmit operation is started by writing transmit data to transmit buffer register n (TXBn). When a transmit operation is started, the data in TXBn is transferred to transmit shift register n (TXSn). Then, the TXSn register outputs data to the TXDn pin sequentially beginning with the LSB (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically (n = 0 to 2).

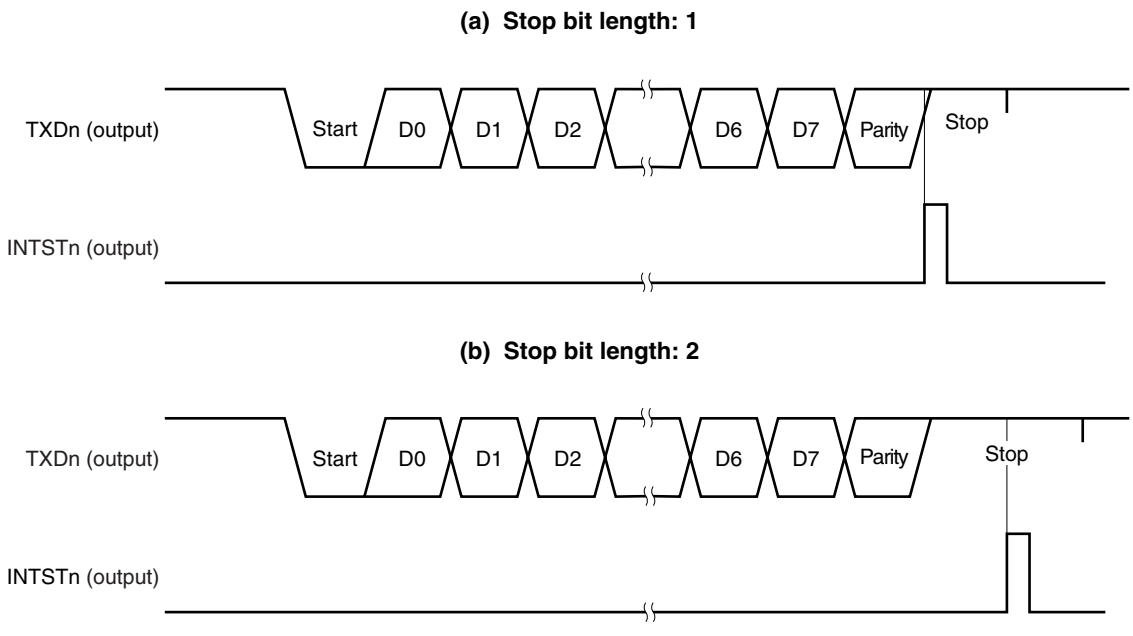
(c) Transmission interrupt request

When the transmit shift register (TXSn) becomes empty, a transmission completion interrupt request (INTSTn) is generated. The timing for generating the INTSTn interrupt differs according to the specification of the number of stop bits. The INTSTn interrupt is generated at the same time that the last stop bit is output (n = 0 to 2).

If the data to be transmitted next has not been written to the TXBn register, the transmit operation is suspended.

Caution Normally, when transmit shift register n (TXSn) becomes empty, a transmission completion interrupt (INTSTn) is generated. However, no transmission completion interrupt (INTSTn) is generated if transmit shift register n (TXSn) becomes empty due to the input of a $\overline{\text{RESET}}$.

Figure 11-3. Asynchronous Serial Interface Transmission Completion Interrupt Timing



Remark n = 0 to 2

(3) Continuous transmission operation

UARTn can write the next data to the TXBn register at the time that the TXSn register starts the shift operation. This enables an efficient transmission rate to be realized by continuously transmitting data even during interrupt servicing after the transmission of one data frame (n = 0 to 2).

When continuous transmission is performed, data should be written after referencing the ASIFn register to confirm the transmission status and whether or not data can be written to the TXBn register (n = 0 to 2).

Caution Transmit data should be written when the TXBFn bit is 0. The transmission unit should be initialized when the TXSFn bit is 0. If these actions are performed at other times, the transmit data cannot be guaranteed.

Table 11-2. Transmission Status and Whether or Not Writing Is Enabled

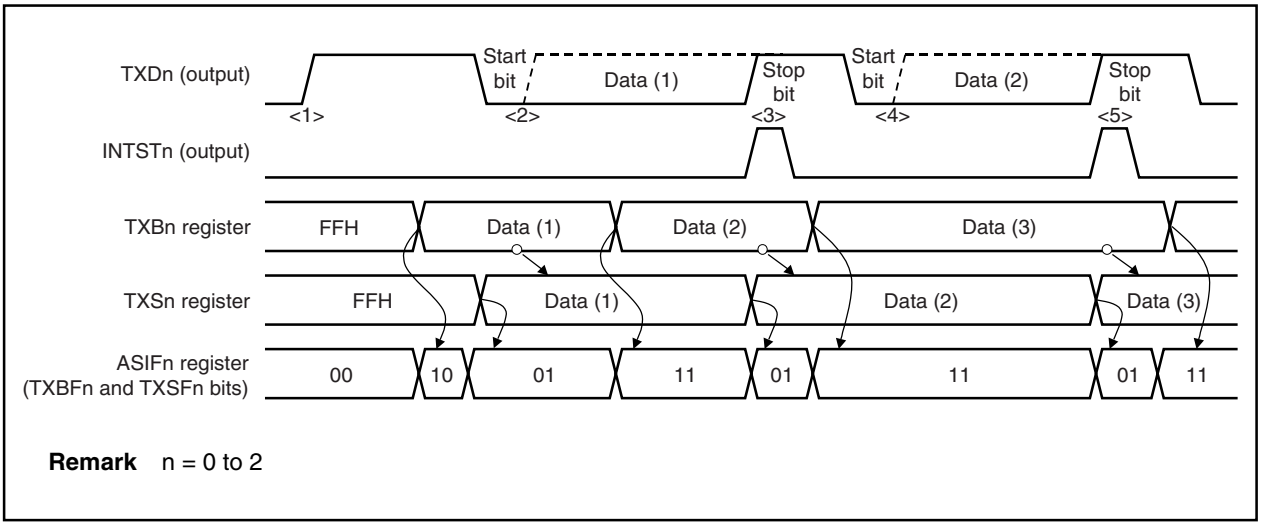
TXBFn	TXSFn	Transmission Status	Whether or Not Write Operation to TXBn Is Enabled
0	0	Initial status or transmission completed	Writing is enabled
0	1	Transmission in progress (no data is in TXBn register)	Writing is enabled
1	0	Awaiting transmission (data is in TXBn register)	Writing is not enabled
1	1	Transmission in progress (data is in TXBn register)	Writing is not enabled

Remark n = 0 to 2

(a) Starting procedure

The procedure for starting continuous transmission is shown below.

Figure 11-4. Continuous Transmission Starting Procedure



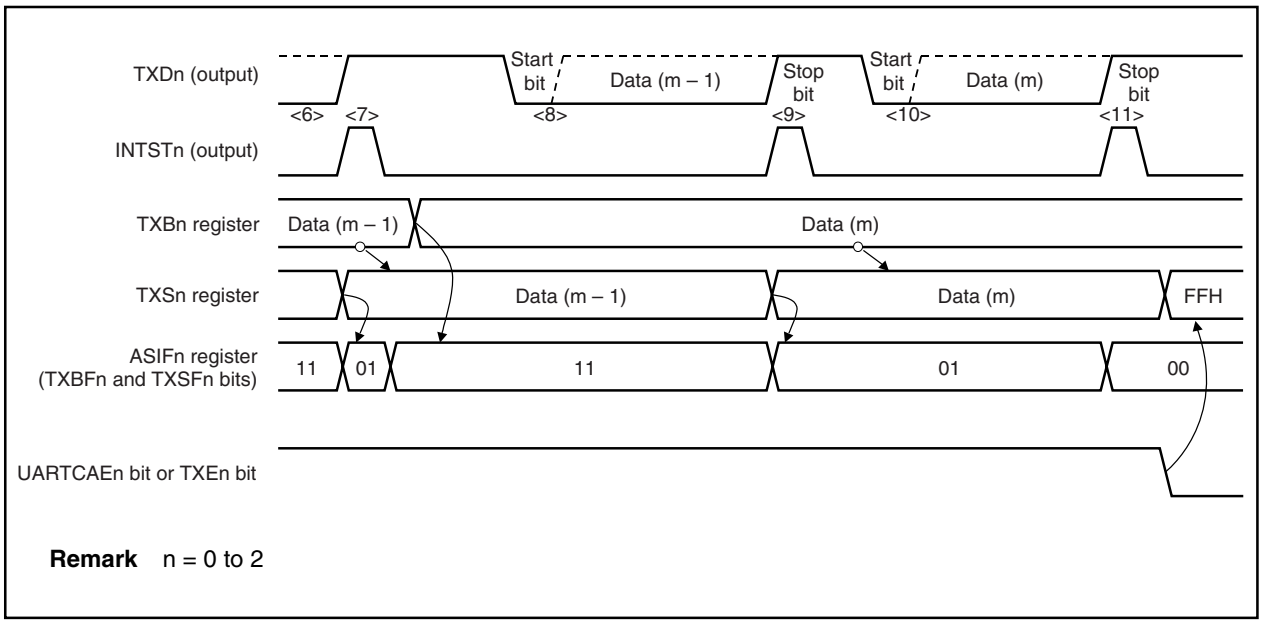
Remark n = 0 to 2

Transmission starting procedure	Internal operation	ASIFn register	
		TXBFn	TXSFn
• Set transmission mode	<1> Start transmission unit	0	0
• Write data (1) →		1	0
	<2> Generate start bit →	0	1
	Start data (1) transmission		
• Read ASIFn register (confirm that TXBFn bit = 0)			
• Write data (2) →		1	1
	<<Transmission in progress>>		
	<3> Generate INTSTn interrupt →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0)			
• Write data (3) →		1	1
	<4> Generate start bit		
	Start data (2) transmission		
	<<Transmission in progress>>		
	<5> Generate INTSTn interrupt →	0	1
• Read ASIFn register (confirm that TXBFn bit = 0)			
• Write data (4) →		1	1

(b) Ending procedure

The procedure for ending continuous transmission is shown below.

Figure 11-5. Continuous Transmission Ending Procedure



Remark n = 0 to 2

Transmission ending procedure	Internal operation	ASIFn register	
		TXBFn	TXSFn
	<6> Transmission of data (m – 2) is in progress	1	1
	<7> Generate INTST interrupt →	0	1
• Read ASIFn register (confirm that the TXBFn bit = 0)			
• Write data (n) →		1	1
	<8> Generate start bit		
	Start data (m – 1) transmission		
	<<Transmission in progress>>		
	<9> Generate INTSTn interrupt →	0	1
• Read ASIFn register (confirm that the TXSFn bit = 1)			
There is no write data			
	<10> Generate start bit		
	Start data (m) transmission		
	<<Transmission in progress>>		
	<11> Generate INTSTn interrupt →	0	0
• Read ASIFn register (confirm that the TXSFn bit = 0)			
• Clear (0) the UARTCAEn bit or TXEn bit	Initialize internal circuits		

(4) Receive operation

The awaiting reception state is set by setting UARTCAEn to 1 in the ASIMn register and then setting RXEn to 1 in the ASIMn register. RXDn pin sampling begins and a start bit is detected. When the start bit is detected, the receive operation begins, and data is stored sequentially in the receive shift register according to the baud rate that was set. A reception completion interrupt (INTSRn) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from the receive buffer (RXBn) to memory by this interrupt servicing (n = 0 to 2).

(a) Reception enabled state

The receive operation is set to the reception enabled state by setting the RXEn bit in the ASIMn register to 1 (n = 0 to 2).

- RXEn = 1: Reception enabled state
- RXEn = 0: Reception disabled state

However, when the reception enabled state is set, to use UART0 and UART1, which share pins with clocked serial interfaces 0 and 1 (CSI0 and CSI1), the operation of CSIn must be disabled by setting the CSICAEn bit of clocked serial interface mode registers 0 and 1 (CSIM0 and CSIM1) to 0 (n = 0 to 2).

In the reception disabled state, the reception hardware stands by in the initial state. At this time, the contents of the receive buffer are retained, and no reception completion interrupt or reception error interrupt is generated.

(b) Starting a receive operation

A receive operation is started by the detection of a start bit.

The RXDn pin is sampled using the serial clock from the baud rate generator (BRGn) (n = 0 to 2).

(c) Reception completion interrupt

When RXEn = 1 in the ASIMn register and the reception of one frame of data is completed (the stop bit is detected), a reception completion interrupt (INTSRn) is generated and the receive data within the receive shift register is transferred to RXBn at the same time (n = 0 to 2).

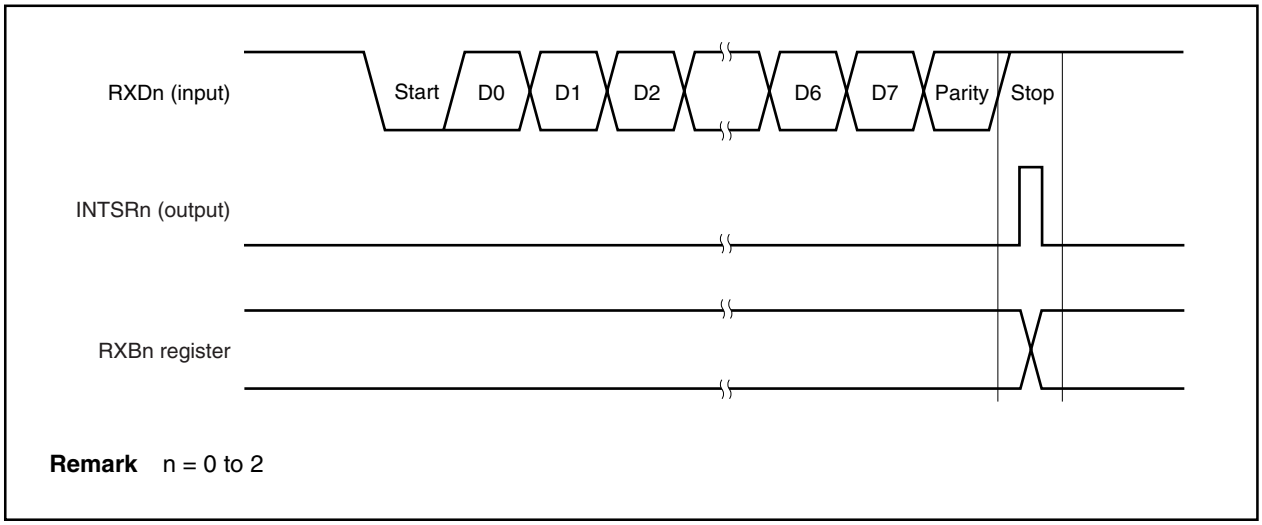
Also, if an overrun error occurs, the receive data at that time is not transferred to the receive buffer (RXBn), and either a reception completion interrupt (INTSRn) or a reception error interrupt (INTSERn) is generated (the receive data within the receive shift register is transferred to RXBn) according to the ISRMn bit setting in the ASIMn register.

Even if a parity error or framing error occurs during a reception operation, the receive operation continues, and after reception is completed, either a reception completion interrupt (INTSRn) or a reception error interrupt (INTSERn) is generated according to the ISRMn bit setting in the ASIMn register.

If the RXEn bit is reset (0) during a receive operation, the receive operation is immediately stopped. The contents of the receive buffer (RXBn) and of the asynchronous serial interface status register (ASISn) at this time do not change, and no reception completion interrupt (INTSRn) or reception error interrupt (INTSERn) is generated.

No reception completion interrupt is generated when RXEn = 0 (reception is disabled).

Figure 11-6. Asynchronous Serial Interface Reception Completion Interrupt Timing



Remark n = 0 to 2

(5) Reception error

The three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. The data reception result is that the various flags of the ASISn register are set (1), and a reception error interrupt (INTSERn) or a reception completion interrupt (INTSRn) is generated at the same time. The ISRMn bit of the ASIMn register specifies whether INTSERn or INTSRn is generated.

The type of error that occurred during reception can be ascertained by reading the contents of the ASISn register during the INTSERn or INTSRn interrupt servicing.

The contents of the ASISn register are reset (0) by reading the ASISn register (if the next reception data contains an error, the corresponding error flag is set (1)).

Table 11-3. Reception Error Causes

Error Flag	Reception Error	Cause
PEn	Parity error	The parity specification during transmission did not match the parity of the reception data
FEn	Framing error	No stop bit was detected
OVEn	Overrun error	The reception of the next data was completed before data was read from the receive buffer

Remark n = 0 to 2

(a) Separation of reception error interrupt

A reception error interrupt can be separated from the INTSRn interrupt and generated as an INTSERn interrupt by clearing the ISRMn bit of the ASIMn register (n = 0 to 2) to 0.

Figure 11-7. When Reception Error Interrupt Is Separated from INTSRn Interrupt (ISRMn Bit = 0)

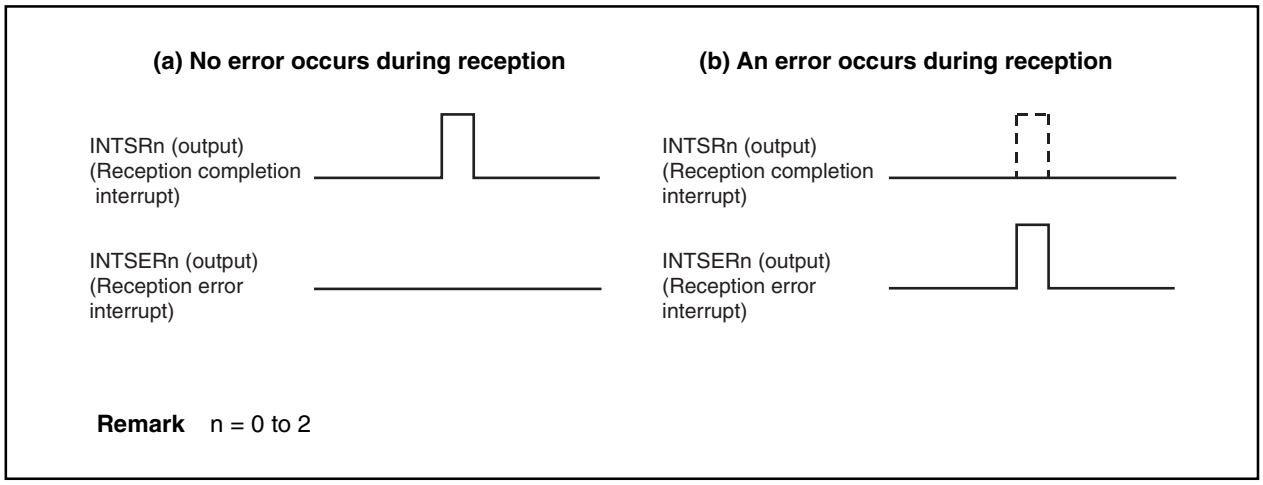
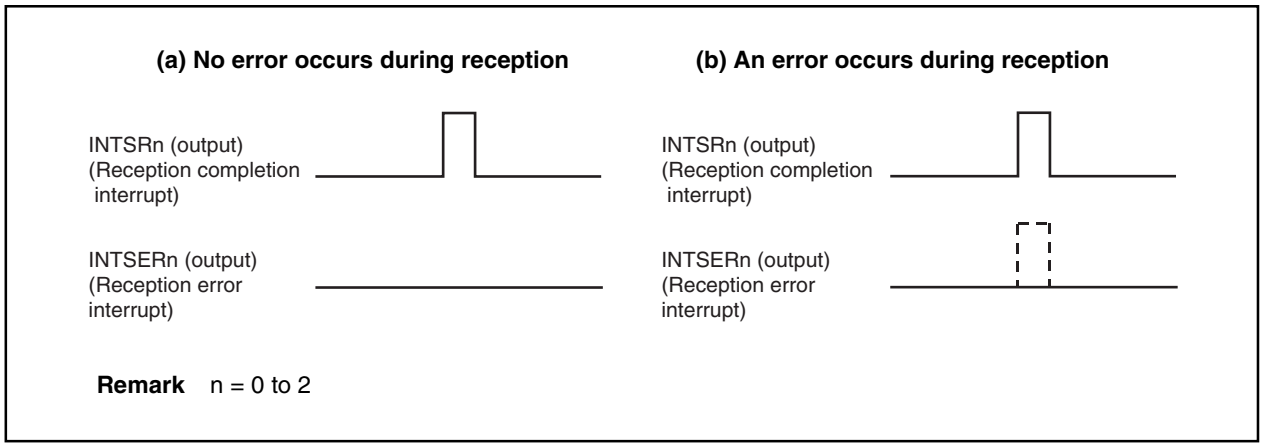


Figure 11-8. When Reception Error Interrupt Is Included in INTSRn Interrupt (ISRMn Bit = 1)



(6) Parity types and corresponding operation

A parity bit is used to detect a bit error in communication data. Normally, the same type of parity bit is used at the transmission and reception sides.

(a) Even parity

(i) During transmission

The parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is even. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 1
- If the number of bits with the value “1” within the transmit data is even: 0

(ii) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is odd.

(b) Odd parity

(i) During transmission

In contrast to even parity, the parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is odd. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 0
- If the number of bits with the value “1” within the transmit data is even: 1

(ii) During reception

The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is even.

(c) 0 parity

During transmission the parity bit is set to “0” regardless of the transmit data.

During reception, no parity bit check is performed. Therefore, no parity error is generated regardless of whether the parity bit is “0” or “1”.

(d) No parity

No parity bit is added to the transmit data.

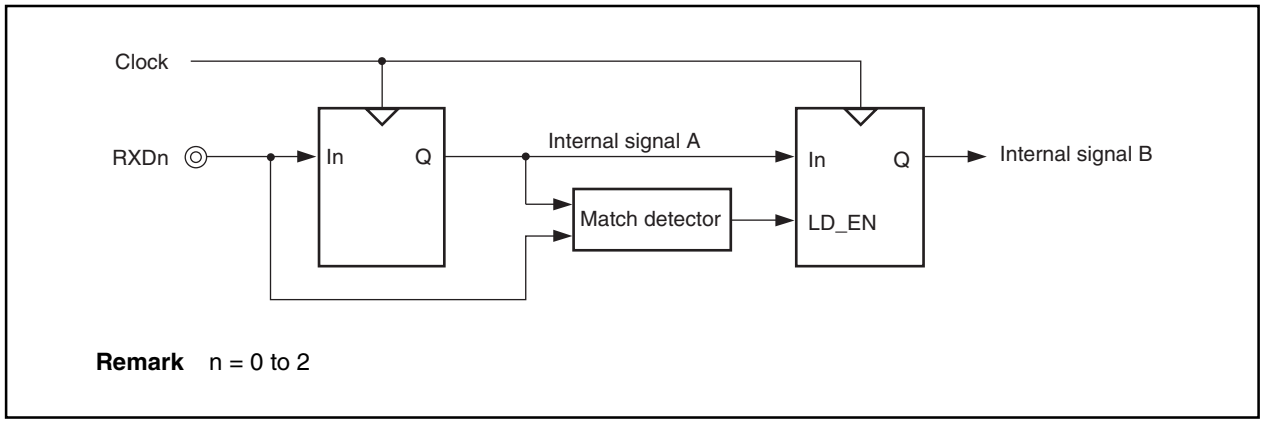
During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

(7) Receive data noise filter

The RXDn signal is sampled at the rising edge of the prescaler output clock. If the same sampling value is obtained twice, the match detector output changes, and this output is sampled as input data. Therefore, data not exceeding one clock width is judged to be noise and is not delivered to the internal circuit (see Figure 11-10). See **11.2.6 (1) (a) Basic clock (Clock)** regarding the basic clock.

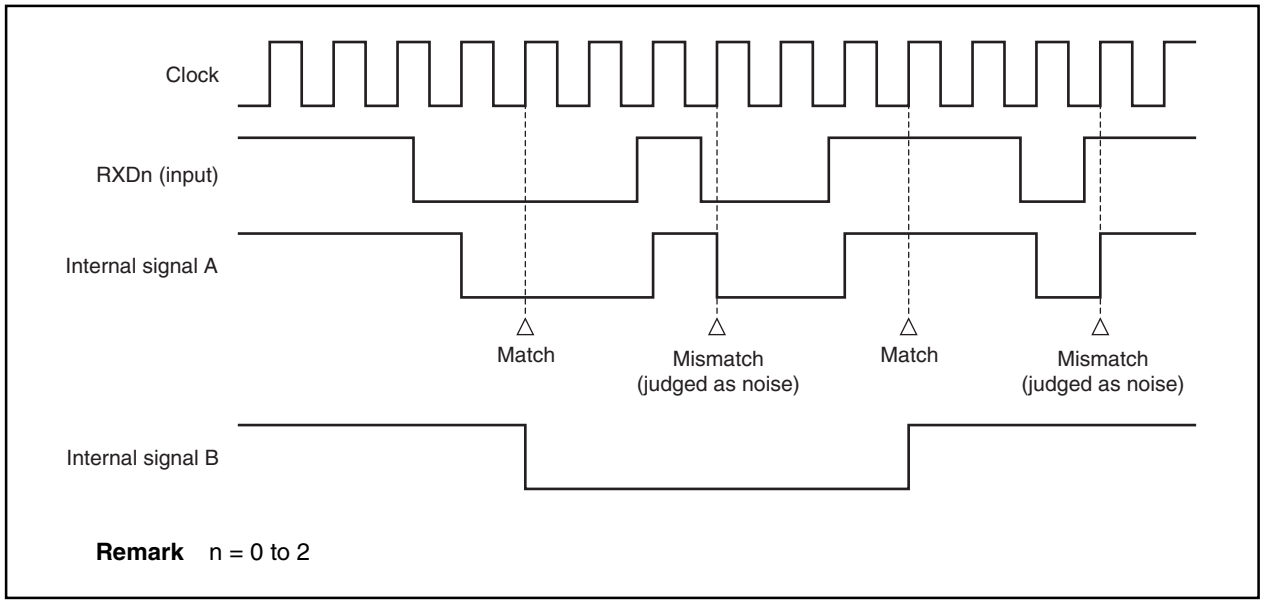
Also, since the circuit is configured as shown in Figure 11-9, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 11-9. Noise Filter Circuit



Remark n = 0 to 2

Figure 11-10. Timing of RXDn Signal Judged as Noise



Remark n = 0 to 2

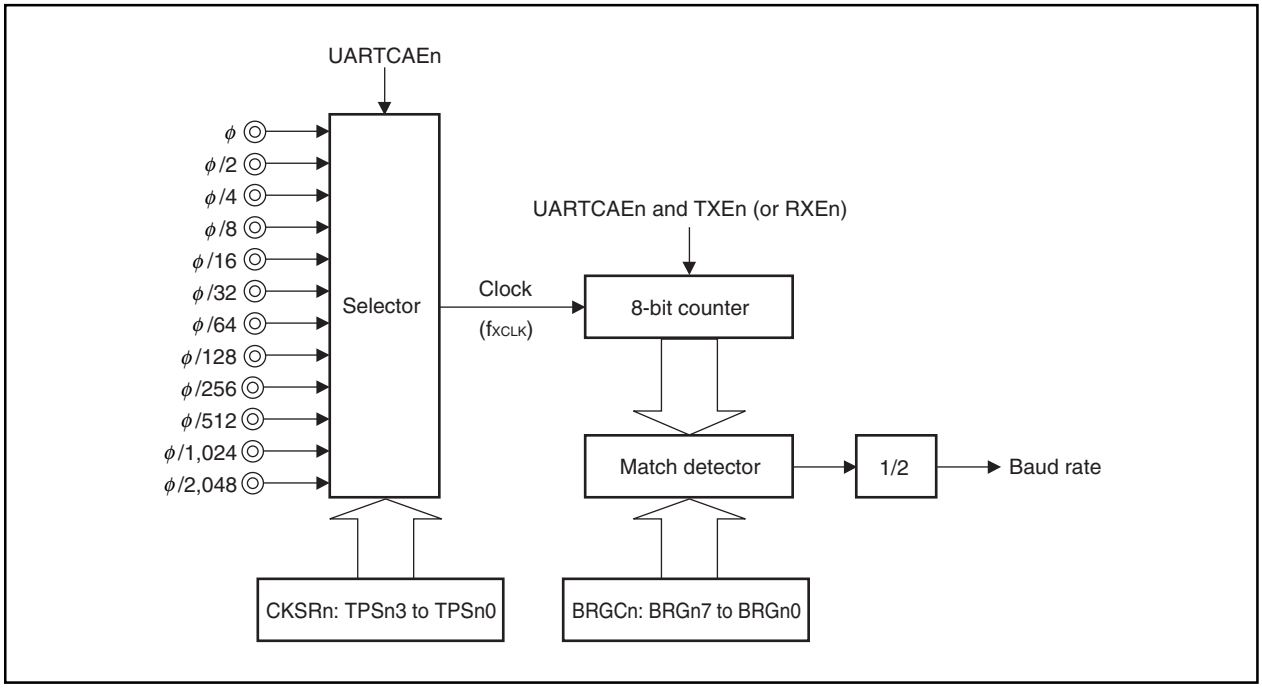
11.2.6 Dedicated baud rate generators 0 to 2 (BRG0 to BRG2)

A dedicated baud rate generator, which consists of a source clock selector and an 8-bit programmable counter, generates serial clocks during transmission/reception in UARTn. The dedicated baud rate generator output can be selected as the serial clock for each channel.

Separate 8-bit counters exist for transmission and for reception.

(1) Baud rate generator configuration

Figure 11-11. Baud Rate Generator Configuration



(a) Basic clock (Clock)

When UARTCAEn = 1 in the ASIMn register, the clock selected according to the TPSn3 to TPSn0 bits of the CKSRn register is supplied to the transmission/reception unit. This clock is called the basic clock, and its frequency is referred to as f_{XCLK}. When UARTCAEn = 0, the clock signal is fixed at low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the CKSRn and BRGCn registers (n = 0 to 2).

The clock input to the 8-bit counter is selected according to the TPSn3 to TPSn0 bits of the CKSRn register.

The 8-bit counter divisor value can be selected according to the BRGn7 to BRGn0 bits of the BRGCn register.

(a) Clock select registers 0 to 2 (CKSR0 to CKSR2)

The CKSRn register is an 8-bit register for selecting the input block according to the TPSn3 to TPSn0 bits. The clock selected by the TPSn3 to TPSn0 bits becomes the basic clock of the transmission/reception module. Its frequency is referred to as f_{XCLK}.

These registers can be read or written in 8-bit units.

Caution The maximum allowable frequency of the basic clock (f_{XCLK}) is 25 MHz. Therefore, when the system clock's frequency is 50 MHz, bits TPSn3 to TPSn0 cannot be set to 0000B (n = 0 to 2).

If the system clock frequency is 50 MHz, set the TPSn3 to TPSn0 bits to a value other than 0000B and set the UARTCAEn bit of the ASIMn register to 1.

	7	6	5	4	3	2	1	0	Address	After reset
CKSR0	0	0	0	0	TPS03	TPS02	TPS01	TPS00	FFFFFA06H	00H
CKSR1	0	0	0	0	TPS13	TPS12	TPS11	TPS10	FFFFFA16H	00H
CKSR2	0	0	0	0	TPS23	TPS22	TPS21	TPS20	FFFFFA26H	00H

Bit position	Bit name	Function
3 to 0	TPSn3 to TPSn0 (n = 0 to 2)	Specifies the basic clock

TPSn3	TPSn2	TPSn1	TPSn0	Basic clock (f_{XCLK})
0	0	0	0	ϕ
0	0	0	1	ϕ/2
0	0	1	0	ϕ/4
0	0	1	1	ϕ/8
0	1	0	0	ϕ/16
0	1	0	1	ϕ/32
0	1	1	0	ϕ/64
0	1	1	1	ϕ/128
1	0	0	0	ϕ/256
1	0	0	1	ϕ/512
1	0	1	0	ϕ/1,024
1	0	1	1	ϕ/2,048
1	1	Arbitrary	Arbitrary	Setting prohibited

(b) Baud rate generator control registers 0 to 2 (BRGC0 to BRGC2)

The BRGCn register is an 8-bit register that controls the baud rate (serial transfer speed) of UARTn.

These registers can be read or written in 8-bit units.

Caution If the BRGn7 to BRGn0 bits are to be overwritten, TXEn and RXEn should be set to 0 in the ASIMn register first (n = 0 to 2).

	7	6	5	4	3	2	1	0	Address	After reset
BRGC0	MDL07	MDL06	MDL05	MDL04	MDL03	MDL02	MDL01	MDL00	FFFFFA07H	FFH
BRGC1	MDL17	MDL16	MDL15	MDL14	MDL13	MDL12	MDL11	MDL10	FFFFFA17H	FFH
BRGC2	MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20	FFFFFA27H	FFH

Bit position	Bit name	Function
7 to 0	BRGn7 to BRGn0 (n = 0 to 2)	Specifies the 8-bit counter's divisor value.

BRGn7	BRGn6	BRGn5	BRGn4	BRGn3	BRGn2	BRGn1	BRGn0	Divisor value (k)	Serial clock
0	0	0	0	0	x	x	x	–	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{XCLK}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK}/10$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	0	1	0	250	$f_{XCLK}/250$
1	1	1	1	1	0	1	1	251	$f_{XCLK}/251$
1	1	1	1	1	1	0	0	252	$f_{XCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK}/255$

Remarks 1. f_{XCLK}: Frequency of clock selected according to TPSn3 to TPSn0 bits of CKSRn register.

2. k: Value set according to BRGn7 to BRGn0 bits (k = 8, 9, 10, ..., 255)

3. x: don't care

(c) Baud rate

The baud rate is the value obtained according to the following formula.

$$\text{Baud rate} = \frac{f_{XCLK}}{2 \times k} \text{ [bps]}$$

f_{XCLK} = Frequency of basic clock selected according to TPSn3 to TPSn0 bits of CKSRn register.
k = Value set according to BRGn7 to BRGn0 bits of BRGCn register (k = 8, 9, 10, ..., 255)

(d) Baud rate error

The baud rate error is obtained according to the following formula.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. Make sure that the baud rate error during transmission does not exceed the allowable error of the reception destination.

2. Make sure that the baud rate error during reception is within the allowable baud rate range during reception, which is described in paragraph (4).

Example: Basic clock frequency = 20 MHz = 20,000,000 Hz
Settings of BRGn7 to BRGn0 bits in BRGCn register = 01000001B (k = 65)
Target baud rate = 153,600 bps

Baud rate = 20 M/(2 × 65)
= 20,000,000/(2 × 65) = 153,846 [bps]

Error = (153,846/153,600 − 1) × 100
= 0.160 [%]

(3) Baud rate setting example

Table 11-4. Baud Rate Generator Setting Data

Baud Rate (bps)	ϕ = 50 MHz			ϕ = 40 MHz			ϕ = 33 MHz			ϕ = 10 MHz		
	f_{XCLK}	k	ERR	f_{XCLK}	k	ERR	f_{XCLK}	k	ERR	f_{XCLK}	k	ERR
300	$\phi/2^9$	163	0.15	$\phi/2^{10}$	65	0.16	$\phi/2^8$	215	–0.07	$\phi/2^7$	130	0.16
600	$\phi/2^8$	163	0.15	$\phi/2^9$	65	0.16	$\phi/2^7$	215	–0.07	$\phi/2^6$	130	0.16
1,200	$\phi/2^7$	163	0.15	$\phi/2^8$	65	0.16	$\phi/2^6$	215	–0.07	$\phi/2^5$	130	0.16
2,400	$\phi/2^6$	163	0.15	$\phi/2^7$	65	0.16	$\phi/2^5$	215	–0.07	$\phi/2^4$	130	0.16
4,800	$\phi/2^5$	163	0.15	$\phi/2^6$	65	0.16	$\phi/2^4$	215	–0.07	$\phi/2^3$	130	0.16
9,600	$\phi/2^4$	163	0.15	$\phi/2^5$	65	0.16	$\phi/2^3$	215	–0.07	$\phi/2^2$	130	0.16
19,200	$\phi/2^3$	163	0.15	$\phi/2^4$	80	0.16	$\phi/2^2$	215	–0.07	$\phi/2^1$	130	0.16
31,250	$\phi/2^3$	100	0	$\phi/2^3$	65	0	$\phi/2^2$	132	0	$\phi/2^1$	80	0
38,400	$\phi/2^2$	163	0.15	$\phi/2^3$	65	0.16	$\phi/2^1$	215	–0.07	$\phi/2^0$	130	0.16
76,800	$\phi/2^2$	81	0.47	$\phi/2^2$	65	0.16	$\phi/2^1$	107	0.39	$\phi/2^0$	65	0.16
153,600	$\phi/2^1$	81	0.47	$\phi/2^1$	65	0.16	$\phi/2^1$	54	–0.54	$\phi/2^0$	33	–1.36
312,500	$\phi/2^1$	40	0	$\phi/2^1$	32	0	$\phi/2^1$	26	1.54	$\phi/2^0$	16	0

Caution The maximum allowable frequency of the basic clock (f_{XCLK}) is 25 MHz.

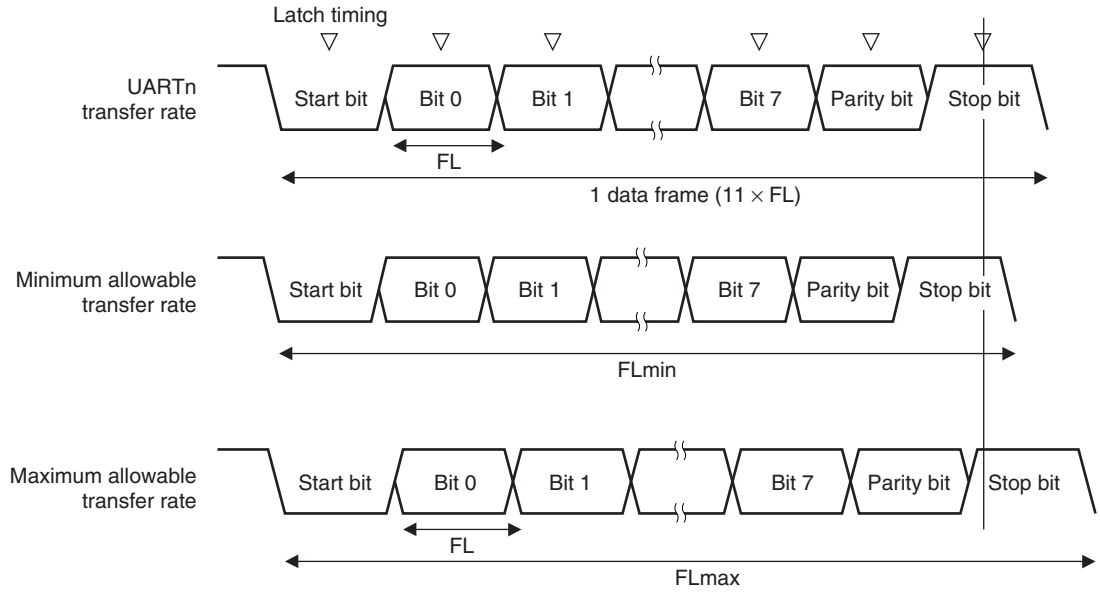
Remarks ϕ: System clock frequency
Clock: Input clock
k: Settings of BRGn7 to BRGn0 bits in BRGCn register (n = 0 to 2)
ERR: Baud rate error [%]

(4) Allowable baud rate range during reception

The degree to which a discrepancy from the transmission destination's baud rate is allowed during reception is shown below.

Caution The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

Figure 11-12. Allowable Baud Rate Range During Reception



Remark n = 0 to 2

As shown in Figure 11-12, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the BRGCn register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.

Applying this to 11-bit reception is, theoretically, as follows.

$FL = (Brate)^{-1}$

Brate: UARTn baud rate (n = 0 to 2)
k: BRGCn setting value (n = 0 to 2)
FL: 1-bit data length
Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received at the transfer destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

Similarly, the maximum allowable transfer rate can be obtained as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the minimum baud rate that can be received at the transfer destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{22k}{21k - 2} Brate$$

The allowable baud rate error of UARTn and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 11-5. Maximum and Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

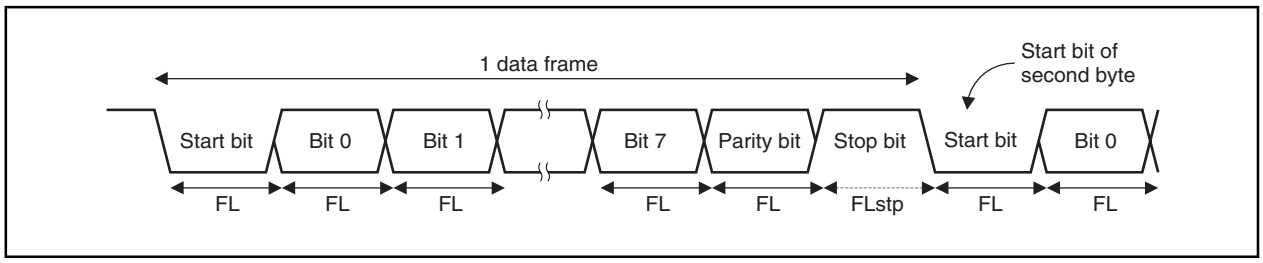
Remarks 1. The reception precision depends on the number of bits in one frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the precision.

2. k: BRGCn setting value (n = 0 to 2)

(5) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

Figure 11-13. Transfer Rate During Continuous Transmission



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the basic clock frequency by f_{XCLK} yields the following equation.

$$FLstp = FL + 2/f_{XCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/f_{XCLK}$$

11.2.7 Cautions

When the supply of clocks to UARTn is stopped (for example, IDLE or STOP mode), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXDn pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting UARTCAEn = 0, RXEn = 0, and TXEn = 0.

11.3 Clocked Serial Interfaces 0 to 2 (CSI0 to CSI2)

11.3.1 Features

- Transfer rate: Master mode: Maximum 3.125 Mbps (when internal system clock operates at 50 MHz)
  Slave mode: Maximum 5 Mbps (when internal system clock operates at 50 MHz)
- Half-duplex communications
- Master mode and slave mode can be selected
- Transmission data length: 8 bits
- Transfer data direction can be switched between MSB first and LSB first
- Eight clock signals can be selected (7 master clocks and 1 slave clock)
- 3-wire method
  SOn: Serial data output
  SIn: Serial data input
  $\overline{\text{SCKn}}$: Serial clock input/output
- Interrupt sources: 1 type
  - Transmission/reception completion interrupt (INTCSIn)
- Transmission/reception mode or reception-only mode can be specified
- On-chip transmit buffer (SOTBn)

Remark n = 0 to 2

11.3.2 Configuration

CSIn is controlled by the clocked serial interface mode register (CSIMn) (n = 0 to 2). Transmit/receive data can be read from or written to the SIOn register.

(1) Clocked serial interface mode registers 0 to 2 (CSIM0 to CSIM2)

The CSIMn register is an 8-bit register for specifying the operation of CSIn.

(2) Clocked serial interface clock selection registers 0 to 2 (CSIC0 to CSIC2)

The CSICn register is an 8-bit register for controlling the transmit operation of CSIn.

(3) Serial I/O shift registers 0 to 2 (SIO0 to SIO2)

The SIOn register is an 8-bit register for converting between serial data and parallel data. SIOn is used for both transmission and reception.

Data is shifted in (reception) or shifted out (transmission) beginning at either the MSB side or the LSB side.

Actual transmit/receive operations are controlled by reading or writing SIOn.

(4) Clocked serial interface transmit buffer registers 0 to 2 (SOTB0 to SOTB2)

The SOTBn register is an 8-bit buffer register for storing transmit data.

(5) Selector

The selector selects the serial clock to be used.

(6) Serial clock controller

The serial clock controller controls the supply of serial clocks to the shift register. When an internal clock is used, it also controls the clocks that are output to the $\overline{\text{SCKn}}$ pin.

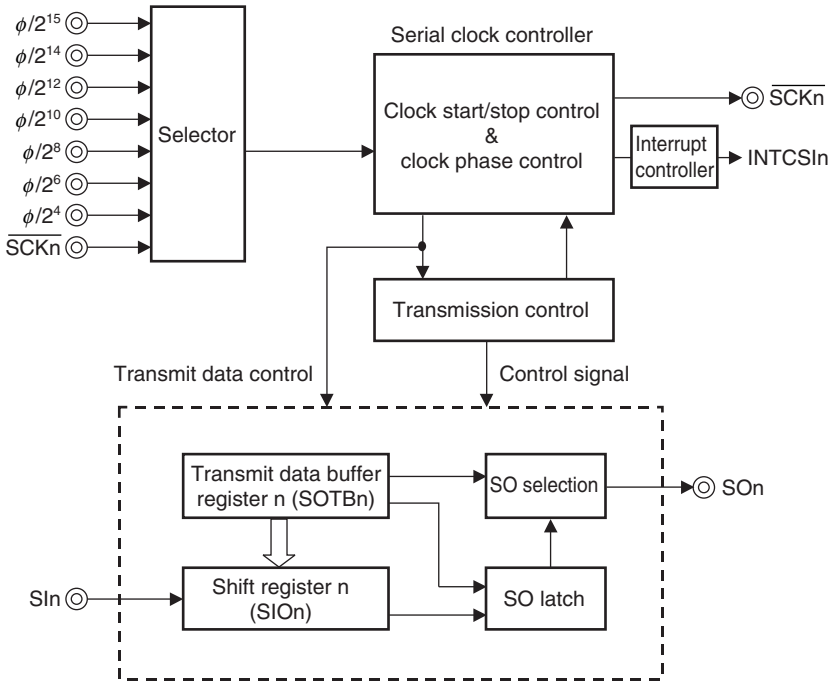
(7) Serial clock counter

The serial clock counter counts serial clocks that are output or input during transmit and receive operations and checks that 8-bit data has been transmitted or received.

(8) Interrupt controller

The interrupt controller controls whether or not an interrupt request is generated when the serial clock counter has counted eight serial clocks.

Figure 11-14. Clocked Serial Interface Block Diagram



Remark n = 0 to 2

11.3.3 Control registers

(1) Clocked serial interface mode registers 0 to 2 (CSIM0 to CSIM2)

The CSIMn register controls the operation of CSIn (n = 0 to 2).

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

Caution To use CSIn, be sure to set the external pins related to the CSIn function to control mode and set the CSICn register. Then set the CSICAEn bit to 1 before setting the other bits.

	<7>	<6>	5	<4>	3	2	1	<0>	Address	After reset
CSIM0	CSICAE0	TRMD0	0	DIR0	0	0	0	CSOT0	FFFFF900H	00H
CSIM1	CSICAE1	TRMD1	0	DIR1	0	0	0	CSOT1	FFFFF910H	00H
CSIM2	CSICAE2	TRMD2	0	DIR2	0	0	0	CSOT2	FFFFF920H	00H

Bit position	Bit name	Function
7	CSICAEn (n = 0 to 2)	CSI Operation Permission/Prohibition Specifies whether CSIn operation is enabled or disabled (n = 0 to 2). 0: CSIn operation is disabled (SOn = low level, $\overline{SCKn}$ = high level) 1: CSIn operation is enabled **Cautions 1. If CSICAEn is set to 0, the CSIn unit can be reset asynchronously.** **2. If CSICAEn = 0, the CSIn unit is in a reset state. Therefore, to operate CSIn, CSICAEn must be set to 1.** **3. If the CSICAEn bit is changed from 1 to 0, all registers of the CSIn unit are initialized. To set CSICAEn to 1 again, the registers of the CSIn unit must be set again.**
6	TRMDn (n = 0 to 2)	Transmission/Reception Mode Control Specifies the transmission/reception mode. 0: Reception-only mode 1: Transmission/reception mode If TRMDn = 0, reception-only transfers are performed. In addition, the SOn pin output is fixed at low level. Data reception is started by reading the SIOn register. If TRMDn = 1, transmission/reception is started by writing data to the SOTBn register. **Caution The TRMDn bit can be overwritten only when CSOTn = 0.**
4	DIRn (n = 0 to 2)	Transmit Direction Mode Control Specifies the transfer direction mode (MSB or LSB). 0: The transfer data's start bit is MSB 1: The transfer data's start bit is LSB **Caution The DIRn bit can be overwritten only when CSOTn = 0.**
0	CSOTn (n = 0 to 2)	CSI Status of Transmission This is a transfer status display flag. 0: Idle status 1: Transfer execution status This flag is used to judge whether writing to the shift register (SIOn) is enabled or not when starting serial data transmission in transmission/reception mode (TRMDn = 1) **Caution The CSOTn bit is reset when the CSICAEn bit is cleared (0).**

(2) Clocked serial interface clock selection registers 0 to 2 (CSIC0 to CSIC2)

The CSICn register is an 8-bit register that controls the transmit operation of CSIn.

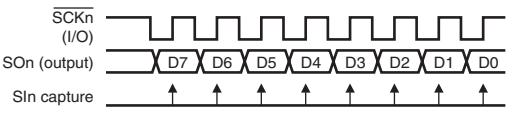
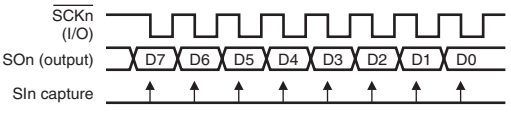
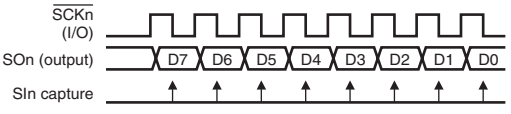
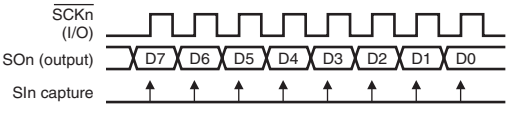
These registers can be read or written in 8-bit units.

Caution The CSIC2 to CSIC0 registers can be overwritten when CSICAEn = 0 in the CSIMn register.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
CSIC0	0	0	0	CKP0	DAP0	CKS02	CKS01	CKS00	FFFFF901H	00H
CSIC1	0	0	0	CKP1	DAP1	CKS12	CKS11	CKS10	FFFFF911H	00H
CSIC2	0	0	0	CKP2	DAP2	CKS22	CKS21	CKS20	FFFFF921H	00H

Bit position	Bit name	Function
4, 3	CKPn, DAPn (n = 0 to 2)	Clock Phase Selection Bit, Data Phase Selection Bit Specifies the data transmission/reception timing for $\overline{\text{SCK}}$n.

CKPn	DAPn	Operation mode
0	0	
0	1	
1	0	
1	1	

(2/2)

Bit position	Bit name	Function
2 to 0	CKSn2 to CKSn0 (n = 0 to 2)	Input Clock Selection Specifies the input clock.

CKSn2	CKSn1	CKSn0	Input clock	Mode
0	0	0	$\phi/2^{15}$	Master mode
0	0	1	$\phi/2^{14}$	Master mode
0	1	0	$\phi/2^{12}$	Master mode
0	1	1	$\phi/2^{10}$	Master mode
1	0	0	$\phi/2^{8}$	Master mode
1	0	1	$\phi/2^{6}$	Master mode
1	1	0	$\phi/2^{4}$	Master mode
1	1	1	External clock (SCKn)	Slave mode

Remark ϕ: Internal system clock frequency

(a) Baud rate

CKSn2	CKSn1	CKSn0	Baud rate (bps)				
			50 MHz operation	40 MHz operation	33 MHz operation	25 MHz operation	20 MHz operation
0	0	0	1,526	1,221	1,007	763	610
0	0	1	3,052	2,441	2,014	1,526	1,221
0	1	0	12,207	9,766	8,057	6,104	4,883
0	1	1	48,828	39,063	32,227	24,414	19,531
1	0	0	195,313	156,250	128,906	97,656	78,125
1	0	1	781,250	625,000	515,625	390,625	312,500
1	1	0	3,125,000	2,500,000	2,062,500	1,562,500	1,250,000

(3) Serial I/O shift registers 0 to 2 (SIO0 to SIO2)

The SIOn register is an 8-bit shift register that converts parallel data to serial data. If TRMDn = 0 in the CSIMn register, the transfer is started by reading SIOn.

Except when a reset is input, the SIOn register becomes 00H even when the CSICAEn bit of the CSIMn register is cleared (0).

These registers are read-only in 8-bit units.

Caution SIOn can be accessed only when the system is in an idle state (CSOTn = 0 in the CSIMn register).

	7	6	5	4	3	2	1	0	Address	After reset
SIO0	SIO07	SIO06	SIO05	SIO04	SIO03	SIO02	SIO01	SIO00	FFFFF902H	00H
SIO1	SIO17	SIO16	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	FFFFF912H	00H
SIO2	SIO27	SIO26	SIO25	SIO24	SIO23	SIO22	SIO21	SIO20	FFFFF922H	00H

Bit position	Bit name	Function
7 to 0	SIOn7 to SIOn0 (n = 0 to 2)	Serial I/O Shifts data in (reception) or shifts data out (transmission) beginning at the MSB or the LSB side.

(4) Receive-only serial I/O shift registers 0 to 2 (SIOE0 to SIOE2)

The SIOEn register is an 8-bit shift register that converts parallel data into serial data. A receive operation does not start even if the SIOEn register is read while the TRMD bit of the CSIMn register is 0. Therefore this register is used to read the value of the SIOn register (receive data) without starting a receive operation.

Except when a reset is input, the SIOEn register becomes 00H even when the CSICAEn bit of the CSIMn register is cleared (0).

These registers are read-only in 8-bit units.

Caution SIOEn can be accessed only when the system is in an idle state (CSOTn = 0 in the CSIMn register).

	7	6	5	4	3	2	1	0	Address	After reset
SIOE0	SIOE07	SIOE06	SIOE05	SIOE04	SIOE03	SIOE02	SIOE01	SIOE00	FFFFF903H	00H
SIOE1	SIOE17	SIOE16	SIOE15	SIOE14	SIOE13	SIOE12	SIOE11	SIOE10	FFFFF913H	00H
SIOE2	SIOE27	SIOE26	SIOE25	SIOE24	SIOE23	SIOE22	SIOE21	SIOE20	FFFFF923H	00H

Bit position	Bit name	Function
7 to 0	SIOEn7 to SIOEn0 (n = 0 to 2)	Serial I/O Shifts data in (reception) beginning at the MSB or the LSB side.

(5) Clocked serial interface transmit buffer registers 0 to 2 (SOTB0 to SOTB2)

The SOTBn register is an 8-bit buffer register for storing transmit data.

If transmission/reception mode is set (TRMDn = 1 in the CSIMn register), a transmit operation is started by writing data to the SOTBn register.

$\overline{\text{RESET}}$ input sets the SOTBn register to 00H.

These registers can be read or written in 8-bit units.

Caution SOTBn can be accessed only when the system is in an idle state (CSOTn = 0 in the CSIMn register).

	7	6	5	4	3	2	1	0	Address	After reset
SOTB0	SOTB07	SOTB06	SOTB05	SOTB04	SOTB03	SOTB02	SOTB01	SOTB00	FFFFF904H	00H
SOTB1	SOTB17	SOTB16	SOTB15	SOTB14	SOTB13	SOTB12	SOTB11	SOTB10	FFFFF914H	00H
SOTB2	SOTB27	SOTB26	SOTB25	SOTB24	SOTB23	SOTB22	SOTB21	SOTB20	FFFFF924H	00H

Bit position	Bit name	Function
7 to 0	SOTBn7 to SOTBn0 (n = 0 to 2)	Serial I/O Writes transmit data.

11.3.4 Operation

(1) Transfer mode

CSIn transmits and receives data in three lines: 1 clock line and 2 data lines.

In reception-only mode (TRMDn = 0 in the CSIMn register), the transfer is started by reading the SIOn register (n = 0 to 2).

In transmission/reception mode (TRMDn = 1 in the CSIMn register), the transfer is started by writing data to the SOTBn register.

★ When an 8-bit transfer of CSIn ends, the CSOTn bit of the CSIMn register becomes 0, and transfer stops automatically. Also, when the transfer ends, a transmission/reception completion interrupt (INTCSIn) is generated.

Cautions 1. When CSOTn = 1 in the CSIMn register, the control registers and data registers should not be accessed.

2. If transmit data is written to the SOTBn register and the TRMDn bit of the CSIMn register is changed from 0 to 1, serial transfer is not performed.

(2) Serial clock

(a) When internal clock is selected as the serial clock

If reception or transmission is started, a serial clock is output from the $\overline{\text{SCKn}}$ pin, and the data of the SIn pin is taken into the SIOn register sequentially or data is output to the SOn pin sequentially from the SIOn register at the timing when the data has been synchronized with the serial clock in accordance with the setting of the CKPn and DAPn bits of the CSICn register (n = 0 to 2).

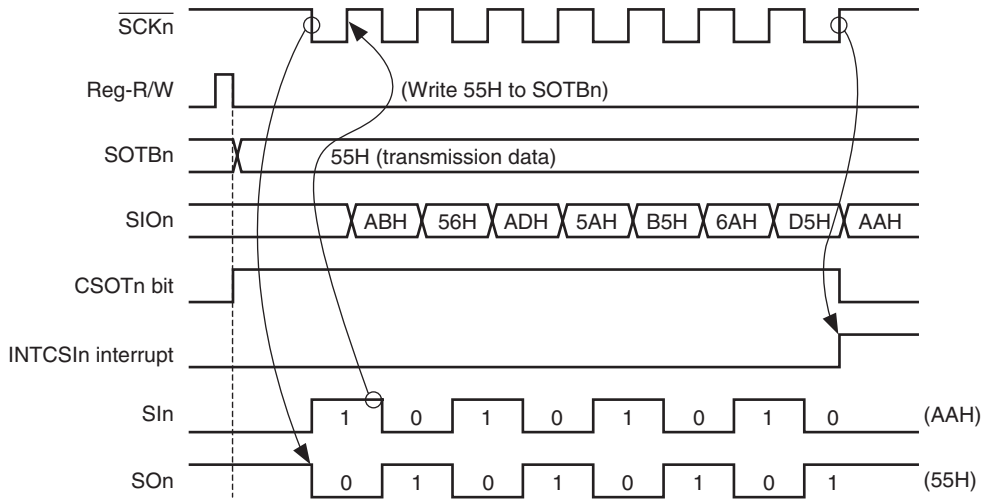
(b) When external clock is selected as the serial clock

If reception or transmission is started, the data of the SIn pin is taken into the SIOn register sequentially or output to the SOn pin sequentially in synchronization with the serial clock that has been input to the $\overline{\text{SCKn}}$ pin following transmission/reception startup in accordance with the setting of the CKPn and DAPn bits of the CSICn register (n = 0 to 2).

If serial clock is input to the $\overline{\text{SCKn}}$ pin when neither reception nor transmission is started, a shift operation will not be executed.

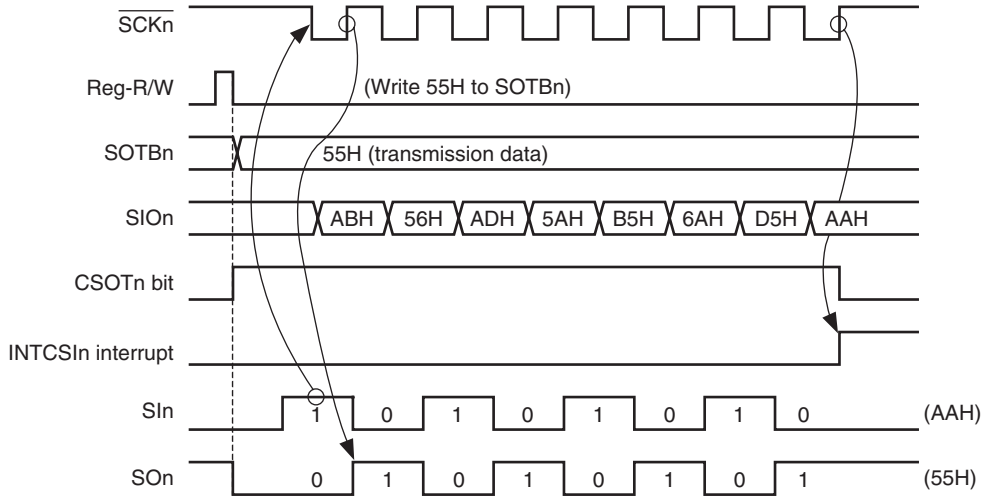
Figure 11-15. Transfer Timing

(a) When TRMDn = 1, DIRn = 0, CKPn = 0, and DAPn = 0



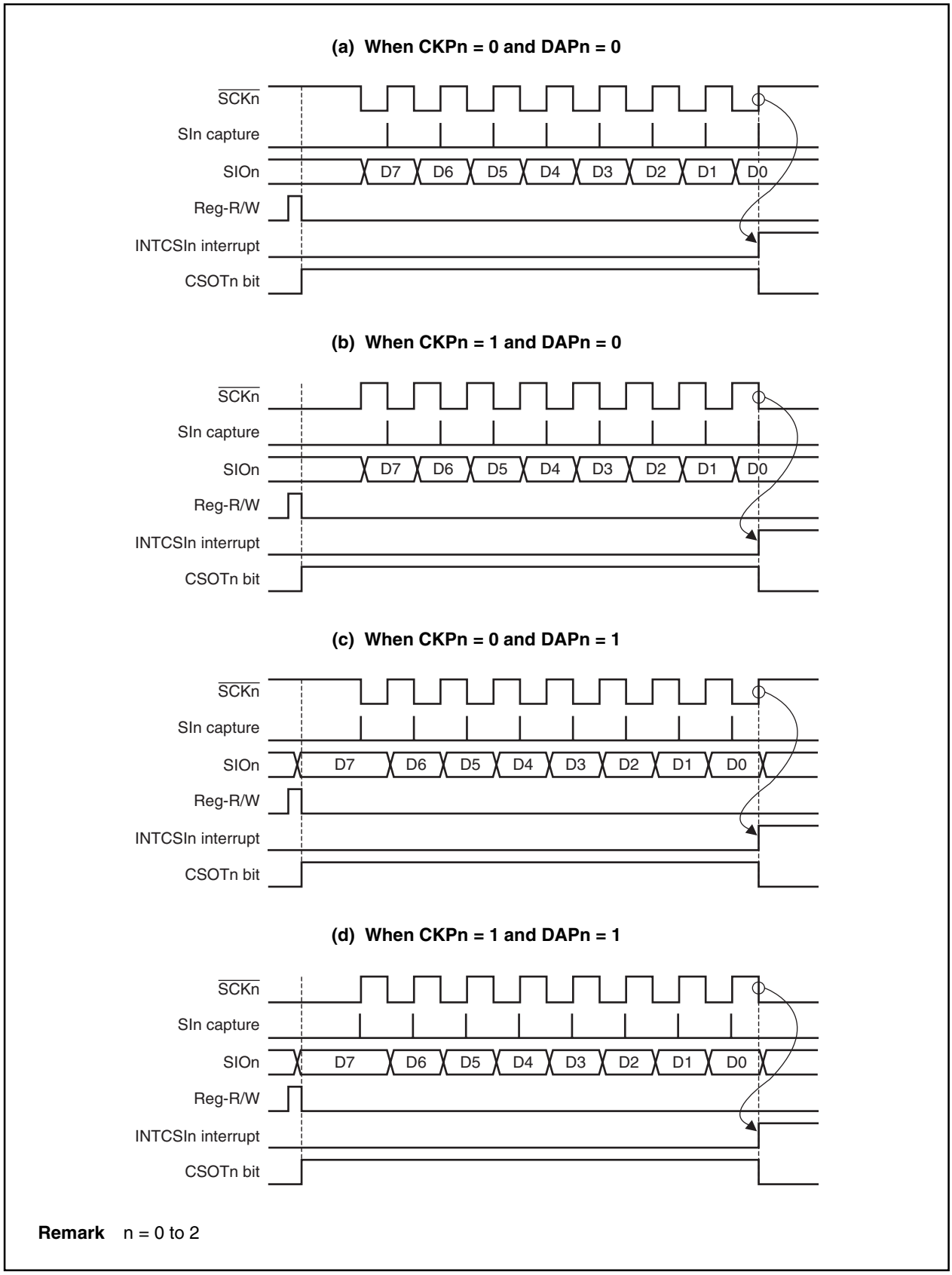
Remark n = 0 to 2

(b) When TRMDn = 1, DIRn = 0, CKPn = 0, and DAPn = 1



Remark n = 0 to 2

Figure 11-16. Clock Timing



Remark n = 0 to 2

11.3.5 Output pins

(1) $\overline{\text{SCKn}}$ pin

When CSIn operation is disabled (CSICAEn = 0), the $\overline{\text{SCKn}}$ pin output state is as follows.

CKPn	$\overline{\text{SCKn}}$ pin output
0	Fixed at high level
1	Fixed at low level

Remarks 1. When the CKPn bit is overwritten, the $\overline{\text{SCKn}}$ pin output changes.

2. n = 0 to 2

(2) SOn pin

When CSIn operation is disabled (CSICAEn = 0), the SOn pin output state is as follows.

TRMDn	DAPn	DIRn	SOn pin output
0	x	x	Fixed at low level
1	0	x	SOn latch value (low level)
	1	0	SOTBn7 value
		1	SOTBn0 value

Remarks 1. If any of the TRMDn, DAPn, and DIRn bits is overwritten, the SOn pin output changes.

2. n = 0 to 2

3. x: don't care

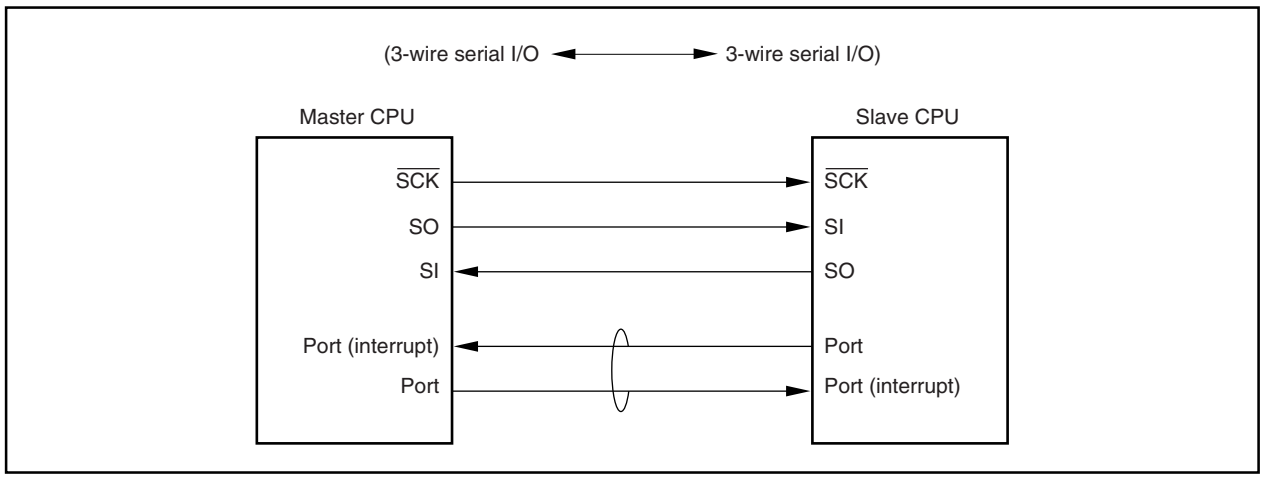
11.3.6 System configuration example

CSIn performs 8-bit length data transfer using three signal lines: a serial clock ($\overline{\text{SCKn}}$), serial input (SIn), and serial output (SOn). This is effective when connecting peripheral I/O that incorporate a conventional clocked serial interface, or a display controller to the V850E/MA1 (n = 2 to 0).

When connecting the V850E/MA1 to several devices, lines for handshake are required.

Since the first communication bit can be selected as an MSB or LSB, communication with various devices can be achieved.

Figure 11-17. System Configuration Example of CSI



CHAPTER 12 A/D CONVERTER

12.1 Features

- Analog input: 8 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
- Successive approximation method

12.2 Configuration

The A/D converter of the V850E/MA1 adopts the successive approximation method, and uses A/D converter mode registers 0, 1, 2 (ADM0, ADM1, ADM2), and the A/D conversion result register (ADCR0 to ADCR7) to perform A/D conversion operations.

(1) Input circuit

The input circuit selects the analog input (ANI0 to ANI7) according to the mode set by the ADM0 and ADM1 registers and sends the input to the sample & hold circuit.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input signals sequentially sent from the input circuit, and sends them to the voltage comparator. This circuit also holds the sampled analog input signal during A/D conversion.

(3) Voltage comparator

The voltage comparator compares the analog input signal with the output voltage of the series resistor string voltage tap.

(4) Series resistor string

The series resistor string is used to generate voltages to match analog inputs.

The series resistor string is connected between the reference voltage pin (AV_{REF}) for the A/D converter and the GND pin (AV_{SS}) for the A/D converter. To make 1,024 equal voltage steps between these 2 pins, it is configured from 1,023 equal resistors and 2 resistors with 1/2 of the resistance value.

The voltage tap of the series resistor string is selected by a tap selector controlled by the successive approximation register (SAR).

(5) Successive approximation register (SAR)

The SAR is a 10-bit register that sets series resistor string voltage tap data, whose values match analog input voltage values, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR all the way to the least significant bit (LSB) (A/D conversion completed), the contents of the SAR (conversion results) are held in the A/D conversion result register (ADCRn).

(6) A/D conversion result register (ADCRn)

ADCRn is a 10-bit register that holds A/D conversion results. Each time A/D conversion is completed, the conversion results are loaded from the successive approximation register (SAR).

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

(7) Controller

The controller selects the analog input, generates the sample & hold circuit operation timing, and controls the conversion trigger according to the mode set by the ADM0 and ADM1 registers.

(8) ANI0 to ANI7 pins

These are 8-channel analog input pins for the A/D converter. They input the analog signals to be A/D converted.

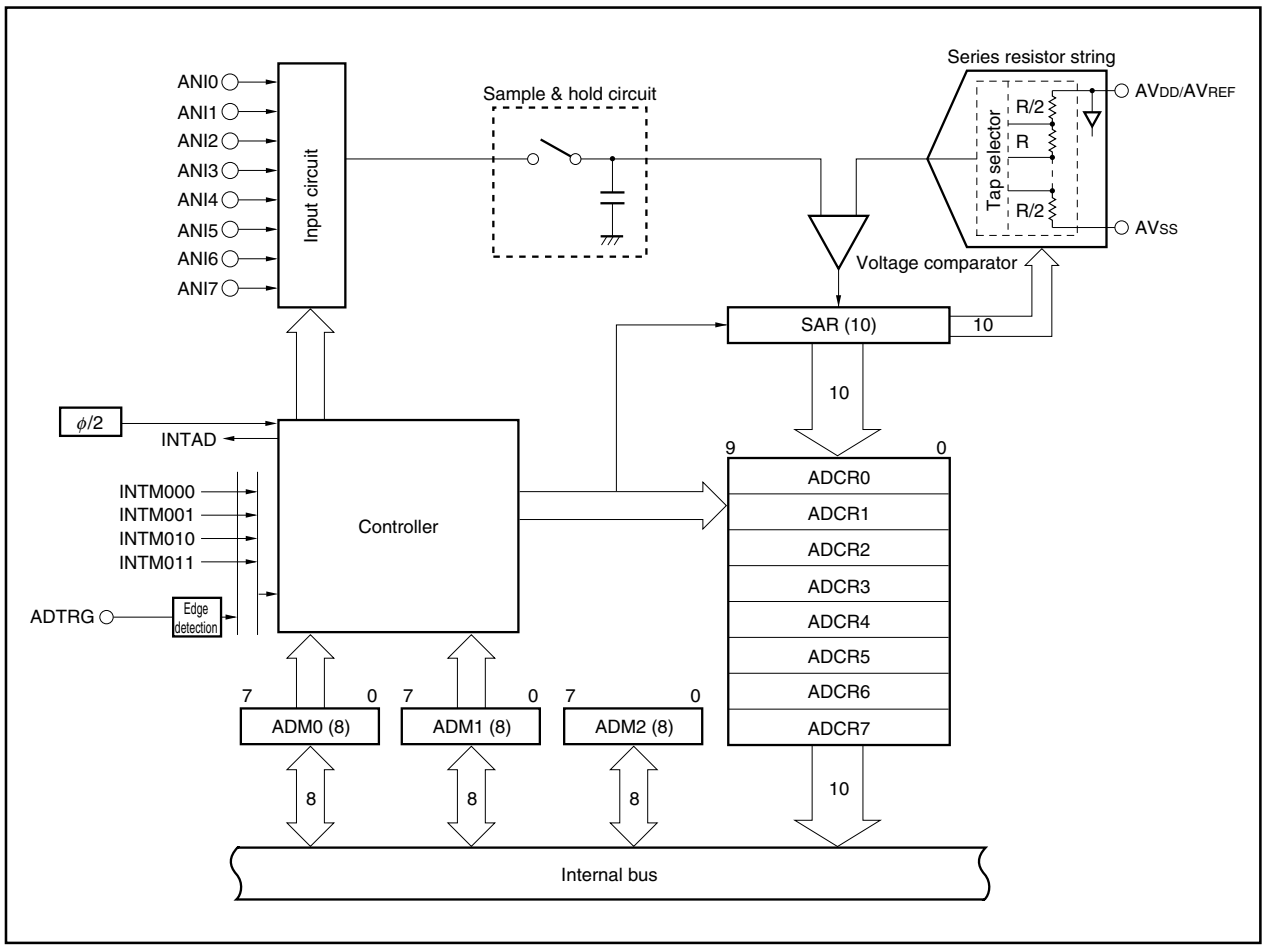
Caution Make sure that the voltages input to ANI0 to ANI7 do not exceed the rated values. If a voltage higher than AV_{DD} or lower than AV_{SS} (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

(9) AV_{REF} pin

This is the pin for inputting the reference voltage of the A/D converter. It converts signals input to the ANIn pin to digital signals based on the voltage applied between AV_{REF} and AV_{SS}.

In the V850E/MA1, the AV_{REF} pin functions alternately as the AV_{DD} pin. It is therefore impossible to set voltage separately for the AV_{REF} pin and the AV_{DD} pin.

Figure 12-1. Block Diagram of A/D Converter



Cautions 1. If there is noise at the analog input pins (ANI0 to ANI7) or at the reference voltage input pin (AV_{REF}), that noise may generate an illegal conversion result.
Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.
An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

2. Do not apply a voltage outside the AV_{SS} to AV_{REF} range to the pins that are used as A/D converter input pins.

12.3 Control Registers

(1) A/D converter mode register 0 (ADM0)

The ADM0 register is an 8-bit register that selects the analog input pin, specifies the operation mode, and executes conversion operations.

This register can be read/written in 8-bit or 1-bit units. However, when data is written to the ADM0 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning. Bit 6 cannot be written to and writing executed is ignored.

Cautions 1. When the ADCE bit is 1 in the timer trigger mode and external trigger mode, the trigger signal standby state is set. To clear the ADCE bit, write "0" or reset.

In the A/D trigger mode, the conversion trigger is set by writing 1 to the ADCE bit. After the operation, when the mode is changed to the timer trigger mode or external trigger mode without clearing the ADCE bit, the trigger input standby state is set immediately after the change.

2. There are 10 clocks between the beginning of conversion and when the ADCS bit becomes 1.

	<7>	<6>	5	4	3	2	1	0	Address	After reset
ADM0	ADCE	ADCS	BS	MS	0	ANIS2	ANIS1	ANIS0	FFFFF200H	00H

Bit position	Bit name	Function
7	ADCE	Convert Enable Enables or disables A/D conversion operation. 0: Disabled 1: Enabled
6	ADCS	Converter Status Indicates the status of A/D converter. This bit is read only. 0: Stopped 1: Operating
5	BS	Buffer Select Specifies buffer mode in the select mode. 0: 1-buffer mode 1: 4-buffer mode
4	MS	Mode Select Specifies operation mode of A/D converter. 0: Scan mode 1: Select mode
2 to 0	ANIS2 to ANIS0	Analog Input Select Specifies the analog input pin to be A/D converted.

ANIS2	ANIS1	ANIS0	Select mode		Scan mode	
			A/D trigger mode	Timer trigger mode	A/D trigger mode	Timer trigger mode[Note]
0	0	0	ANI0	ANI0	ANI0	1
0	0	1	ANI1	ANI1	ANI0, ANI1	2
0	1	0	ANI2	ANI2	ANI0 to ANI2	3
0	1	1	ANI3	ANI3	ANI0 to ANI3	4
1	0	0	ANI4	Setting prohibited	ANI0 to ANI4	4 + ANI4
1	0	1	ANI5	Setting prohibited	ANI0 to ANI5	4 + ANI4, ANI5
1	1	0	ANI6	Setting prohibited	ANI0 to ANI6	4 + ANI4 to ANI6
1	1	1	ANI7	Setting prohibited	ANI0 to ANI7	4 + ANI4 to ANI7

Note In the timer trigger mode (4-trigger mode) in the scan mode, because the scanning sequence of the ANI0 to ANI3 pins is specified by the sequence in which the match signals are generated from the compare register, the number of trigger inputs should be specified instead of specifying a certain analog input pin. When ANIS2 = 1, perform conversion after shifting to A/D trigger scan mode, which is possible after the trigger has been counted 4 times.

(2) A/D converter mode register 1 (ADM1)

The ADM1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

This register can be read/written in 8-bit units. However, when data is written to the ADM1 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

	7	6	5	4	3	2	1	0	Address	After reset
ADM1	0	TRG2	TRG1	TRG0	0	FR2	FR1	FR0	FFFFF201H	07H

Bit position	Bit name	Function
6 to 4	TRG2 to TRG0	Trigger Mode Specifies the trigger mode.

TRG2	TRG1	TRG0	Trigger mode
0	0	0/1	A/D trigger mode
0	1	0	Timer trigger mode (1-trigger mode)
0	1	1	Timer trigger mode (4-trigger mode)
1	1	1	External trigger mode
Other than above			Setting prohibited

Remark The valid edge of the external input signal in the external trigger mode is specified by bits 7 and 6 (ES1231, ES1230) of the external interrupt mode register (INTM3). For details, refer to **7.3.9 (1) External interrupt mode registers 1 to 4 (INTM1 to INTM4)**.

2 to 0 — FR2 to FR0

Frequency

Specifies the conversion operation time. These bits control the conversion time so that it is the same value irrespective of the oscillation frequency.

FR2	FR1	FR0	Number of conversion clocks	Conversion operation time[Note]		
				ϕ = 50 MHz	ϕ = 40 MHz	ϕ = 33 MHz
0	0	0	96	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1	144	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	192	Setting prohibited	Setting prohibited	5.82 μs
0	1	1	240	4.80 μs	6.00 μs	7.27 μs
1	0	0	336	6.72 μs	8.40 μs	10.18 μs
1	0	1	384	7.68 μs	9.60 μs	Setting prohibited
1	1	0	480	9.60 μs	Setting prohibited	Setting prohibited
1	1	1	672	Setting prohibited	Setting prohibited	Setting prohibited

★ **Note** Figures in the conversion operation time are target values. Set the conversion operation time in the range of 5 to 10 μs.

Remark ϕ = internal system clock frequency

(3) A/D converter mode register 2 (ADM2)

The ADM2 register is an 8-bit register that controls the reset and clock of the A/D converter.

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

★ **Caution Because ADCAE = 0 after reset release, the A/D converter enters the reset state. When operating the A/D converter, be sure to write to the ADM0 and ADM1 registers after setting the ADCAE bit of the ADM2 register to 1 (it is impossible to write to the ADM0 and ADM1 registers when ADCAE = 0). Moreover, when the ADCAE bit is set to 0, all registers related to the A/D converter are initialized.**

	7	6	5	4	3	2	1	<0>	Address	After reset
ADM2	0	0	0	0	0	0	0	ADCAE	FFFFF202H	00H

Bit position	Bit name	Function
0	ADCAE	Clock Action Enable Controls the A/D converter operation. 0: Clock supply to the A/D converter is stopped, the A/D converter is in the reset state 1: The clock is supplied to the A/D converter, A/D converter operation is enabled

(4) A/D conversion result registers (ADCR0 to ADCR7, ADCR0H to ADCR7H)

The ADCRn register is a 10-bit register holding the A/D conversion results. There are eight 10-bit registers. These registers are read-only in 16-bit or 8-bit units. During 16-bit access, the ADCRn register is specified, and during higher 8-bit access, the ADCRnH register is specified (n = 0 to 7).

When reading the 10-bit data of the A/D conversion results from the ADCRn register during 16-bit access, only the lower 10 bits are valid and the higher 6 bits are always read as 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADCRn	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFFFF210H to FFFFF21EH	0000H

	7	6	5	4	3	2	1	0	Address	After reset
ADCRnH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	FFFFF220H to FFFFF227H	00H

Remark n = 0 to 7

The correspondence between each analog input pin and the ADCRn register (except the 4-buffer mode) is shown below.

Analog Input Pin	ADCRn Register
ANI0	ADCR0, ADCR0H
ANI1	ADCR1, ADCR1H
ANI2	ADCR2, ADCR2H
ANI3	ADCR3, ADCR3H
ANI4	ADCR4, ADCR4H
ANI5	ADCR5, ADCR5H
ANI6	ADCR6, ADCR6H
ANI7	ADCR7, ADCR7H

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (of the A/D conversion result register (ADCRn)) is as follows:

$$ADCR = INT\left(\frac{V_{IN}}{AV_{REF}} \times 1{,}024 + 0.5\right)$$

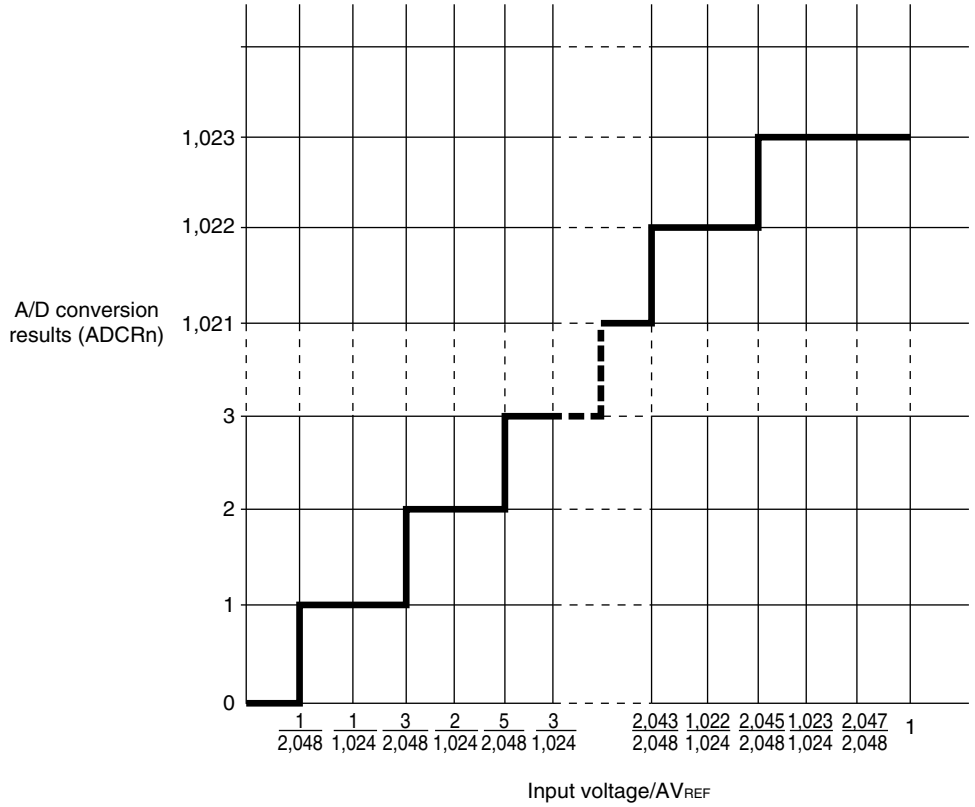
or,

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1{,}024} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: Value of A/D conversion result register (ADCRn)

Figure 12-2 shows the relationship between the analog input voltage and the A/D conversion results.

Figure 12-2. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark n = 0 to 7

12.4 A/D Converter Operation

12.4.1 Basic operation of A/D converter

A/D conversion is executed by the following procedure.

(1) The ADCAE bit of the ADM2 register is set (1).

(2) The selection of the analog input and specification of the operation mode, trigger mode, etc. should be specified using the ADM0 and ADM1 registers[Note 1].
When the ADCE bit of the ADM0 register is set (1), A/D conversion starts in the A/D trigger mode. In the timer trigger mode and external trigger mode, the trigger standby state[Note 2] is set.

(3) The voltage generated from the voltage tap of the series resistor string and analog input are compared by the comparator.

(4) When the comparison of the 10 bits ends, the conversion results are stored in the ADCRn register. When A/D conversion has been performed the specified number of times, the A/D conversion end interrupt (INTAD) is generated (n = 0 to 7).

Notes 1. When the ADM0 to ADM2 registers are changed during the A/D conversion operation, the A/D conversion operation before the change is stopped and the conversion results are not stored in the ADCRn register.

2. During the timer trigger mode and external trigger mode, if the ADCE 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.

12.4.2 Operation mode and trigger mode

Various conversion operations can be specified for the A/D converter by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADM0 and ADM1 registers.

The following shows the relationship between the operation mode and trigger mode.

Trigger Mode		Operation Mode		Setting Value		Analog Input
				ADM0	ADM1	
A/D trigger		Select	1 buffer	xx010xxxB	000x0xxxB	ANI0 to ANI7
			4 buffers	xx110xxxB	000x0xxxB	
		Scan		xxx00xxxB	000x0xxxB	
Timer trigger	1 trigger	Select	1 buffer	xx010xxxB	00100xxxB	ANI0 to ANI3
			4 buffers	xx110xxxB	00100xxxB	
		Scan		xxx00xxxB	00100xxxB	
	4 trigger	Select	1 buffer	xx010xxxB	00110xxxB	
			4 buffers	xx110xxxB	00110xxxB	
		Scan		xxx00xxxB	00110xxxB	
External trigger		Select	1 buffer	xx010xxxB	01100xxxB	
			4 buffers	xx110xxxB	01100xxxB	
		Scan		xxx00xxxB	01100xxxB	

(1) Trigger mode

There are three types of trigger modes that serve as the start timing of A/D conversion processing: A/D trigger mode, timer trigger mode, and external trigger mode. The ANI0 to ANI3 pins are able to specify all of these modes, but the ANI4 to ANI7 pins can only specify the A/D trigger mode. The timer trigger mode consists of the 1-trigger mode and 4-trigger mode as the sub-trigger modes. These trigger modes are set by the ADM1 register.

(a) A/D trigger mode

This mode starts the conversion timing of the analog input set to the ANI0 to ANI7 pins, and by setting the ADCE bit of the ADM0 register to 1, starts A/D conversion. The ANI4 to ANI7 pins are always set in this mode.

(b) Timer trigger mode

Specifies the conversion timing of the analog input set for the ANI0 to ANI3 pins using the values set to the timer C compare register. This mode can only be specified by pins ANI0 to ANI3.

This register creates the analog input conversion timing by generating the match interrupts of the four capture/compare registers (CCC00, CCC01, CCC10, CCC11) connected to the 16-bit timer C (TMC0, TMC1).

There are two sub-trigger modes: 1-trigger mode and 4-trigger mode.

- **1-trigger mode**
  A mode that uses one match interrupt from timer C as the A/D conversion start timing.

- **4-trigger mode**
  A mode that uses four match interrupts from timer C as the A/D conversion start timing.

(c) External trigger mode

A mode that specifies the conversion timing of the analog input to the ANI0 to ANI3 pins using the ADTRG pin. This mode can be specified only with the ANI0 to ANI3 pins.

(2) Operation mode

There are two operation modes that set the ANI0 to ANI7 pins: select mode and scan mode. The select mode has sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the ADM0 register.

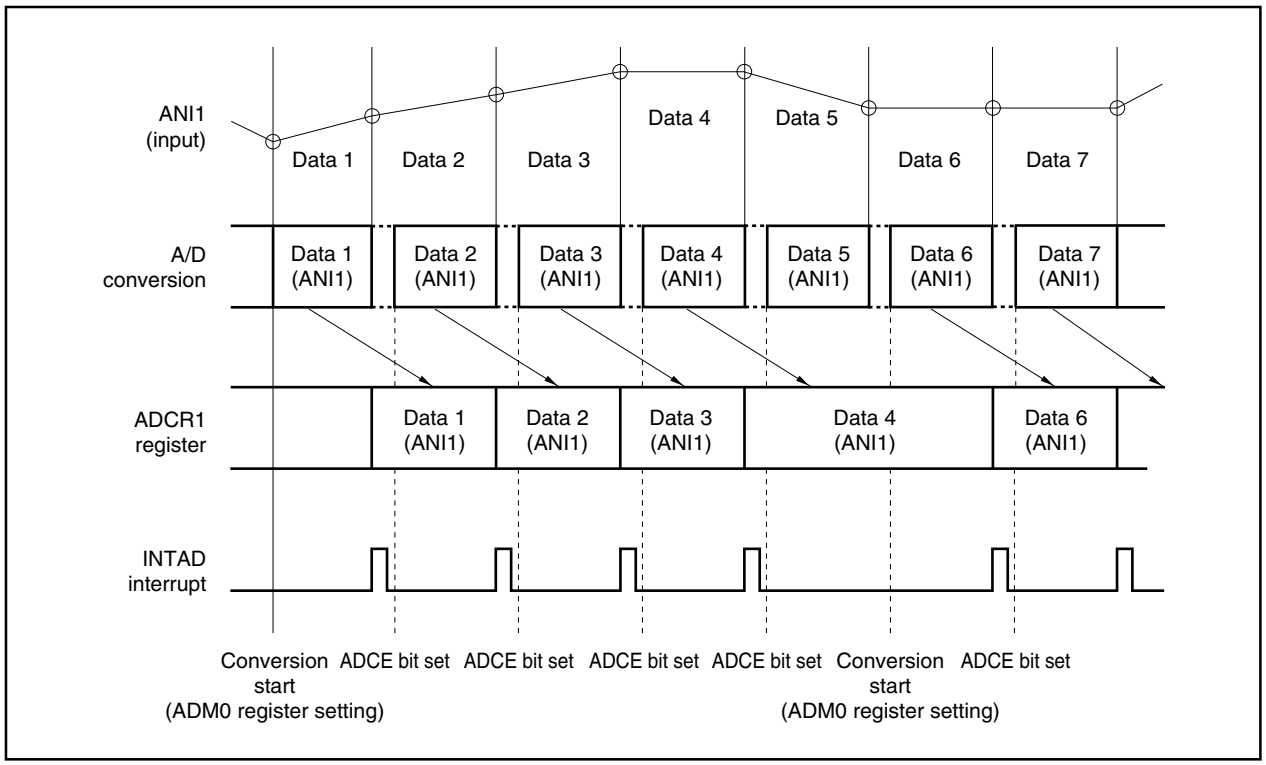
(a) Select mode

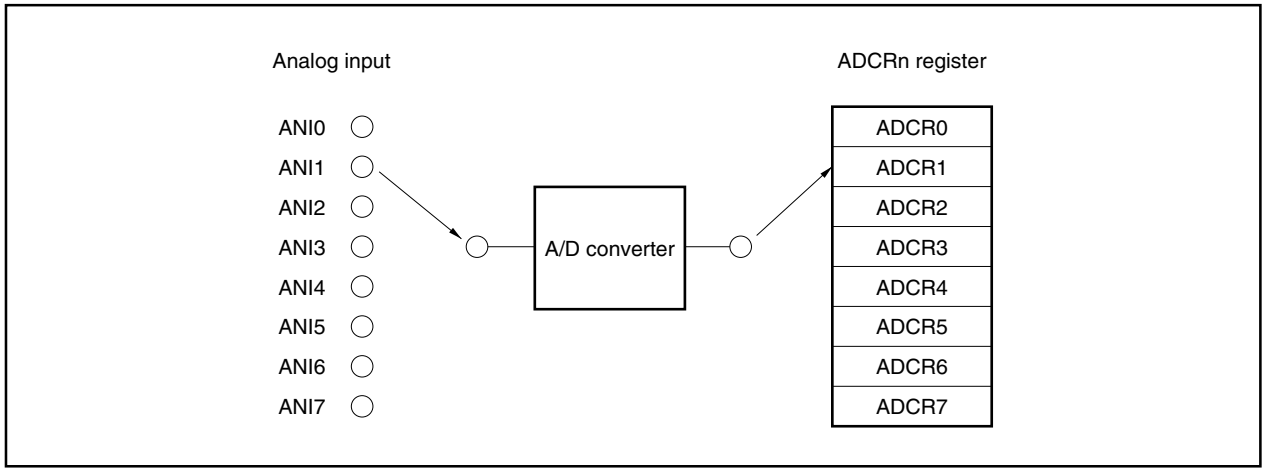
In this mode, one analog input specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input (ANIn). For this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results (n = 0 to 7).

- **1-buffer mode**

  In this mode, one analog input specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input (ANIn). The ANIn 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 12-3. Select Mode Operation Timing: 1-Buffer Mode (ANI1)

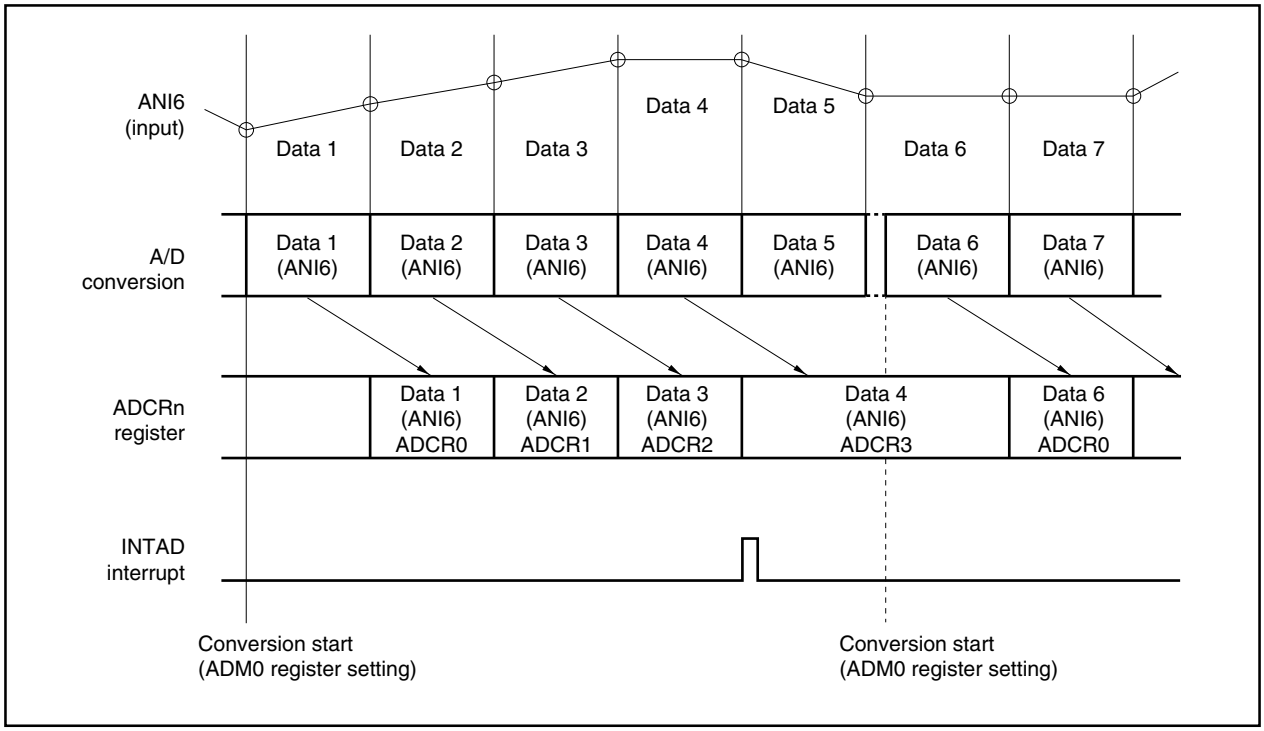


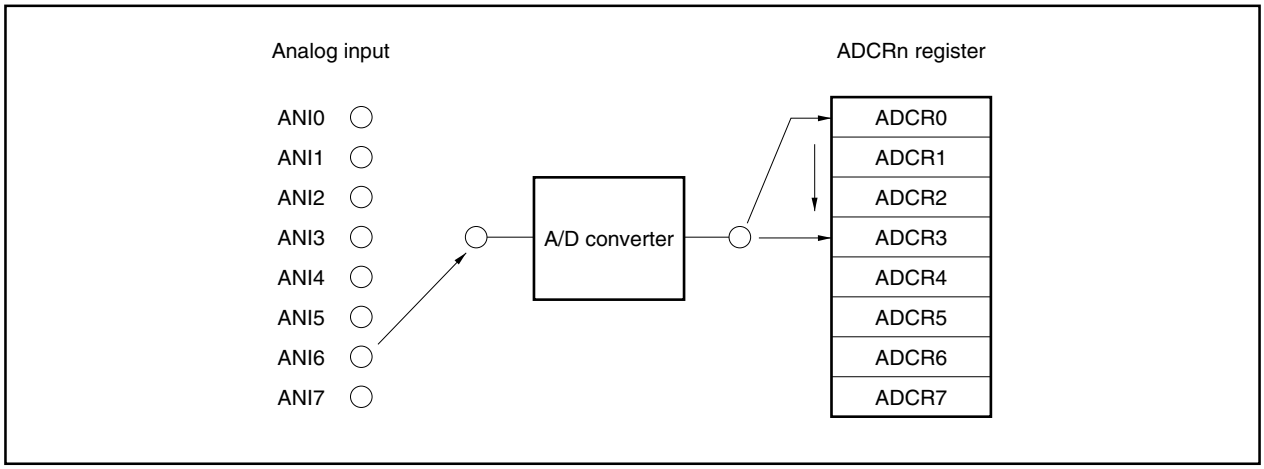


- **4-buffer mode**

  In this mode, one analog input is A/D converted four times and the results are stored in the ADCR0 to ADCR3 registers. The A/D conversion end interrupt (INTAD) is generated when the four A/D conversions end.

Figure 12-4. Select Mode Operation Timing: 4-Buffer Mode (ANI6)

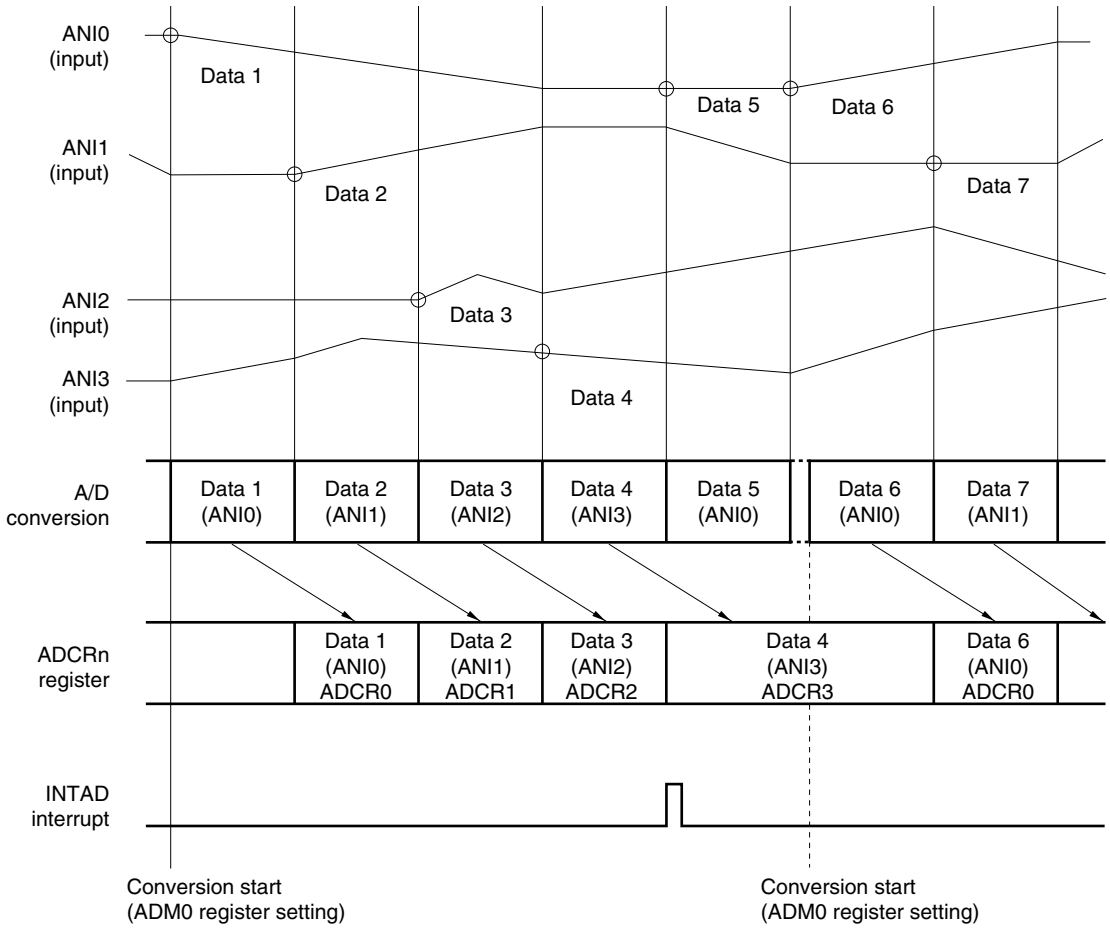


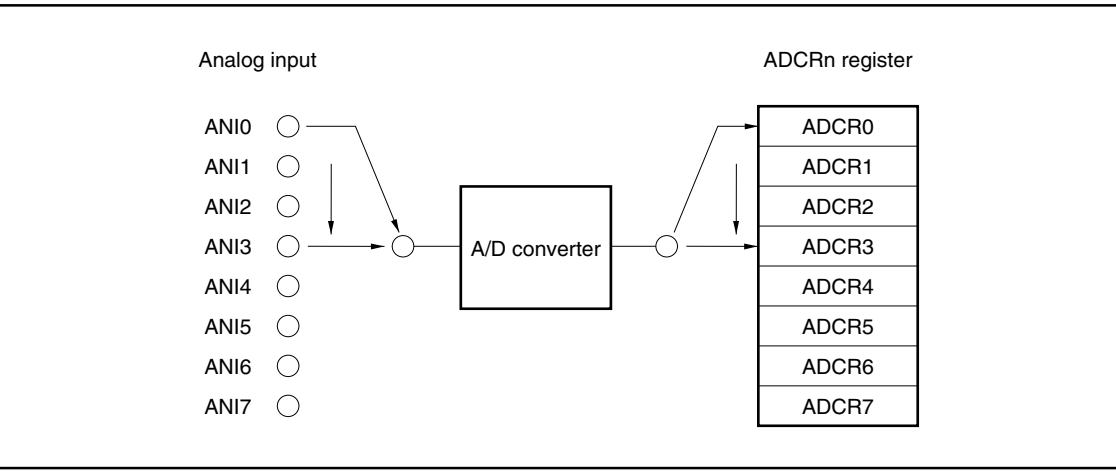


(b) Scan mode

In this mode, the analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7). When the conversion of the specified analog input ends, the A/D conversion end interrupt (INTAD) is generated.

Figure 12-5. Scan Mode Operation Timing: 4-Channel Scan (ANI0 to ANI3)





12.5 Operation in A/D Trigger Mode

When the ADCE bit of the ADM0 register is set to 1, A/D conversion is started.

12.5.1 Select mode operation

In this mode, the analog input specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are supported according to the storing method of the A/D conversion results (n = 0 to 7).

(1) 1-buffer mode (A/D trigger select: 1 buffer)

In this mode, one analog input is A/D converted once. The conversion results are stored in one ADCRn register. The analog input and ADCRn register correspond one to one.

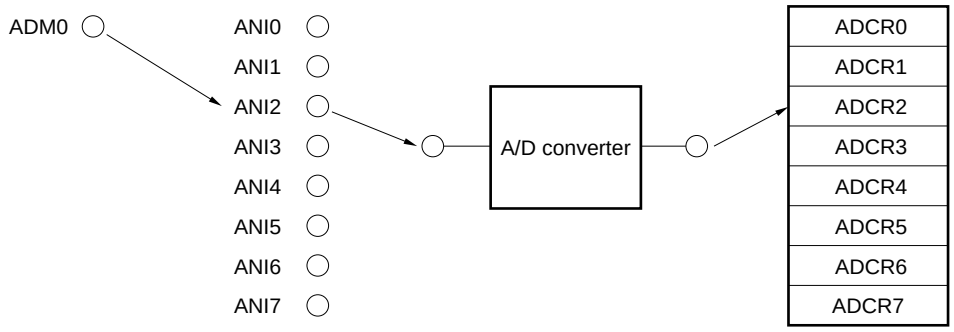
Each time an A/D conversion is executed, an A/D conversion end interrupt (INTAD) is generated and A/D conversion ends.

Analog Input	A/D Conversion Result Register
ANIn	ADCRn

If 1 is written in the ADCE bit of the ADM0 register, A/D conversion can be restarted.

This mode is most appropriate for applications in which the results of each first-time A/D conversion are read.

Figure 12-6. Example of 1-Buffer Mode Operation (A/D Trigger Select: 1 Buffer)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) ANI2 is A/D converted
(3) The conversion result is stored in ADCR2
(4) The INTAD interrupt is generated

(2) 4-buffer mode (A/D trigger select: 4 buffers)

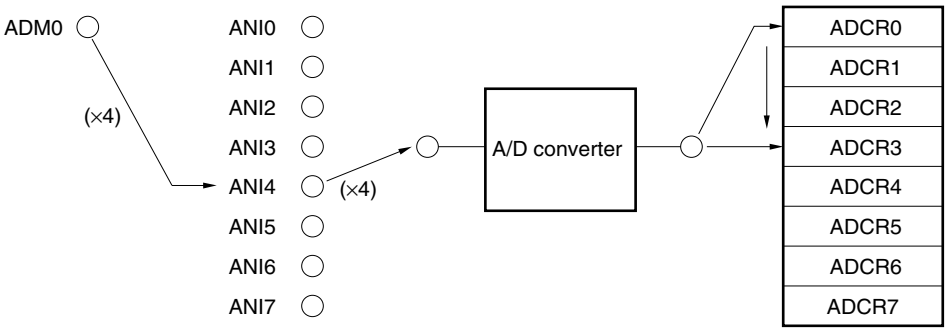
In this mode, one analog input is A/D converted four times and the results are stored in the ADCR0 to ADCR3 registers. When the 4th A/D conversion ends, an A/D conversion end interrupt (INTAD) is generated and the A/D conversion is stopped.

Analog Input	A/D Conversion Result Register
ANIn	ADCR0
ANIn	ADCR1
ANIn	ADCR2
ANIn	ADCR3

If 1 is written in the ADCE bit of the ADM0 register, A/D conversion can be restarted.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Figure 12-7. Example of 4-Buffer Mode Operation (A/D Trigger Select: 4 Buffers)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) ANI4 is A/D converted
(3) The conversion result is stored in ADCR0
(4) ANI4 is A/D converted
(5) The conversion result is stored in ADCR1
(6) ANI4 is A/D converted
(7) The conversion result is stored in ADCR2
(8) ANI4 is A/D converted
(9) The conversion result is stored in ADCR3
(10) The INTAD interrupt is generated

12.5.2 Scan mode operations

In this mode, the analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7).

When conversion of all the specified analog input ends, the A/D conversion end interrupt (INTAD) is generated, and A/D conversion is stopped.

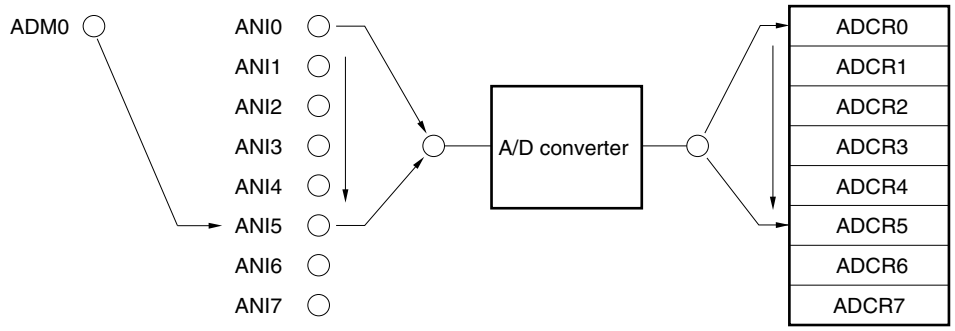
Analog Input	A/D Conversion Result Register
ANIn	ADCR0
⋮	⋮
ANIn[Note]	ADCRn

Note Set by the ANI0 to ANI2 bits of the ADM0 register.

If 1 is written in the ADCE bit of the ADM0 register, A/D conversion can be restarted.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

Figure 12-8. Example of Scan Mode Operation (A/D Trigger Scan)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) ANI0 is A/D converted
(3) The conversion result is stored in ADCR0
(4) ANI1 is A/D converted
(5) The conversion result is stored in ADCR1
(6) ANI2 is A/D converted
(7) The conversion result is stored in ADCR2
(8) ANI3 is A/D converted
(9) The conversion result is stored in ADCR3
(10) ANI4 is A/D converted
(11) The conversion result is stored in ADCR4
(12) ANI5 is A/D converted
(13) The conversion result is stored in ADCR5
(14) The INTAD interrupt is generated

12.6 Operation in Timer Trigger Mode

Conversion timings for up to four-channel analog inputs (ANI0 to ANI3) can be set for the A/D converter using the interrupt signal output from the TMC compare register.

Two 16-bit timers (TMC0, TMC1) and four capture/compare registers (CCC00, CCC01, CCC10, CC11) are used for the timer to specify the analog conversion trigger.

The following two modes are provided according to the value set in the TMCC01 or TMCC11 register.

(1) 1-shot mode

To use the 1-shot mode, set the OSTn bit of the TMCCn1 register (overflow stop mode) to 1 (n = 0, 1).

When TMC overflows, 0000H is held, and counter operation stops. Thereafter, TMCn does not output the match interrupt signal (A/D conversion trigger) of the compare register, and the A/D converter enters the A/D conversion standby state. The TMCn count operation restarts when the TMCCEn bit of the TMCCn0 register is set to 1. The 1-shot mode is used when the A/D conversion cycle is longer than the TMC cycle. (n = 0, 1).

(2) Loop mode

To use the loop mode, set the OST bit (free-running mode) of the TMCCn1 register to 0 (n = 0, 1).

When TMCn overflows, it starts counting from 0000H again, and the match interrupt signal (A/D conversion trigger) of the compare register is repeatedly output. A/D conversion is also repeated.

12.6.1 Select mode operation

In this mode, an analog input (ANI0 to ANI3) specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register. In the select mode, the 1-buffer mode and 4-buffer mode are provided according to the storing method of the A/D conversion results (n = 0 to 3).

(1) 1-buffer mode operation (timer trigger select: 1 buffer)

In this mode, one analog input is A/D converted once and the conversion results are stored in one ADCRn register.

There are two modes in the 1-buffer mode: 1-trigger mode and 4-trigger mode, according to the number of triggers.

(a) 1-trigger mode (timer trigger select: 1 buffer, 1 trigger)

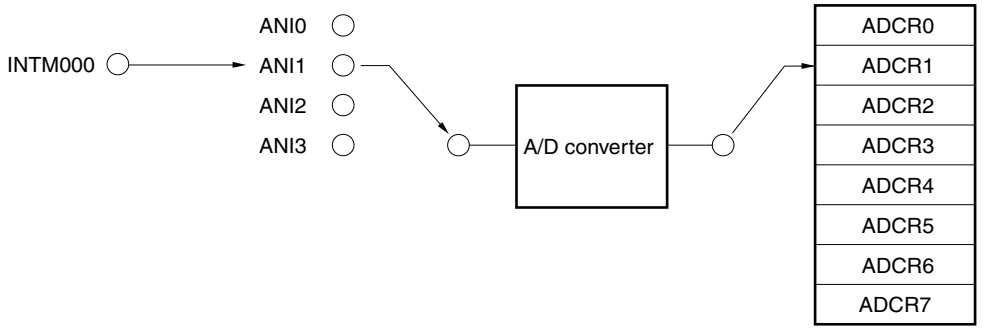
In this mode, one analog input is A/D converted once using the trigger of the match interrupt signal (INTM000) and the results are stored in one ADCRn register. An A/D conversion end interrupt (INTAD) is generated for each A/D conversion and A/D conversion is stopped (n = 0 to 3).

Trigger	Analog Input	A/D Conversion Result Register
INTM000 interrupt	ANIn	ADCRn

In 1-shot mode, A/D conversion stops after one conversion. To restart A/D conversion, set the TMCCEn bit of the TMCCn0 register to 1 (n = 0, 1).

When set to the loop mode, unless the ADCE bit of the ADM0 register is set to 0, A/D conversion is repeated each time a match interrupt is generated.

Figure 12-9. Example of 1-Trigger Mode Operation (Timer Trigger Select: 1 Buffer 1 Trigger)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The CCC00 compare is generated
(3) ANI1 is A/D converted
(4) The conversion result is stored in ADCR1
(5) The INTAD interrupt is generated

(b) 4-trigger mode (timer trigger select: 1 buffer, 4 triggers)

In this mode, one analog input is A/D converted four times using four match interrupt signals (INTM000, INTM001, INTM010, INTM011) as triggers and the results are stored in one ADCRn register. The A/D conversion end interrupt (INTAD) is generated with each A/D conversion, and the ADCS bit of the ADM0 register is reset (0). The results of one A/D conversion are held in the ADCRn register until the next A/D conversion ends. Perform transmission of the conversion results to the memory and other operations using the INTAD interrupt after each A/D conversion ends (n = 0 to 3).

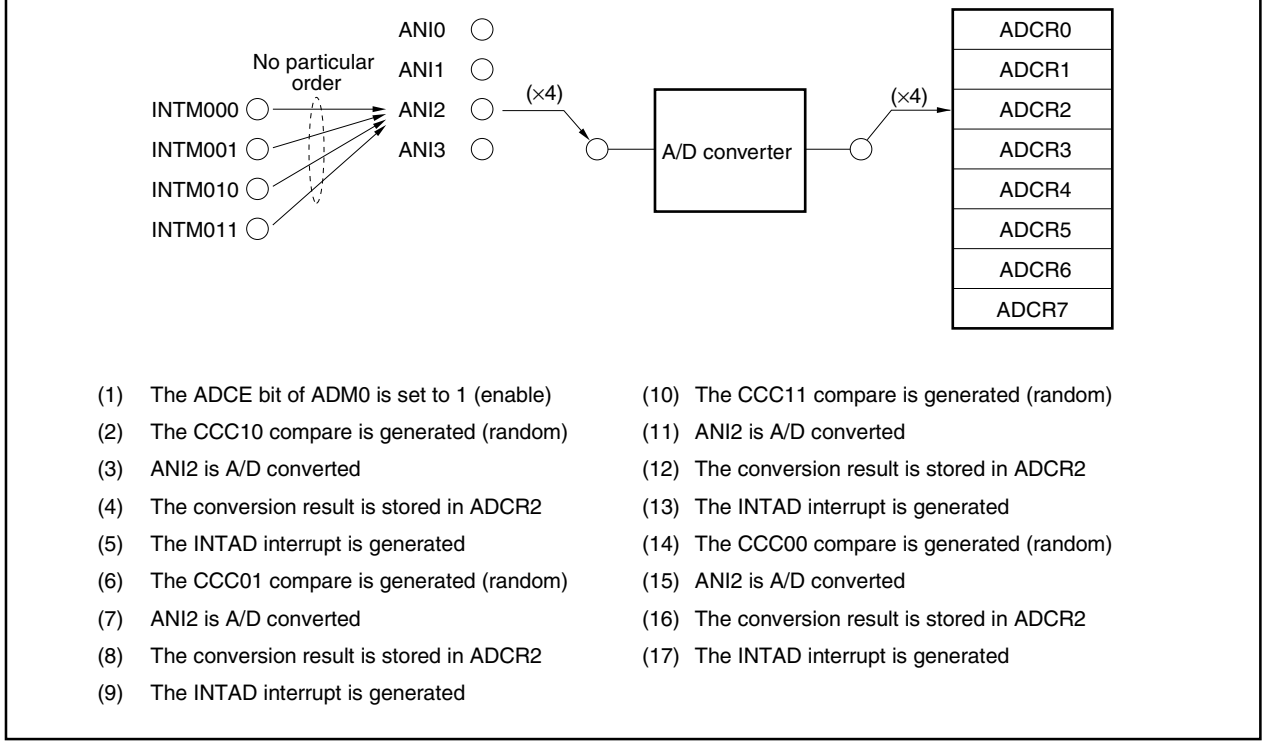
Trigger	Analog Input	A/D Conversion Result Register
INTM000 interrupt	ANIn	ADCRn
INTM001 interrupt	ANIn	ADCRn
INTM010 interrupt	ANIn	ADCRn
INTM011 interrupt	ANIn	ADCRn

In 1-shot mode, A/D conversion stops after four conversions. To restart A/D conversion, set the TMCCEn bit of the TMCCn0 register to 1 to restart the TMCn. When the first match interrupt after TMCn is restarted is generated, the ADCS bit is set (1) and A/D conversion is started (n = 0, 1).

When set to the loop mode, unless the ADCE bit of the ADM0 register is set to 0, A/D conversion is repeated each time a match interrupt is generated.

The match interrupts (INTM000, INTM001, INTM010, INTM011) can be generated in any order. Also, even in cases where the same trigger is input continuously, it is received as a trigger.

Figure 12-10. Example of 4-Trigger Mode Operation (Timer Trigger Select: 1 Buffer 4 Triggers)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The CCC10 compare is generated (random)
(3) ANI2 is A/D converted
(4) The conversion result is stored in ADCR2
(5) The INTAD interrupt is generated
(6) The CCC01 compare is generated (random)
(7) ANI2 is A/D converted
(8) The conversion result is stored in ADCR2
(9) The INTAD interrupt is generated
(10) The CCC11 compare is generated (random)
(11) ANI2 is A/D converted
(12) The conversion result is stored in ADCR2
(13) The INTAD interrupt is generated
(14) The CCC00 compare is generated (random)
(15) ANI2 is A/D converted
(16) The conversion result is stored in ADCR2
(17) The INTAD interrupt is generated

(2) 4-buffer mode operation (timer trigger select: 4 buffers)

In this mode, A/D conversion of one analog input is executed four times, and the results are stored in the ADCR0 to ADCR3 registers. There are two 4-buffer modes: 1-trigger mode and 4-trigger mode, according to the number of triggers.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

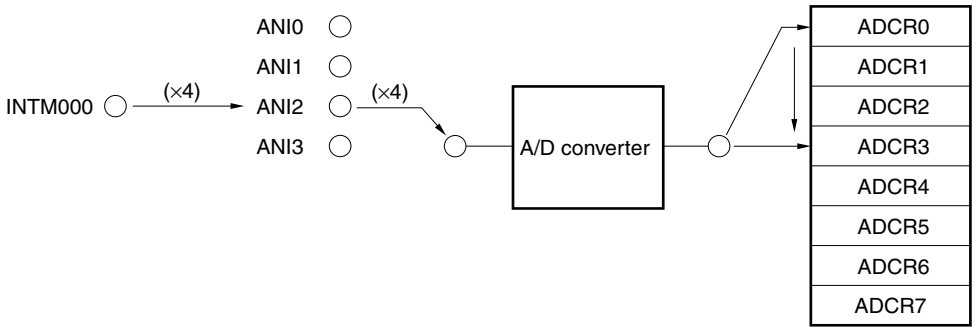
(a) 1-trigger mode

In this mode, one analog input is A/D converted four times using the match interrupt signal (INTM000) as a trigger, and the results are stored in ADCR0 to ADCR3 registers. The A/D conversion end interrupt (INTAD) is generated when the four A/D conversions end and A/D conversion is stopped.

Trigger	Analog Input	A/D Conversion Result Register
INTM000 interrupt	ANIn	ADCR0
INTM000 interrupt	ANIn	ADCR1
INTM000 interrupt	ANIn	ADCR2
INTM000 interrupt	ANIn	ADCR3

If the one-shot mode is set and the TMCCEn bit of the TMCCn0 register is set to 1, and if the match interrupt occurs less than four times, the INTAD interrupt does not occur and the standby state is set (n = 0, 1).

Figure 12-11. Example of 1-Trigger Mode Operation (Timer Trigger Select: 4 Buffers 1 Trigger)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The CCC00 compare is generated
(3) ANI2 is A/D converted
(4) The conversion result is stored in ADCR0
(5) The CCC00 compare is generated
(6) ANI2 is A/D converted
(7) The conversion result is stored in ADCR1
(8) The CCC00 compare is generated
(9) ANI2 is A/D converted
(10) The conversion result is stored in ADCR2
(11) The CCC00 compare is generated
(12) ANI2 is A/D converted
(13) The conversion result is stored in ADCR3
(14) The INTAD interrupt is generated

(b) 4-trigger mode

In this mode, one analog input is A/D converted four times using four match interrupt signals (INTM000, INTM001, INTM010, INTM011) as triggers and the results are stored in four ADCRn registers. The A/D conversion end interrupt (INTAD) is generated when the four A/D conversions end, the ADCS bit is reset (0), and A/D conversion is stopped.

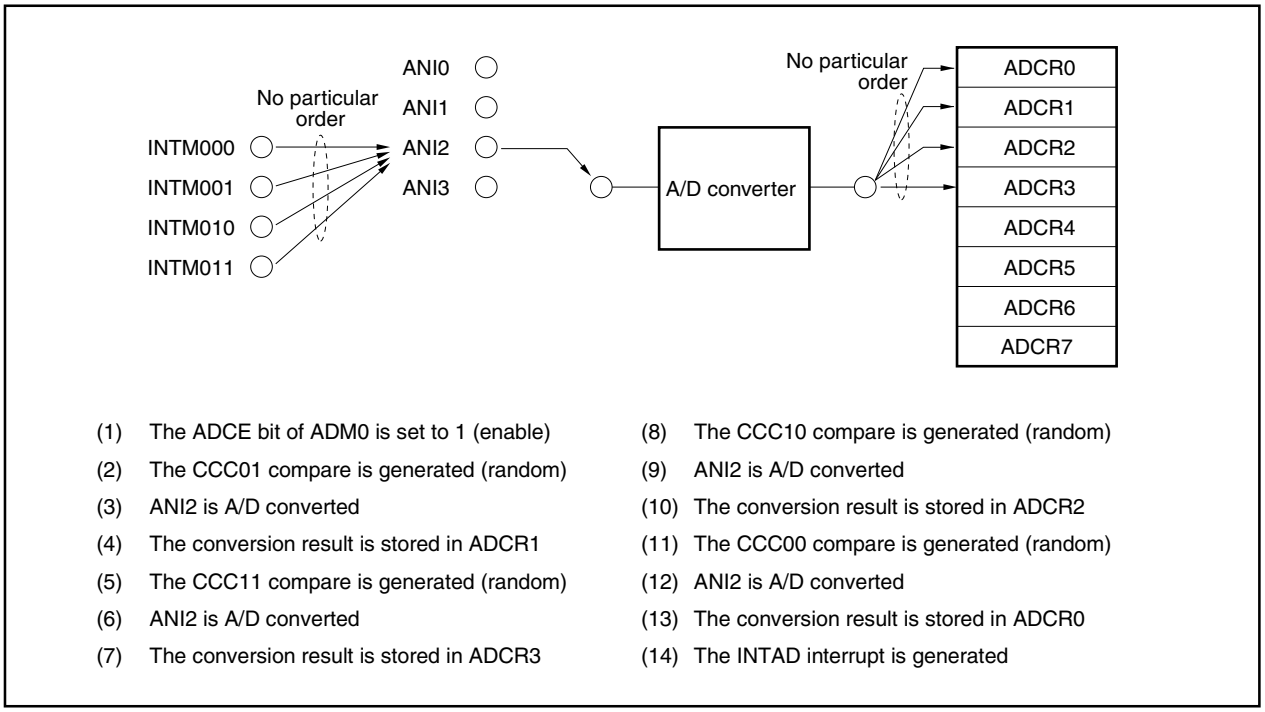
Trigger	Analog Input	A/D Conversion Result Register
INTM000 interrupt	ANIn	ADCR0
INTM001 interrupt	ANIn	ADCR1
INTM010 interrupt	ANIn	ADCR2
INTM011 interrupt	ANIn	ADCR3

In 1-shot mode, A/D conversion stops after four conversions. To restart the A/D conversion, set the TMCCEn bit of the TMCCn0 register to 1 to restart TMCn. When the first match interrupt after TMCn is restarted is generated, the ADCS bit is set (1) and A/D conversion is started (n = 0, 1).

When set to the loop mode, unless the ADCE bit of the ADM0 register is set to 0, A/D conversion is repeated each time a match interrupt is generated.

The match interrupts (INTM000, INTM001, INTM010, INTM011) can be generated in any order, and the conversion results are stored in the ADCRn register corresponding to the input trigger. Also, even in cases where the same trigger is input continuously, it is received as a trigger.

Figure 12-12. Example of 4-Trigger Mode Operation (Timer Trigger Select: 4 Buffers 4 Triggers)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The CCC01 compare is generated (random)
(3) ANI2 is A/D converted
(4) The conversion result is stored in ADCR1
(5) The CCC11 compare is generated (random)
(6) ANI2 is A/D converted
(7) The conversion result is stored in ADCR3
(8) The CCC10 compare is generated (random)
(9) ANI2 is A/D converted
(10) The conversion result is stored in ADCR2
(11) The CCC00 compare is generated (random)
(12) ANI2 is A/D converted
(13) The conversion result is stored in ADCR0
(14) The INTAD interrupt is generated

12.6.2 Scan mode operation

In this mode, the analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin and are A/D converted the specified number of times using the match interrupt signal as a trigger.

In the conversion operation, first the analog input lower channels (ANI0 to ANI3) are A/D converted the specified number of times. If the lower channels (ANI0 to ANI3) of the analog input are set by the ADM0 register so that they are scanned, and when the set number of A/D conversions ends, the A/D conversion end interrupt (INTAD) is generated and A/D conversion is stopped.

When the higher channels (ANI4 to ANI7) of the analog input are set by the ADM0 register so that they are scanned, after the conversion of the lower channel is ended, the mode is shifted to the A/D trigger mode, and the remaining A/D conversions are executed.

The conversion results are stored in the ADCRn register corresponding to the analog input. When conversion of all the specified analog inputs has ended, the A/D conversion end interrupt (INTAD) is generated and A/D conversion is stopped (n = 0 to 7).

There are two scan modes: 1-trigger mode and 4-trigger mode, according to the number of triggers.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

(1) 1-trigger mode (timer trigger scan: 1 trigger)

In this mode, analog inputs are A/D converted the specified number of times using the match interrupt signal (INTM000) as a trigger. The analog input and ADCRn register correspond one to one.

When all the specified A/D conversions have ended, the A/D conversion end interrupt (INTAD) is generated and A/D conversion is stopped.

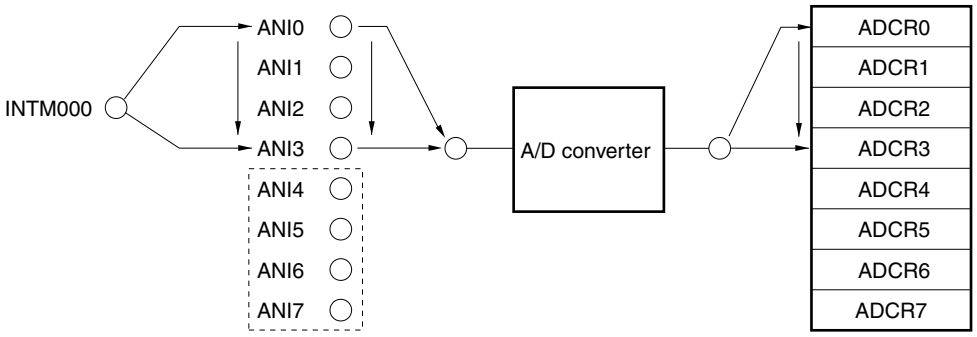
Trigger	Analog Input	A/D Conversion Result Register
INTM000 interrupt	ANI0	ADCR0
INTM000 interrupt	ANI1	ADCR1
INTM000 interrupt	ANI2	ADCR2
INTM000 interrupt	ANI3	ADCR3
(A/D trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

When the match interrupt is generated after all the specified A/D conversions have ended, A/D conversion is restarted.

In 1-shot mode, and when less than a specified number of match interrupts are generated, if the ADCEn bit is set to 1, the INTAD interrupt is not generated and the standby state is set.

Figure 12-13. Example of 1-Trigger Mode Operation (Timer Trigger Scan: 1 Trigger)

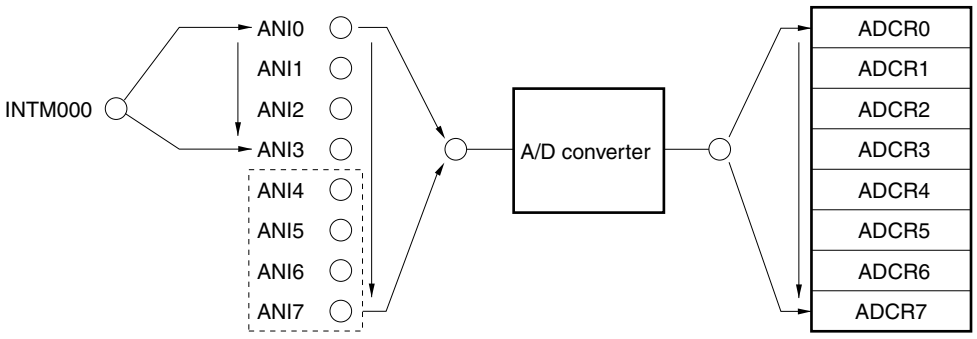
(a) Setting when scanning ANI0 to ANI3



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The CCC00 compare is generated
(3) ANI0 is A/D converted
(4) The conversion result is stored in ADCR0
(5) The CCC00 compare is generated
(6) ANI1 is A/D converted
(7) The conversion result is stored in ADCR1
(8) The CCC00 compare is generated
(9) ANI2 is A/D converted
(10) The conversion result is stored in ADCR2
(11) The CCC00 compare is generated
(12) ANI3 is A/D converted
(13) The conversion result is stored in ADCR3
(14) The INTAD interrupt is generated

Caution INTM0nn cannot be used as a trigger for the analog inputs enclosed in the broken lines (n = 0, 1). When a setting is made to scan ANI0 to ANI7, ANI4 to ANI7 are converted in A/D trigger mode (see (b) below).

(b) Setting when scanning ANI0 to ANI7



(1) to (13) Same as (a)
(14) ANI4 is A/D converted
(15) The conversion result is stored in ADCR4
(16) ANI5 is A/D converted
(17) The conversion result is stored in ADCR5
(18) ANI6 is A/D converted
(19) The conversion result is stored in ADCR6
(20) ANI7 is A/D converted
(21) The conversion result is stored in ADCR7
(22) The INTAD interrupt is generated

(2) 4-trigger mode

In this mode, analog inputs are A/D converted for the number of times specified using the match interrupt signal (INTM000, INTM001, INTM010, INTM011) as a trigger.

The analog input and ADCRn register correspond one to one.

When all the specified A/D conversions have ended, the A/D conversion end interrupt (INTAD) is generated and A/D conversion is stopped.

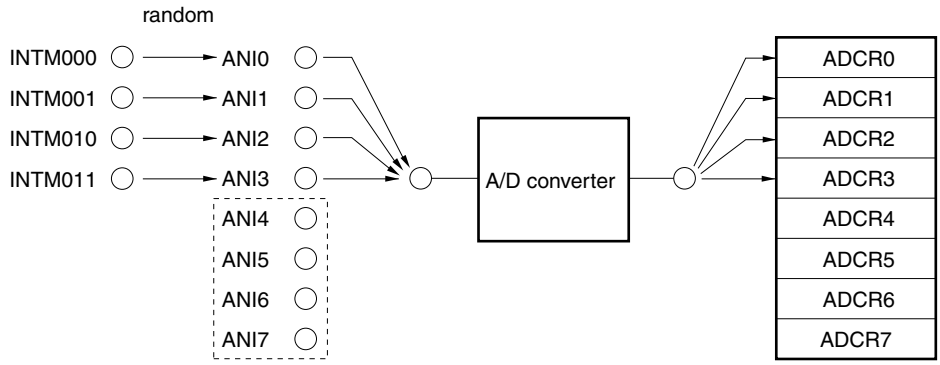
Trigger	Analog Input	A/D Conversion Result Register
INTM000 interrupt	ANI0	ADCR0
INTM001 interrupt	ANI1	ADCR1
INTM010 interrupt	ANI2	ADCR2
INTM011 interrupt	ANI3	ADCR3
(A/D trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

To restart A/D conversion in 1-shot mode, restart TMCn. If set to the loop mode and the ADCEn bit is 1, A/D conversion is restarted when a match interrupt is generated after conversion has ended.

The match interrupt can be generated in any order. However, because the trigger signal and the analog input correspond one to one, the scanning sequence is determined according to the order in which the match signals of the compare register are generated.

Figure 12-14. Example of 4-Trigger Mode Operation (Timer Trigger Scan: 4 Triggers)

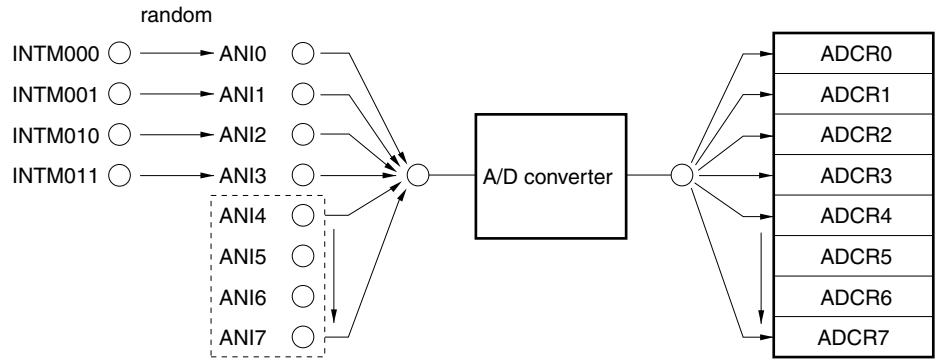
(a) Setting when scanning ANI0 to ANI3



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The CCC01 compare is generated (random)
(3) ANI1 is A/D converted
(4) The conversion result is stored in ADCR1
(5) The CCC11 compare is generated (random)
(6) ANI3 is A/D converted
(7) The conversion result is stored in ADCR3
(8) The CCC00 compare is generated (random)
(9) ANI0 is A/D converted
(10) The conversion result is stored in ADCR0
(11) The CCC10 compare is generated (random)
(12) ANI2 is A/D converted
(13) The conversion result is stored in ADCR2
(14) The INTAD interrupt is generated

Caution INTM0nn cannot be used as a trigger for the analog inputs enclosed in the broken lines TM0nn (n = 0, 1). When a setting is made to scan ANI0 to ANI7, ANI4 to ANI7 are converted in A/D trigger mode (see (b) below).

(b) Setting when scanning ANI0 to ANI7



(1) to (13) Same as (a)
(14) ANI4 is A/D converted
(15) The conversion result is stored in ADCR4
(16) ANI5 is A/D converted
(17) The conversion result is stored in ADCR5
(18) ANI6 is A/D converted
(19) The conversion result is stored in ADCR6
(20) ANI7 is A/D converted
(21) The conversion result is stored in ADCR7
(22) The INTAD interrupt is generated

12.7 Operation in 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 has an alternate function as the P37 and $\overline{\text{INTP123}}$ pins. To set the external trigger mode, set the PMC37 bit of the PMC3 register to 1 and bits TRG2 to TRG0 of the ADM1 register to 110.

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 bits ES1231 and ES1230 of the INTM3 register. For details, see **7.3.9 (1) External interrupt mode registers 1 to 4 (INTM1 to INTM4)**.

12.7.1 Select mode operations (external trigger select)

In this mode, one analog input (ANI0 to ANI3) specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input. There are two select modes: 1-buffer mode and 4-buffer mode, according to the storing method of the A/D conversion results (n = 0 to 3).

(1) 1-buffer mode (external trigger select: 1-buffer)

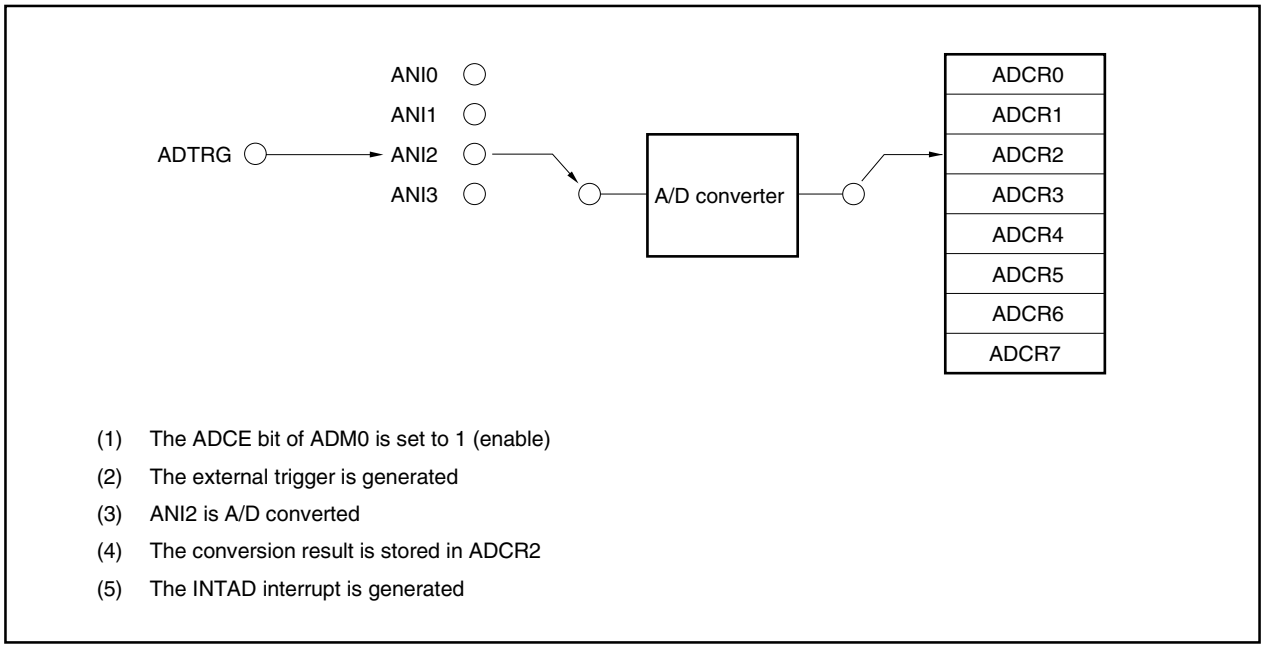
In this mode, one analog input is A/D converted using the ADTRG signal as a trigger. The conversion results are stored in one ADCRn register. The analog input and the A/D conversion results register correspond one to one. The A/D conversion end interrupt (INTAD) is generated for each A/D conversion, and A/D conversion is stopped.

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANIn	ADCRn

While the ADCE 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 most appropriate for applications in which the results are read after each A/D conversion.

Figure 12-15. Example of 1-Buffer Mode Operation (External Trigger Select: 1 Buffer)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The external trigger is generated
(3) ANI2 is A/D converted
(4) The conversion result is stored in ADCR2
(5) The INTAD interrupt is generated

(2) 4-buffer mode (external trigger select: 4 buffers)

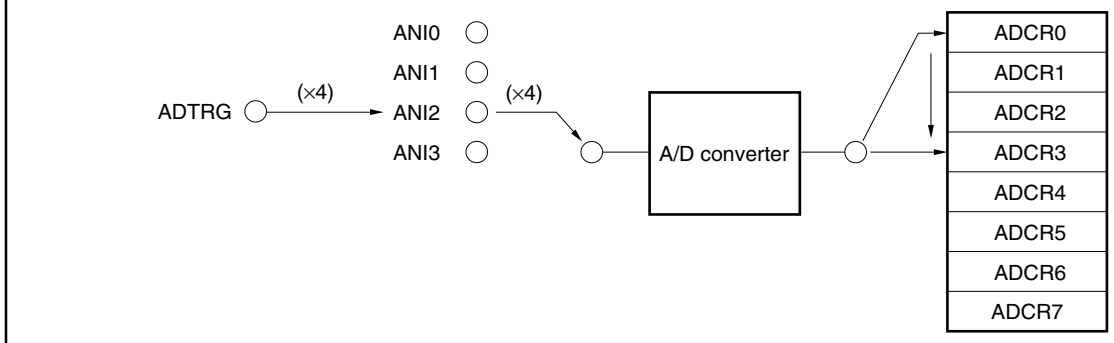
In this mode, one analog input is A/D converted four times using the ADTRG signal as a trigger and the results are stored in the ADCR0 to ADCR3 registers. The A/D conversion end interrupt (INTAD) is generated and A/D conversion is stopped after the 4th A/D conversion.

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANIn	ADCR0
ADTRG signal	ANIn	ADCR1
ADTRG signal	ANIn	ADCR2
ADTRG signal	ANIn	ADCR3

While the ADCE 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 in which calculate the average of A/D conversion result is calculated.

Figure 12-16. Example of 4-Buffer Mode Operation (External Trigger Select: 4 Buffers)



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The external trigger is generated
(3) ANI2 is A/D converted
(4) The conversion result is stored in ADCR0
(5) The external trigger is generated
(6) ANI2 is A/D converted
(7) The conversion result is stored in ADCR1
(8) The external trigger is generated
(9) ANI2 is A/D converted
(10) The conversion result is stored in ADCR2
(11) The external trigger is generated
(12) ANI2 is A/D converted
(13) The conversion result is stored in ADCR3
(14) The INTAD interrupt is generated

12.7.2 Scan mode operation (external trigger scan)

In this mode, the analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin using the ADTRG signal as a trigger, and A/D converted. The A/D conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7).

When the lower 4 channels (ANI0 to ANI3) of the analog input are set by the ADM0 register so that they are scanned, the A/D conversion end interrupt (INTAD) is generated when the number of A/D conversions specified have ended, and A/D conversion is stopped.

When the higher 4 channels (ANI4 to ANI7) of the analog input are set by the ADM0 register so that they are scanned, after the conversion of the lower 4 channels is ended, the mode is shifted to the A/D trigger mode, and the remaining A/D conversions are executed. The conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7).

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANI0	ADCR0
ADTRG signal	ANI1	ADCR1
ADTRG signal	ANI2	ADCR2
ADTRG signal	ANI3	ADCR3
(A/D trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

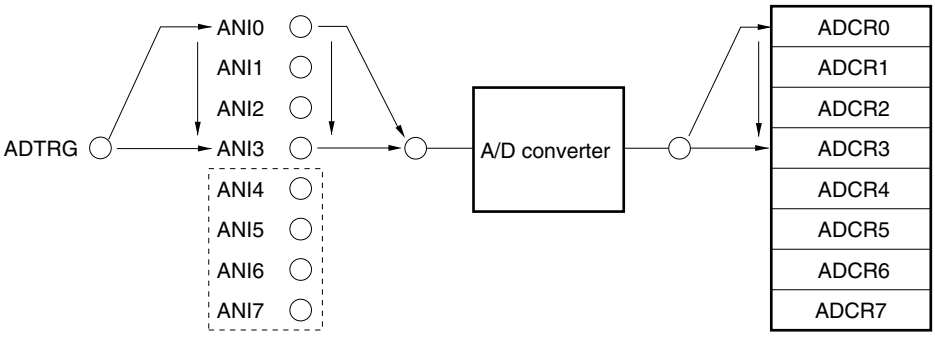
When the conversion of all the specified analog inputs has ended, the INTAD interrupt is generated and A/D conversion is stopped.

When a trigger is input to the ADTRG pin while the ADCE bit of the ADM0 register is 1, A/D conversion is started again.

This is most appropriate for applications in which multiple analog inputs are constantly monitored.

Figure 12-17. Example of Scan Mode Operation (External Trigger Scan)

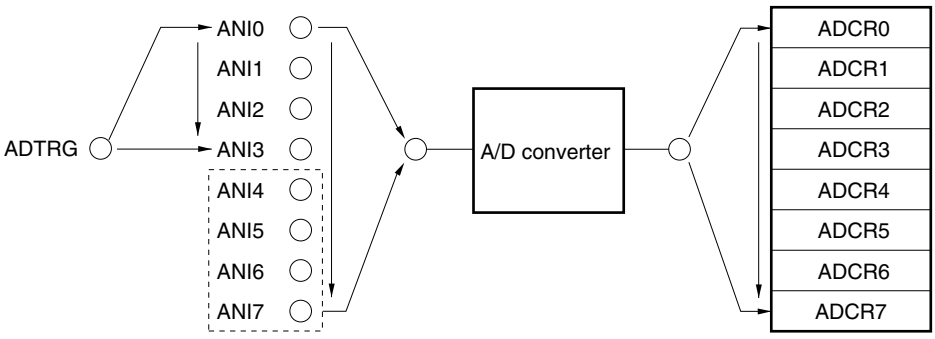
(a) Setting when scanning ANI0 to ANI3



(1) The ADCE bit of ADM0 is set to 1 (enable)
(2) The external trigger is generated
(3) ANI0 is A/D converted
(4) The conversion result is stored in ADCR0
(5) The external trigger is generated
(6) ANI1 is A/D converted
(7) The conversion result is stored in ADCR1
(8) The external trigger is generated
(9) ANI2 is A/D converted
(10) The conversion result is stored in ADCR2
(11) The external trigger is generated
(12) ANI3 is A/D converted
(13) The conversion result is stored in ADCR3
(14) The INTAD interrupt is generated

Caution ADTRG cannot be used as a trigger for the analog inputs enclosed in the broken lines. When a setting is made to scan ANI0 to ANI7, ANI4 to ANI7 are converted in A/D trigger mode (see (b) below).

(b) Setting when scanning ANI0 to ANI7



(1) to (13) Same as (a)
(14) ANI4 is A/D converted
(15) The conversion result is stored in ADCR4
(16) ANI5 is A/D converted
(17) The conversion result is stored in ADCR5
(18) ANI6 is A/D converted
(19) The conversion result is stored in ADCR6
(20) ANI7 is A/D converted
(21) The conversion result is stored in ADCR7
(22) The INTAD interrupt is generated

12.8 Notes on Operation

12.8.1 Stopping conversion operation

When the ADCE bit of the ADM0 register is set to 0 during a conversion operation, the conversion operation stops and the conversion results are not stored in the ADCRn register (n = 0 to 7).

12.8.2 External/timer trigger interval

Set the interval (input time interval) of the trigger in the external or timer trigger mode longer than the conversion time specified by the FR2 to FR0 bits of the ADM1 register.

(1) When interval = 0

When several triggers are input simultaneously, the analog input with the smaller ANIn pin number is converted. The other trigger signals input simultaneously are ignored, and the number of trigger input is not counted. Note, therefore, that the saving of the result to the ADCRn register upon the generation of an interrupt is an abnormality (n = 0 to 7).

(2) When 0 < interval < conversion operation time

When the timer trigger is input during a conversion operation, the conversion operation is aborted and the conversion starts according to the last timer trigger input.

When conversion operations are aborted, the conversion results are not stored in the ADCRn register, and the number of trigger input are not counted. Note, therefore, that the saving of the result to the ADCRn register upon the generation of an interrupt is an abnormality (n = 0 to 7).

(3) When interval = conversion operation time

When a trigger is input concurrently with the end of conversion (the end of conversion signal and the trigger are in contention), although the number of triggers input are counted, an interrupt is generated, and the value at the end of conversion is correctly saved in the ADCRn register, design should be performed so that the interval is greater than the conversion operation time.

12.8.3 Operation in standby mode

(1) HALT mode

In this mode, A/D conversion continues. When this mode is released by NMI input, the ADM0 and ADM1 registers 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 these modes are released by NMI input or maskable interrupt input (INTP1xx), the ADM0 and ADM1 registers and the ADCRn register hold the value. However, when the IDLE or software STOP mode is set during a conversion operation, the conversion operation is stopped. At this time, if the mode released by NMI input or maskable interrupt input (INTP1xx), the conversion operation resumes, but the conversion result written to the ADCRn register will become undefined (x = 0 to 3, n = 0 to 7).

12.8.4 Compare match interrupt in timer trigger mode

The compare register's match interrupt becomes an A/D conversion start trigger and starts the conversion operation. When this happens, the compare register's match interrupt also functions as a compare register match interrupt for the CPU. In order to prevent match interrupts from the compare register for the CPU, disable interrupts using the mask bits (P00MK0, P00MK1, P01MK0, P01MK1) of the interrupt control register (P00IC0, P00IC1, P01IC0, P01IC1).

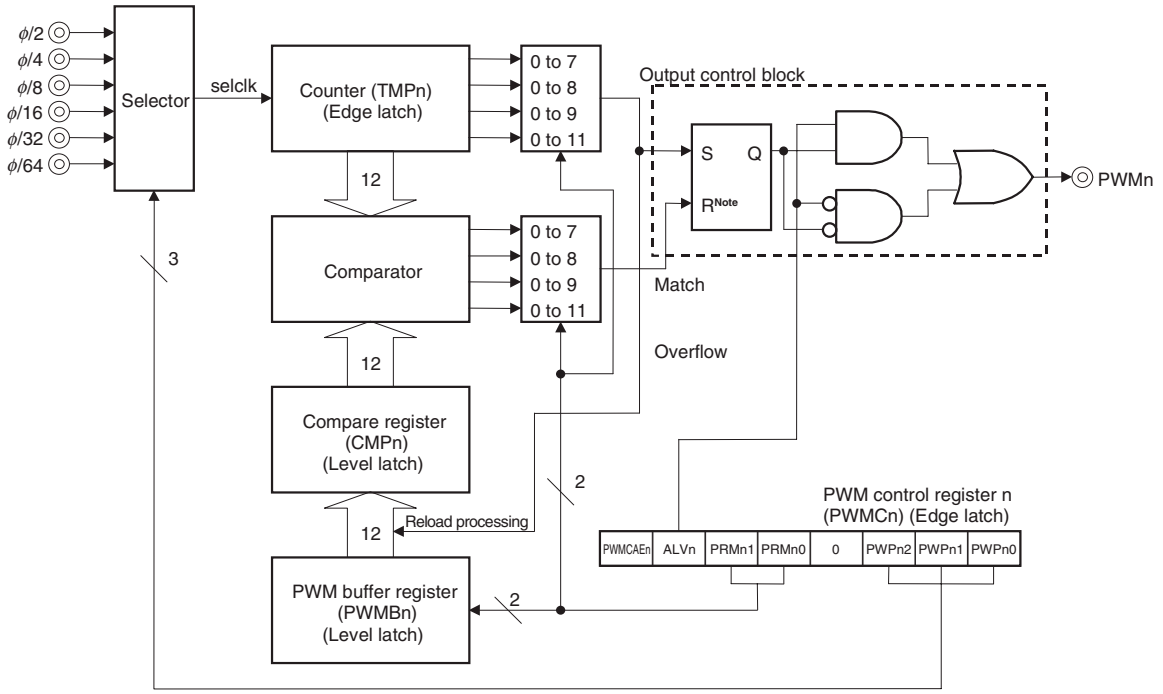
CHAPTER 13 PWM UNIT

13.1 Features

- PWMn: 2 channels
- PWMn: Output pulse active level can be selected
- Operation clock can be selected from among ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64 (ϕ is the internal system clock)
- PWMn output resolution can be selected from among 8, 9, 10, 12 bits

Remark n = 0, 1

13.2 Block Diagram



Note Reset has priority.

Remark n = 0, 1

13.3 Control Register

(1) PWM control registers 0, 1 (PWMC0, PWMC1)

The PWMCn register is used to control the PWMn's operations (n = 0, 1).

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

Caution When PWMn is used, be sure to set external pins related to PWMn to control mode. Following that, set the operation clock, etc. using the PWMCn register and set the PWMEn bit to 1 after the PWMBn register setting is made.

	<7>	<6>	5	4	3	2	1	0	Address	After Reset
PWMCn	PWMEn	ALVn	PRMn1	PRMn0	0	PWPn2	PWPn1	PWPn0	FFFFFC00H, FFFFFC10H	40H

Bit position	Bit name	Description
7	PWMEn[Note] (n = 0, 1)	PWM Enable This bit is used to enable or disable PWMn operation. 0: PWM operation disabled 1: PWM operation enabled
6	ALVn (n = 0, 1)	Active Level This bit is used to specify the active level for PWMn output. 0: Active level is low level 1: Active level is high level The PWMn outputs inactive level (low level) of the ALVn bit after reset.
5, 4	PRMn1, PRMn0 (n = 0, 1)	Prescaler Mode This bit is used to select the bit length for the counter (TMPn) and compare register (CMPn).

PRMn1	PRMn0	Bit length for TMPn and CMPn
0	0	8 bits
0	1	9 bits
1	0	10 bits
1	1	12 bits

2 to 0	PWPn2 to PWPn0 (n = 0, 1)	PWM Prescaler Clock Mode This bit is used to select the PWMn's operating clock.

PWPn2	PWPn1	PWPn0	Operating clock
0	0	0	ϕ/2
0	0	1	ϕ/4
0	1	0	ϕ/8
0	1	1	ϕ/16
1	0	0	ϕ/32
1	0	1	ϕ/64
Other than above			Setting prohibited

Note If PWMEn is changed from 0 to 1, the counter (TMPn) is reset to start counting from 000H (in 12 bits). The first overflow permits the PWMn signal activation. If the bit length and operating clock of PWM0 and PWM1 are the same, the activation timing of these two PWMn signals can be adjusted. If PWMEn was already 1, the counter is not reset upon an additional write of 1. When setting PWMEn to 1, set it to 0 beforehand.

Remark n = 0, 1

(2) PWM buffer registers 0, 1 (PWMB0, PWMB1)

The PWMBn register is a 12-bit buffer register that is used to set control data for the active signal width of PWMn output. Bits 15 to 12 are fixed to zero. Even if 1 is written in these bits, it is ignored. It is possible to directly read the values of bits 11 to 8 as written irrespective of the bit length setting made by the PWMCn register.

The contents in PWMBn registers are transferred to compare registers (CMPn) at the timing of the generation of an overflow from the PWMn output control counter (TMPn).

The PWM buffer registers can be read or written in 16-bit units.

Remark n = 0, 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
PWMB0	0	0	0	0	PWM11	PWM10	PWM9	PWM8	PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0	FFFFFC02H	0000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
PWMB1	0	0	0	0	PWM11	PWM10	PWM9	PWM8	PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0	FFFFFC12H	0000H

13.4 Operation

13.4.1 Basic operations

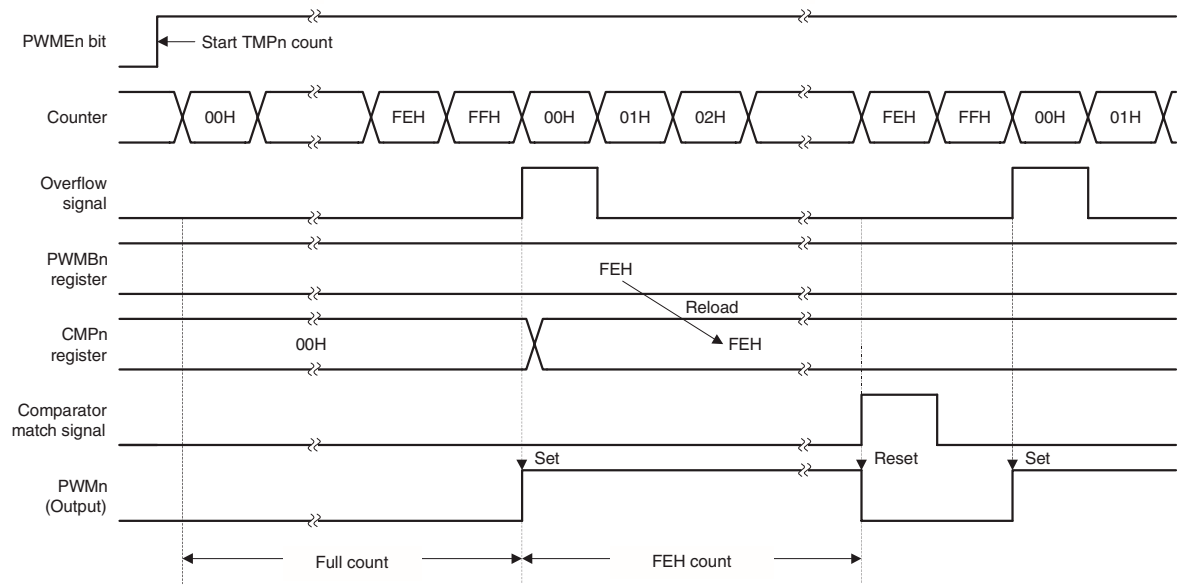
When a PWMn pulse is output, the required data is first set to the PWMCn and PWMBn registers, then the PWMCn register's PWMEn bit is set (1). This clears (0) the counter (TMPn) and, when the first overflow occurs, the active level is set for PWMn output and the data is transferred from the PWMBn register to the compare register (CMPn). Afterward, PWMn output goes inactive when a match occurs between the TMPn and CMPn register values. When this is repeated, a PWMn signal whose active level is specified by the ALVn bit in the PWMCn register is output from the PWMn pin.

When the PWMCn register's PWMEn bit is cleared (0), PWMn output is stopped immediately and is set to the inactive level for the ALVn level specified by the PWMCn register.

If, during PWMn signal output, the values of the PWPn2 to PWPn0 bits, PRMn1 and PRMn0 bits, or ALVn bit are changed, the cycle width and pulse width of the PWMn signal are not guaranteed within the cycle where the changes were made.

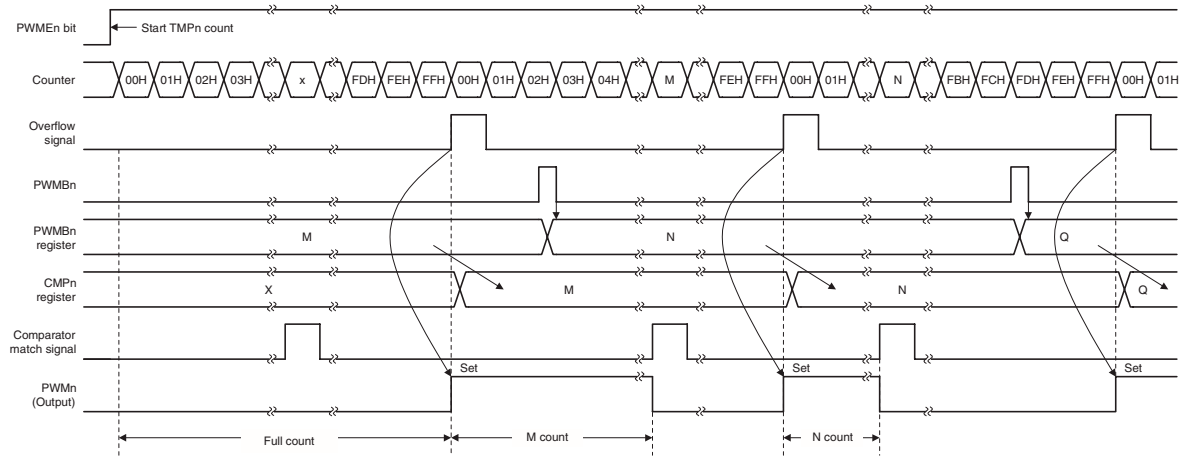
Remark n = 0, 1

Figure 13-1. PWM Basic Operation Timing



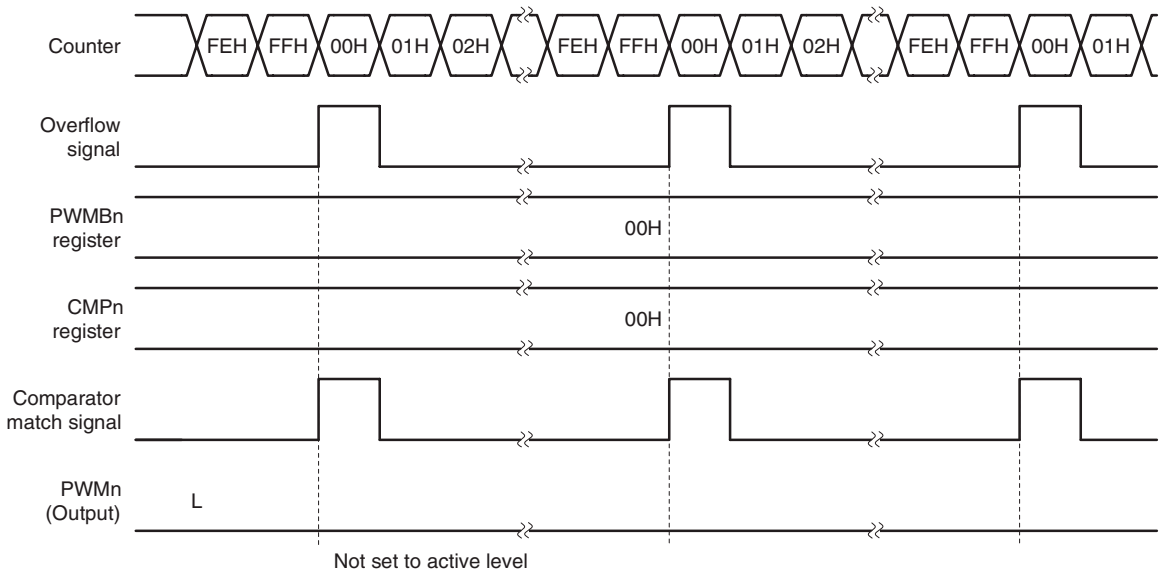
Remark n = 0, 1

Figure 13-2. Timing for Write Operation to PWMBn Register



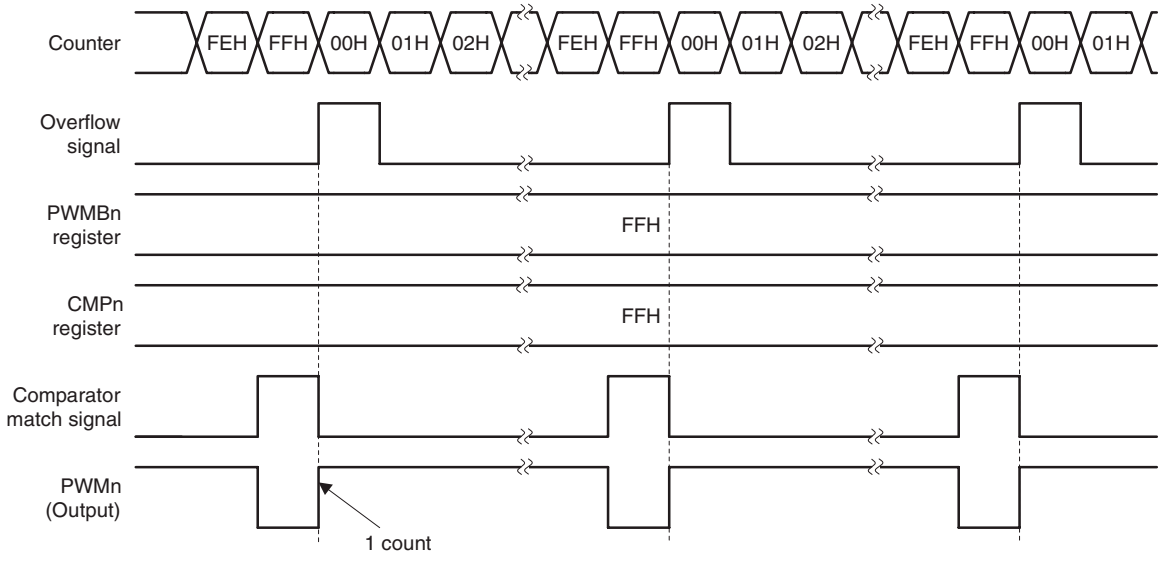
Remark n = 0, 1

Figure 13-3. Timing When PWMBn Register Is Set to 00H



Remark n = 0, 1

Figure 13-4. Timing When PWMBn Register Is Set to FFH



Remark n = 0, 1

13.4.2 Repetition frequency

The repetition frequencies of PWMn are shown below (n = 0, 1).

PWMn Operating Frequency	Resolution	Repetition Frequency
ϕ/2	8 bits	$\phi/2^{9}$
	9 bits	$\phi/2^{10}$
	10 bits	$\phi/2^{11}$
	12 bits	$\phi/2^{13}$
ϕ/4	8 bits	$\phi/2^{10}$
	9 bits	$\phi/2^{11}$
	10 bits	$\phi/2^{12}$
	12 bits	$\phi/2^{14}$
ϕ/8	8 bits	$\phi/2^{11}$
	9 bits	$\phi/2^{12}$
	10 bits	$\phi/2^{13}$
	12 bits	$\phi/2^{15}$
ϕ/16	8 bits	$\phi/2^{12}$
	9 bits	$\phi/2^{13}$
	10 bits	$\phi/2^{14}$
	12 bits	$\phi/2^{16}$
ϕ/32	8 bits	$\phi/2^{13}$
	9 bits	$\phi/2^{14}$
	10 bits	$\phi/2^{15}$
	12 bits	$\phi/2^{17}$
ϕ/64	8 bits	$\phi/2^{14}$
	9 bits	$\phi/2^{15}$
	10 bits	$\phi/2^{16}$
	12 bits	$\phi/2^{18}$

13.5 Cautions

The PWM0 pin has an alternate function as the P00 pin (Port 0) and the PWM1 pin has an alternate function as the P10 pin (Port 1). When using these pins for PWMn output, set the bits corresponding to the PMC0 and PMC1 registers to 1.

If the bit settings corresponding to the PMC0 and PMC1 registers are changed during PWMn pulse output, the PWMn pulse output is not guaranteed.

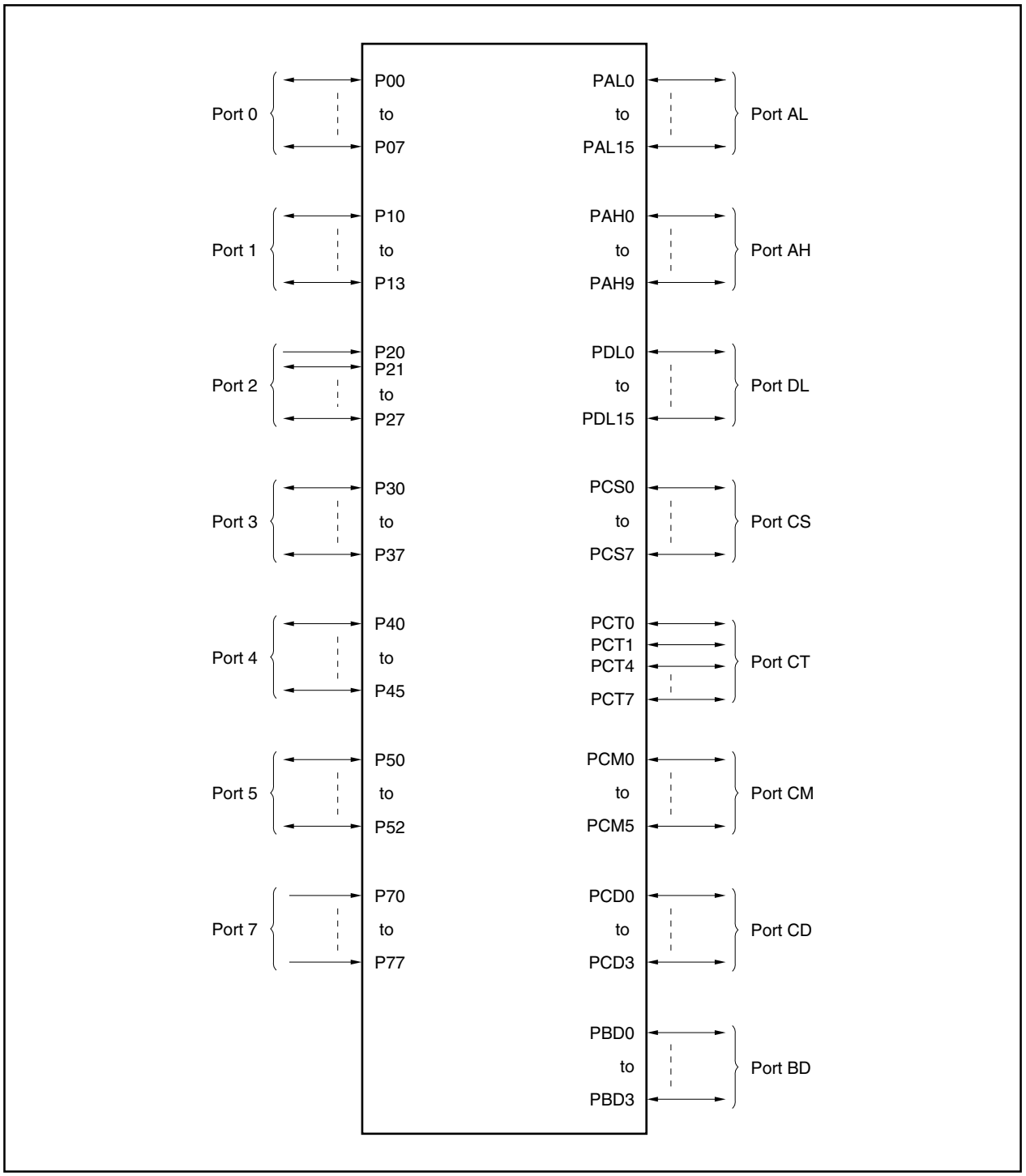
CHAPTER 14 PORT FUNCTIONS

14.1 Features

- Input-only ports: 9
  Input/output ports: 106
- Function alternately as other peripheral I/O pins.
- It is possible to specify input and output in 1-bit units.

14.2 Port Configuration

The V850E/MA1 incorporates a total of 115 input/output ports (including 9 input-only ports) labeled ports 0 through 5, and AL, AH, DL, CS, CT, CM, CD, and BD. The port configuration is shown below.



(1) Function of each port

The port functions of this product are shown below.

8-bit and 1-bit operations are possible on all ports, allowing various kinds of control to be performed. In addition to their port functions, these pins also function as internal peripheral I/O input/output pins in the control mode. For the block types of each port, see (3) Block diagram of port.

Port Name	Pin Name	Port Function	Function in Control Mode	Block Type
Port 0	P00 to P07	8-bit I/O	Real-time pulse unit (RPU) I/O External interrupt input PWM output DMA controller input	A, B, H
Port 1	P10 to P13	4-bit I/O	Real-time pulse unit (RPU) I/O External interrupt input PWM output	A, B
Port 2	P20 to P27	1-bit input, 7-bit I/O	NMI input Real-time pulse unit (RPU) I/O External interrupt input DMA controller output	A, B, F, N
Port 3	P30 to P37	8-bit I/O	Serial interface I/O (CSI2, UART2) External interrupt input A/D converter external trigger input	B, H, I, N
Port 4	P40 to P45	6-bit I/O	Serial interface I/O (UART0/CSI0, UART1/CSI1)	H, L, M
Port 5	P50 to P52	3-bit I/O	Real-time pulse unit (RPU) I/O External interrupt input	A, B
Port 7	P70 to P77	8-bit input	A/D converter input	C
Port AL	PAL0 to PAL15	16-bit I/O	External address bus (A0 to A15)	J
Port AH	PAH0 to PAH9	10-bit I/O	External address bus (A16 to A25)	J
Port DL	PDL0 to PDL15	16-bit I/O	External data bus (D0 to D15)	O
Port CS	PCS0 to PCS7	8-bit I/O	External bus interface control signal output	J, K
Port CT	PCT0,PCT1, PCT4 to PCT7	6-bit I/O	External bus interface control signal output	J
Port CM	PCM0 to PCM5	6-bit I/O	Wait insertion signal input Internal system clock output/Bus clock output External bus interface control signal I/O Self-refresh request signal input	D, E, J, K
Port CD	PCD0 to PCD3	4-bit I/O	External bus interface control signal output	J, K
Port BD	PBD0 to PBD3	4-bit I/O	DMA controller output	J

Caution When switching the mode of a port that functions as an output or I/O pin to control mode, be sure to follow the procedure below.

<1> Set the inactive level of the signals output in control mode to the appropriate bits in port n (n = 0 to 5, AL, AH, DL, CS, CT, CM, CD, and BD).

<2> The mode is switched to control mode by the port n mode control register (PMCn).

If <1> above is not performed, the contents of port n may be output for a moment when the mode is switched from port mode to control mode.

(2) Function when each port's pins are reset and registers that set the port/control mode

(1/2)

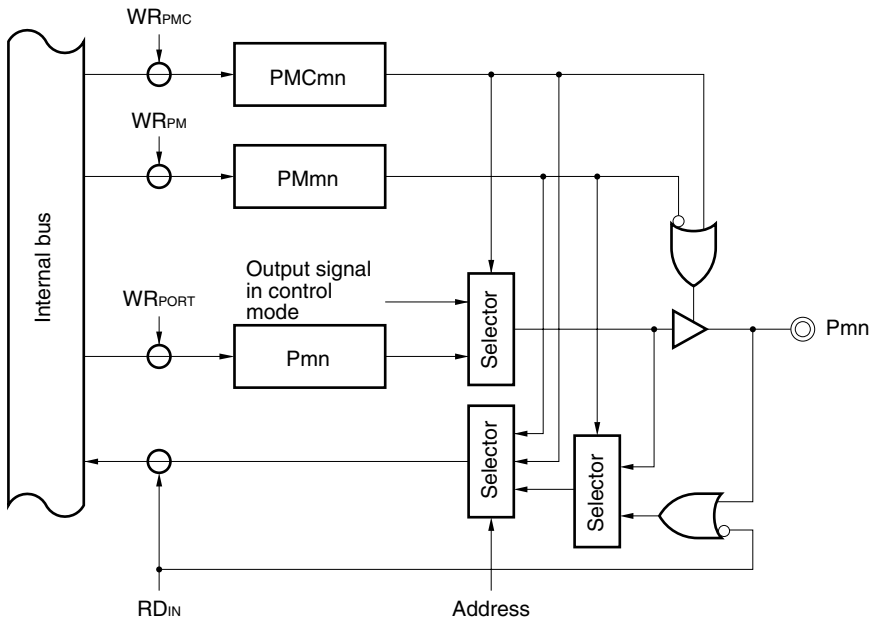
Port Name	Pin Name	Pin Function After Reset				Register That Sets the Mode
		Single-Chip Mode 0	Single-Chip Mode 1	ROMless Mode 0	ROMless Mode 1	
Port 0	P00/PWM0	P00 (input mode)				PMC0
	P01/INTP000/TI000	P01 (input mode)				
	P02/INTP001	P02 (input mode)				
	P03/TO00	P03 (input mode)				
	P04/$\overline{\text{DMARQ0}}$/$\overline{\text{INTP100}}$	P04 (input mode)				PMC0, PFC0
	P05/$\overline{\text{DMARQ1}}$/$\overline{\text{INTP101}}$	P05 (input mode)				
	P06/$\overline{\text{DMARQ2}}$/$\overline{\text{INTP102}}$	P06 (input mode)				
	P07/$\overline{\text{DMARQ3}}$/$\overline{\text{INTP103}}$	P07 (input mode)				
Port 1	P10/PWM1	P10 (input mode)				PMC1
	P11/INTP010/TI010	P11 (input mode)				
	P12/INTP011	P12 (input mode)				
	P13/TO01	P13 (input mode)				
Port 2	P20/NMI	NMI				–
	P21/INTP020/TI020	P21 (input mode)				PMC2
	P22/INTP021	P22 (input mode)				
	P23/TO02	P23 (input mode)				
	P24/$\overline{\text{TC0}}$/$\overline{\text{INTP110}}$	P24 (input mode)				PMC2, PFC2
	P25/$\overline{\text{TC1}}$/$\overline{\text{INTP111}}$	P25 (input mode)				
	P26/$\overline{\text{TC2}}$/$\overline{\text{INTP112}}$	P26 (input mode)				
	P27/$\overline{\text{TC3}}$/$\overline{\text{INTP113}}$	P27 (input mode)				
Port 3	P30/SO2/$\overline{\text{INTP130}}$	P30 (input mode)				PMC3, PFC3
	P31/SI2/$\overline{\text{INTP131}}$	P31 (input mode)				
	P32/$\overline{\text{SCK2}}$/INTP132	P32 (input mode)				
	P33/TXD2/$\overline{\text{INTP133}}$	P33 (input mode)				
	P34/RXD2/$\overline{\text{INTP120}}$	P34 (input mode)				
	P35/$\overline{\text{INTP121}}$	P35 (input mode)				PMC3
	P36/$\overline{\text{INTP122}}$	P36 (input mode)				
	P37/ADTRG/$\overline{\text{INTP123}}$	P37 (input mode)				
Port 4	P40/TXD0/SO0	P40 (input mode)				PMC4, PFC4
	P41/RXD0/SI0	P41 (input mode)				
	P42/$\overline{\text{SCK0}}$	P42 (input mode)				PMC4
	P43/TXD1/SO1	P43 (input mode)				PMC4, PFC4
	P44/RXD1/SI1	P44 (input mode)				
	P45/$\overline{\text{SCK1}}$	P45 (input mode)				PMC4

(2/2)

Port Name	Pin Name	Pin Function After Reset				Register That Sets the Mode
		Single-Chip Mode 0	Single-Chip Mode 1	ROMless Mode 0	ROMless Mode 1	
Port 5	P50/INTP030/TI030	P50 (input mode)				PMC5
	P51/INTP031	P51 (input mode)				
	P52/TO03	P52 (input mode)				
Port 7	P70/ANI0 to P77/ANI7	P70 to P77 (input mode)				–
Port BD	PBD0/$\overline{\text{DMAAK0}}$ to PBD3/$\overline{\text{DMAAK3}}$	PBD0 to PBD3 (input mode)				PMCBD
Port CM	PCM0/$\overline{\text{WAIT}}$	PCM0 (input mode)	$\overline{\text{WAIT}}$			PMCCM
	PCM1/CLKOUT/BUSCLK	PCM1 (input mode)	CLKOUT/BUSCLK			PMCCM,PFCCM
	PCM2/$\overline{\text{HLDAK}}$	PCM2 (input mode)	$\overline{\text{HLDAK}}$			PMCCM
	PCM3/$\overline{\text{HLDRQ}}$	PCM3 (input mode)	$\overline{\text{HLDRQ}}$			
	PCM4/$\overline{\text{REFRQ}}$	PCM4 (input mode)	$\overline{\text{REFRQ}}$			
	PCM5/SELFREF	PCM5 (input mode)	SELFREF			
Port CT	PCT0/$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM	PCT0 (input mode)	$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM			PMCCT
	PCT1/$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM	PCT1 (input mode)	$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM			
	PCT4/$\overline{\text{RD}}$	PCT4 (input mode)	$\overline{\text{RD}}$			
	PCT5/$\overline{\text{WE}}$	PCT5 (input mode)	$\overline{\text{WE}}$			
	PCT6/$\overline{\text{OE}}$	PCT6 (input mode)	$\overline{\text{OE}}$			
	PCT7/$\overline{\text{BCYST}}$	PCT7 (input mode)	$\overline{\text{BCYST}}$			
Port CS	PCS0/$\overline{\text{CS0}}$	PCS0 (input mode)	$\overline{\text{CS0}}$			PMCCS
	PCS1/$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$	PCS1 (input mode)	$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$			
	PCS2/$\overline{\text{CS2}}$/$\overline{\text{IOWR}}$	PCS2 (input mode)	$\overline{\text{CS2}}$/$\overline{\text{IOWR}}$			PMCCS,PFCCS
	PCS3/$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$	PCS3 (input mode)	$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$			PMCCS
	PCS4/$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$	PCS4 (input mode)	$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$			
	PCS5/$\overline{\text{CS5}}$/$\overline{\text{IORD}}$	PCS5 (input mode)	$\overline{\text{CS5}}$/$\overline{\text{IORD}}$			PMCCS,PFCCS
	PCS6/$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$	PCS6 (input mode)	$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$			PMCCS
	PCS7/$\overline{\text{CS7}}$	PCS7 (input mode)	$\overline{\text{CS7}}$			
Port CD	PCD0/SDCKE	PCD0 (input mode)	SDCKE			PMCCD
	PCD1/SDCLK	PCD1 (input mode)	SDCLK			
	PCD2/$\overline{\text{LBE}}$/$\overline{\text{SDCAS}}$	PCD2 (input mode)	$\overline{\text{LBE}}$/$\overline{\text{SDCAS}}$			PMCCD,PFCCD
	PCD3/$\overline{\text{UBE}}$/$\overline{\text{SDRAS}}$	PCD3 (input mode)	$\overline{\text{UBE}}$/$\overline{\text{SDRAS}}$			
Port AH	PAH0/A16 to PAH9/A25	PAH0 to PAH9 (input mode)	A16 to A25			PMCAH
Port AL	PAL0/A0 to PAL15/A15	PAL0 to PAL15 (input mode)	A0 to A15			PMCAL
Port DL	PDL0/D0 to PDL15/D15	PDL0 to PDL15 (input mode)	D0 to D15			PMCDL

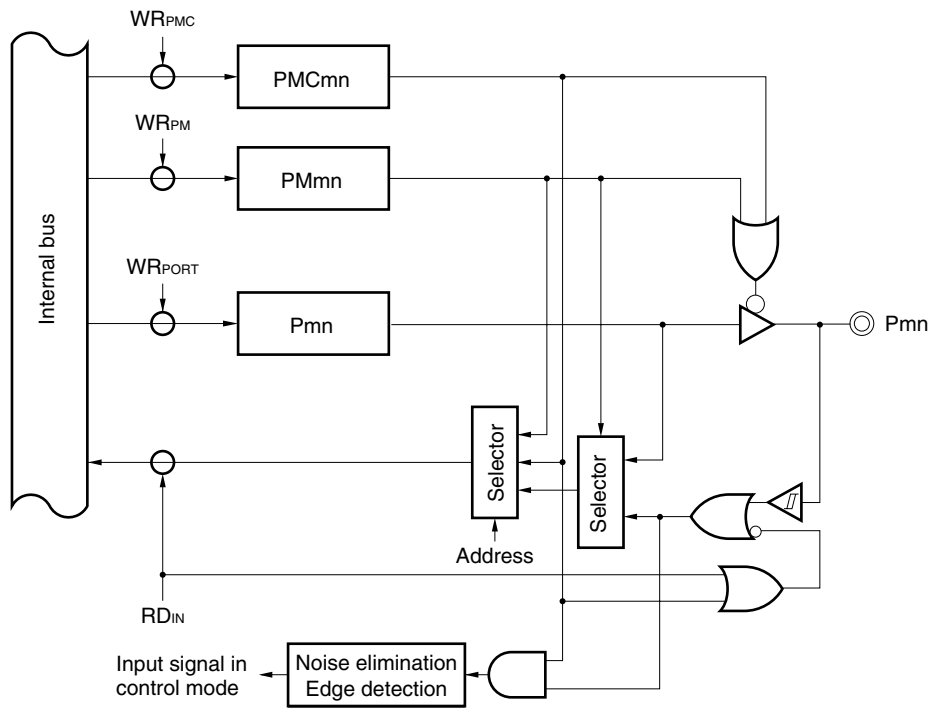
(3) Block diagram of port

Figure 14-1. Block Diagram of Type A



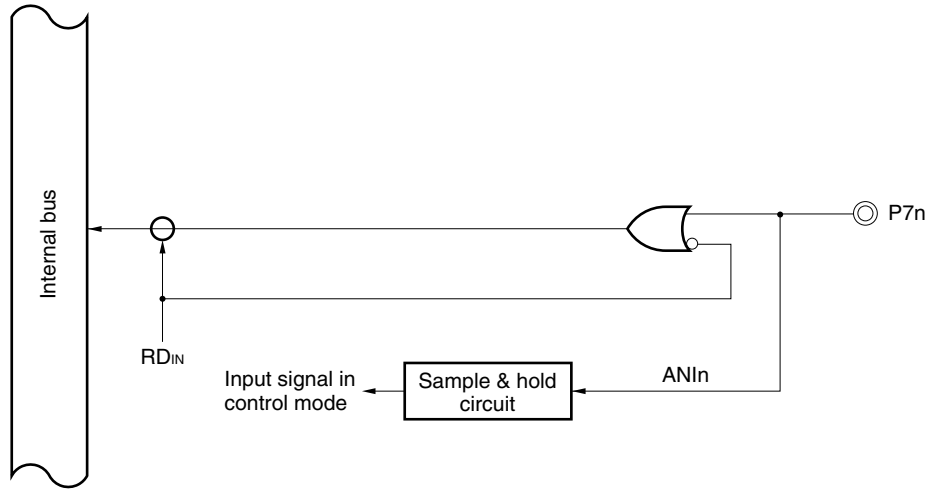
Remark m: Port number
n: Bit number

Figure 14-2. Block Diagram of Type B



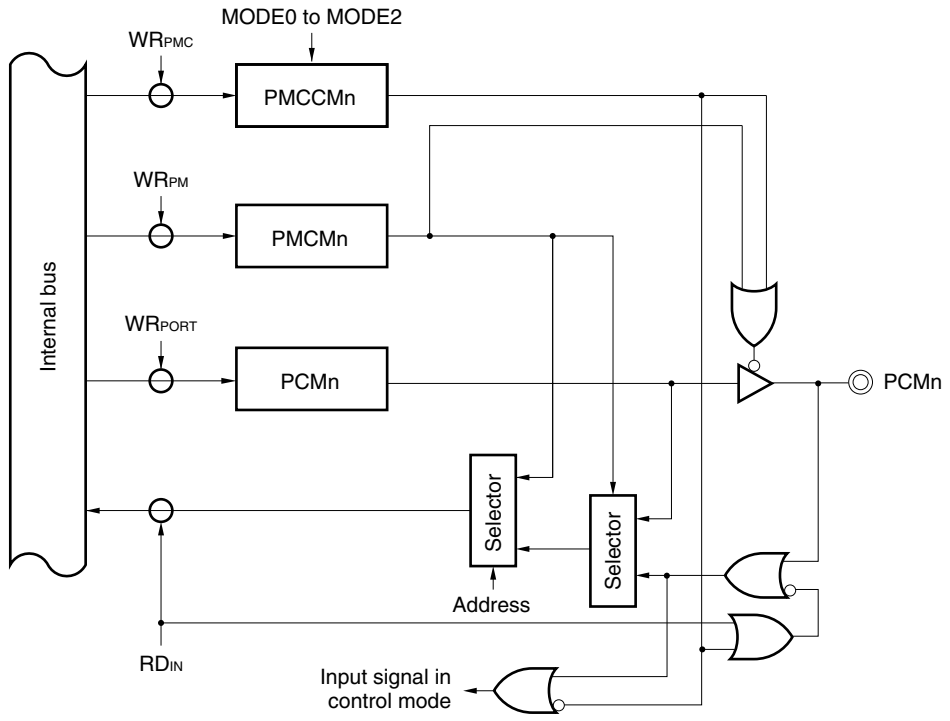
Remark m: Port number
n: Bit number

Figure 14-3. Block Diagram of Type C



Remark n = 0 to 7

Figure 14-4. Block Diagram of Type D



Remark n = 0, 3

Figure 14-5. Block Diagram of Type E

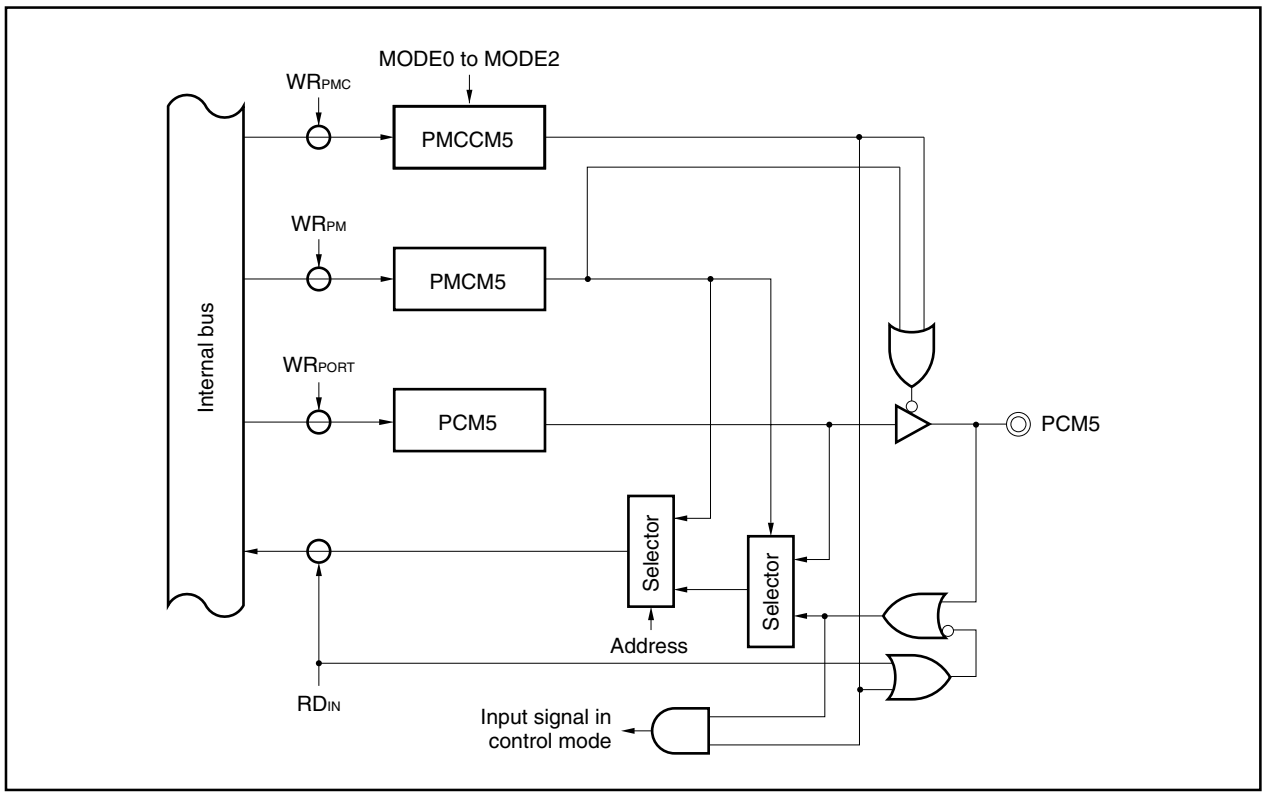


Figure 14-6. Block Diagram of Type F

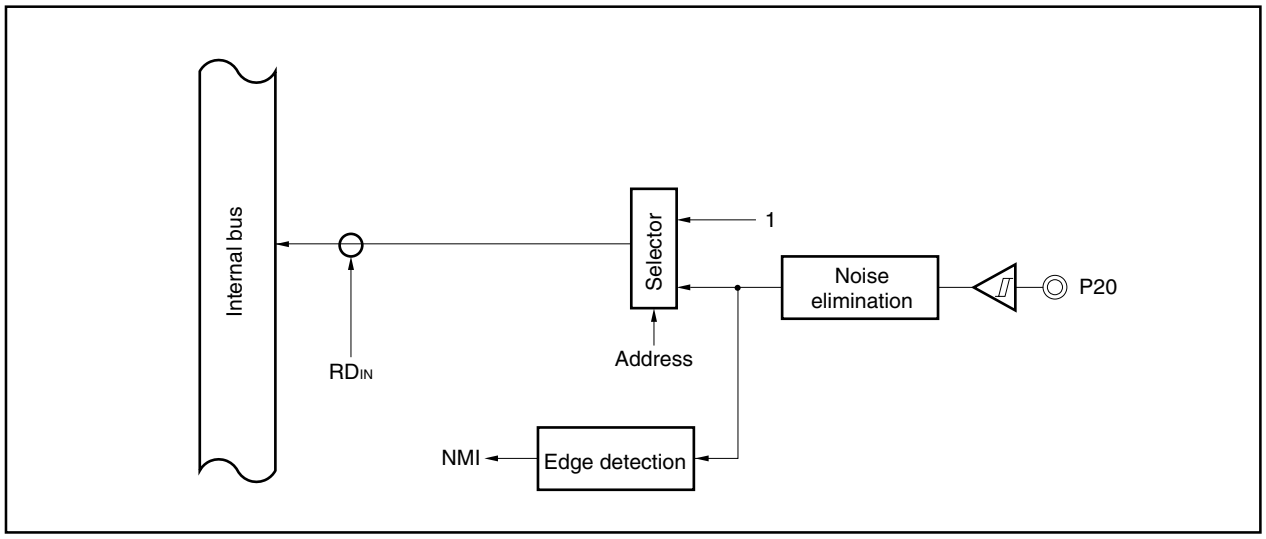
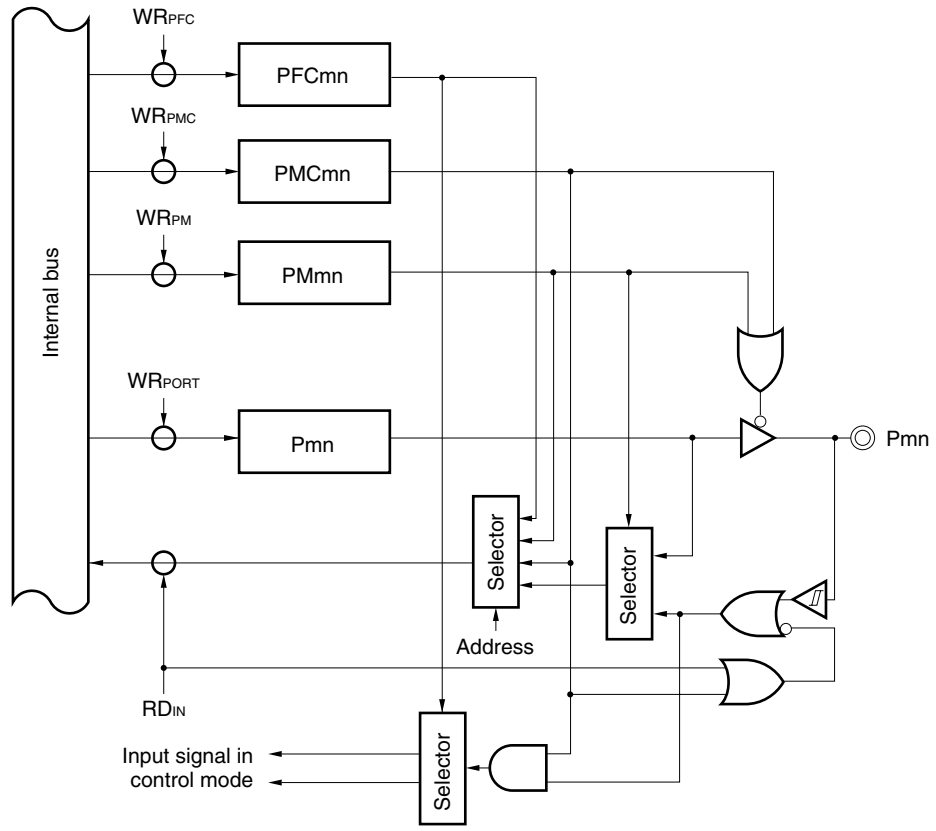


Figure 14-7. Block Diagram of Type H



Remark m: Port number
n: Bit number

Figure 14-8. Block Diagram of Type I

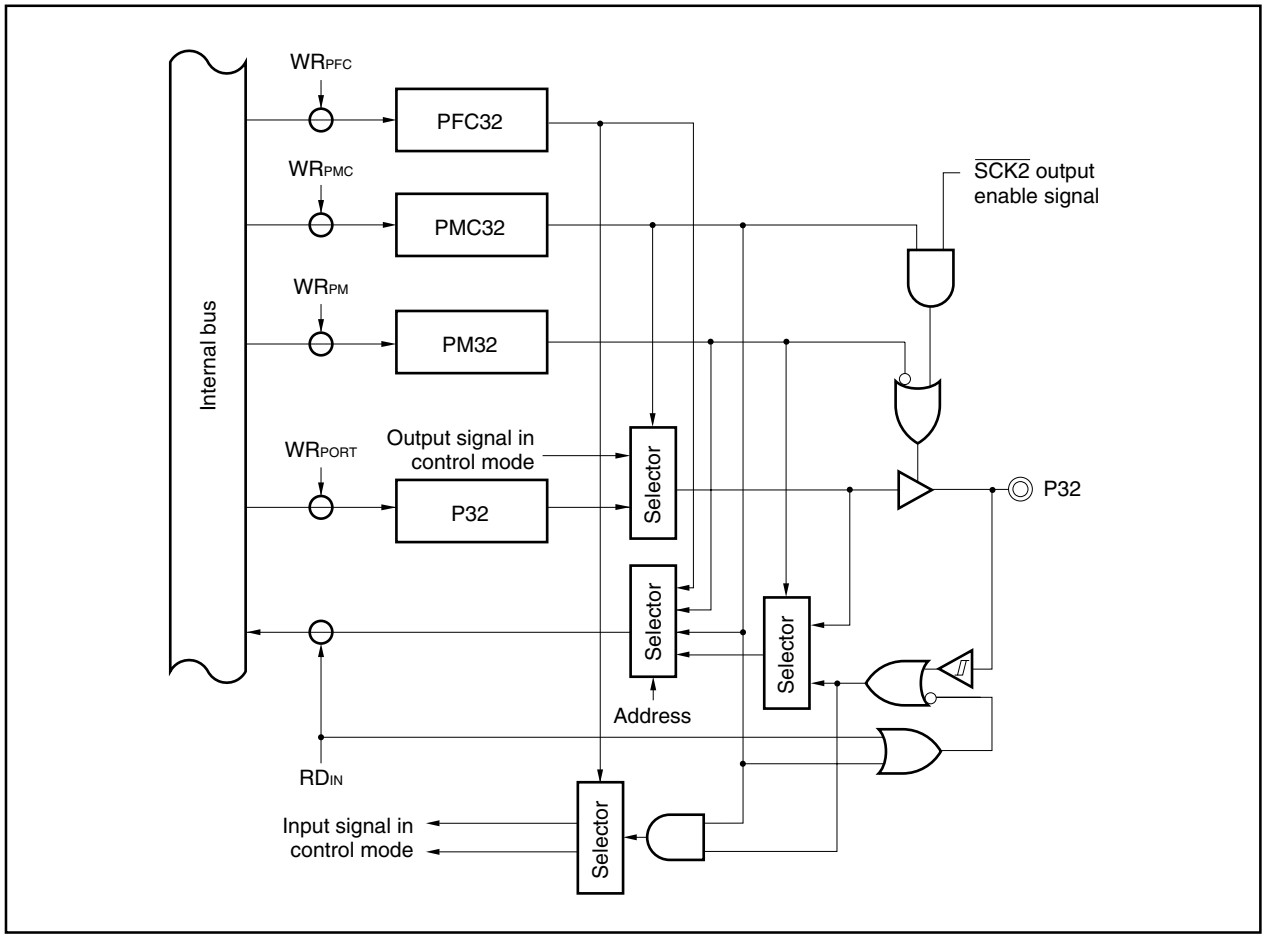
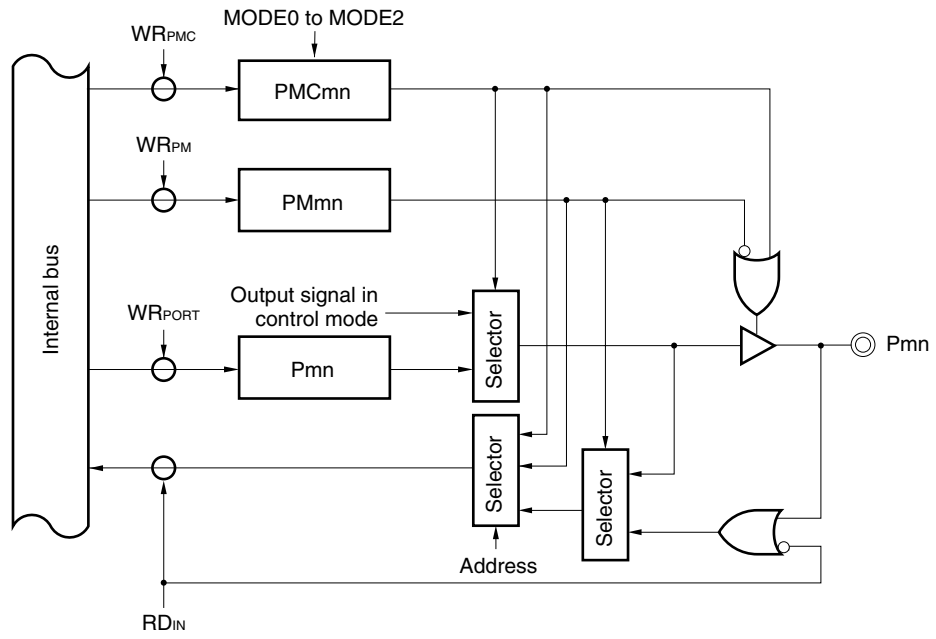
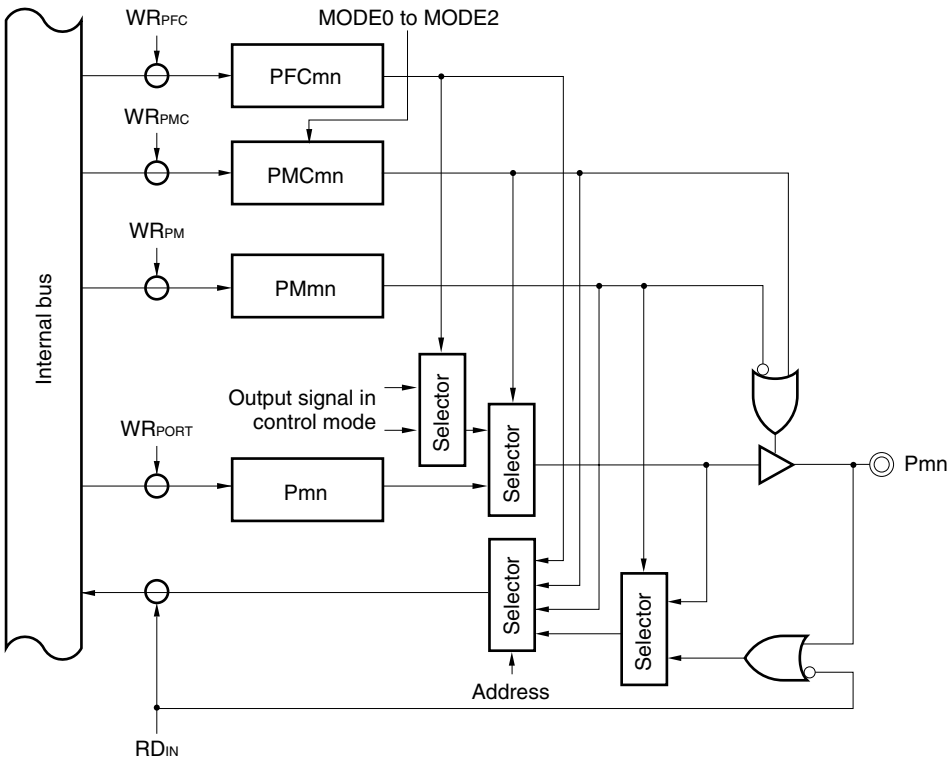


Figure 14-9. Block Diagram of Type J



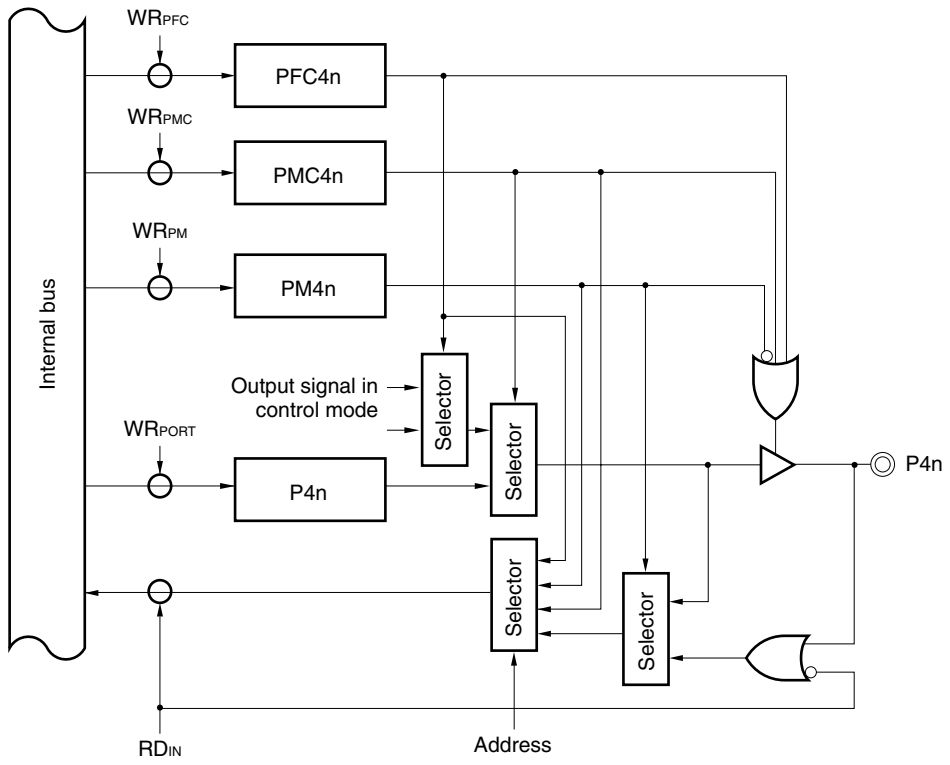
Remark m: Port number
n: Bit number

★ **Figure 14-10. Block Diagram of Type K**



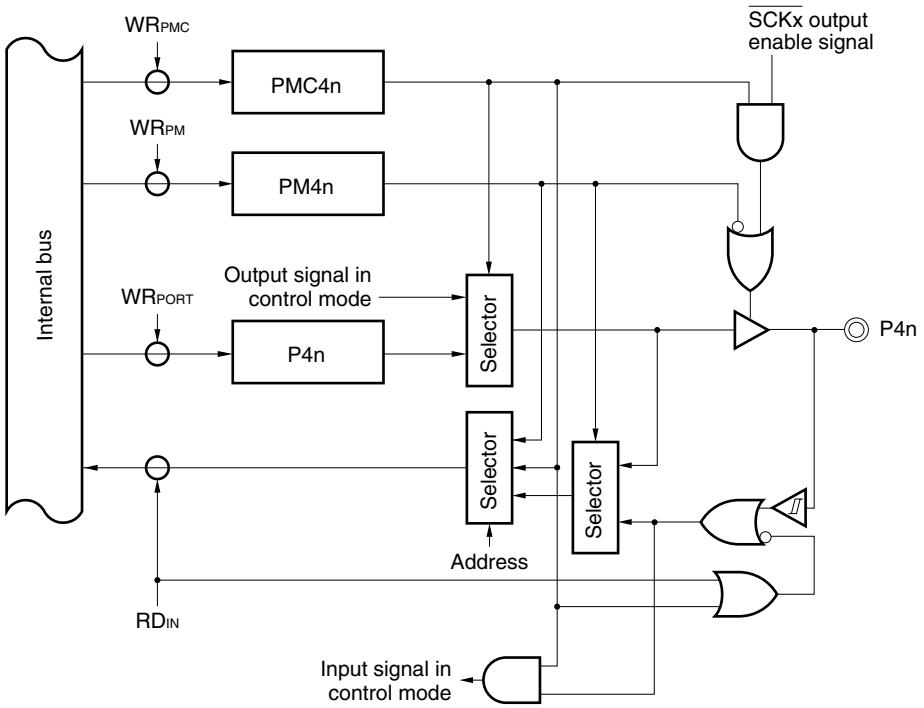
Remark m: Port number
n: Bit number

Figure 14-11. Block Diagram of Type L



Remark n = 0, 3

Figure 14-12. Block Diagram of Type M

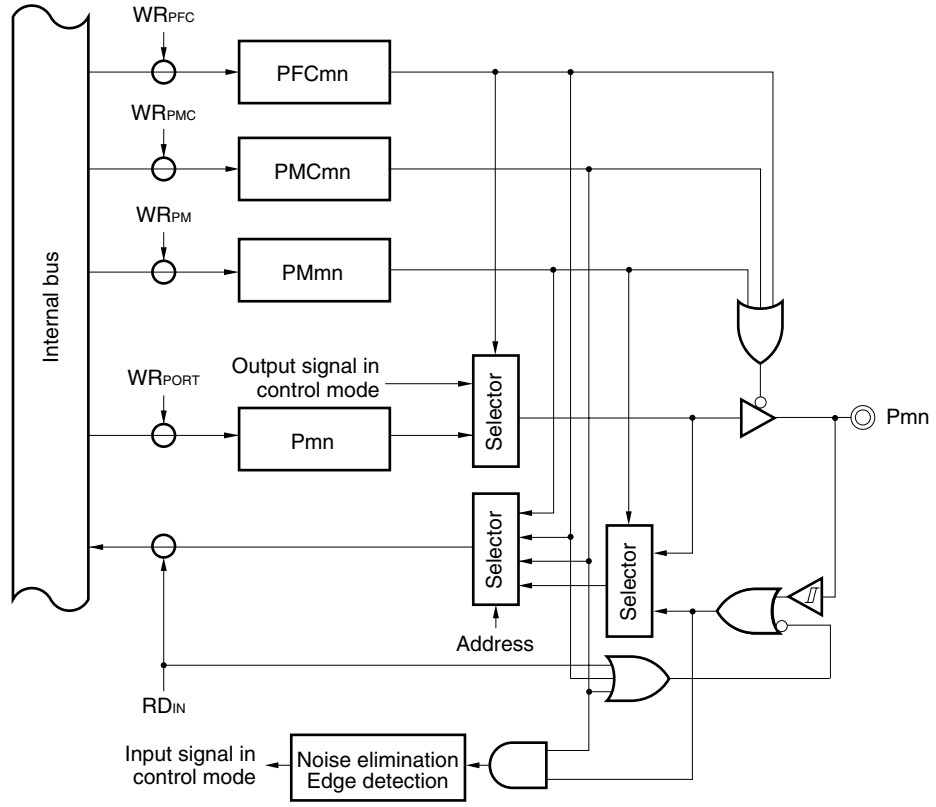


Remark n = 2, 5

x: 0 (when n = 2)

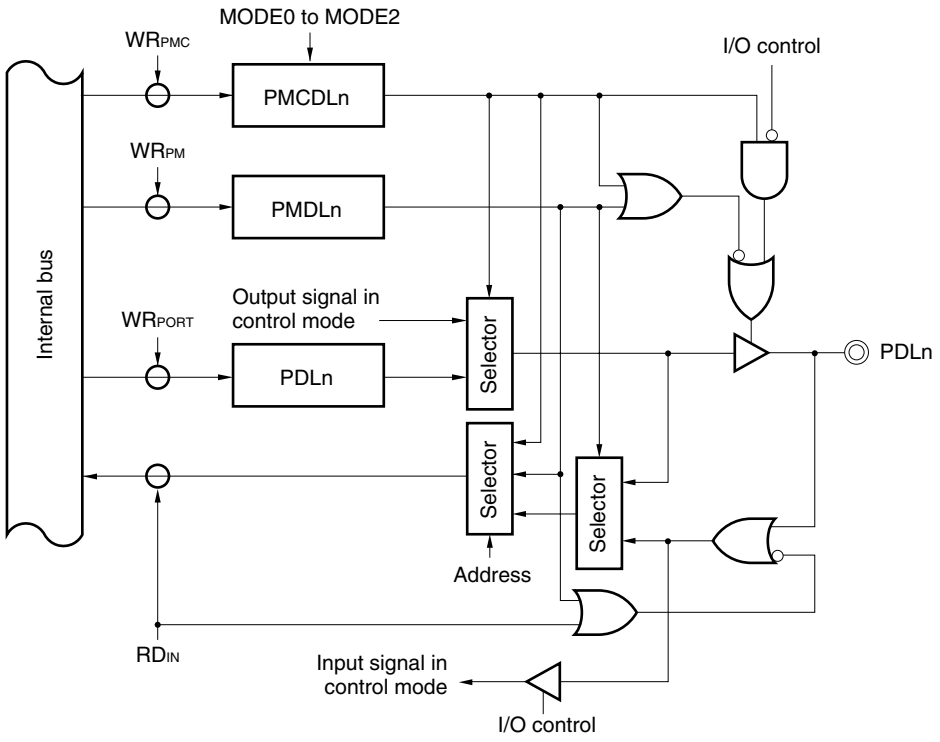
1 (when n = 5)

Figure 14-13. Block Diagram of Type N



Remark m: Port number
n: Bit number

Figure 14-14. Block Diagram of Type O



Remark n = 0 to 15

14.3 Port Pin Functions

14.3.1 Port 0

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

	7	6	5	4	3	2	1	0	Address	After reset
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFFFF400H	Undefined

Bit position	Bit name	Function
7 to 0	P0n (n = 7 to 0)	Port 0 I/O port

In addition to their function as port pins, the port 0 pins can also operate as real-time pulse unit (RPU) I/O, external interrupt request inputs, PWM output, and DMA request inputs in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 0	P00	PWM0	PWM output	A
	P01	INTP000/TI000	External interrupt request input/ Real-time pulse unit (RPU) input	B
	P02	INTP001	External interrupt request input	
	P03	TO00	Real-time pulse unit (RPU) output	A
	P04 to P07	$\overline{\text{DMARQ0}}$/$\overline{\text{INTP100}}$ to $\overline{\text{DMARQ3}}$/$\overline{\text{INTP103}}$	DMA request input/ external interrupt request input	H

(2) I/O mode/control mode setting

The port 0 I/O mode setting is performed by the port 0 mode register (PM0), and the control mode setting is performed by the port 0 mode control register (PMC0) and the port 0 function control register (PFC0).

(a) Port 0 mode register (PM0)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFFFF420H	FFH

Bit position	Bit name	Function
7 to 0	PM0n (n = 7 to 0)	Port Mode Specifies input/output mode for P0n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 0 mode control register (PMC0)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00	FFFFF440H	00H

Bit position	Bit name	Function
7 to 4	PMC0n (n = 7 to 4)	Port Mode Control Specifies operation mode of P0n pin in combination with the PFC0 register. 0: I/O port mode (output buffer on) 1: External interrupt request ($\overline{\text{INTP103}}$ to $\overline{\text{INTP100}}$) input mode/DMA request ($\overline{\text{DMARQ3}}$ to $\overline{\text{DMARQ0}}$) input mode
3	PMC03	Port Mode Control Specifies operation mode of P03 pin. 0: I/O port mode 1: TO00 output mode
2	PMC02	Port Mode Control Specifies operation mode of P02 pin. 0: I/O port mode 1: External interrupt request (INTP001) input mode
1	PMC01	Port Mode Control Specifies operation mode of P01 pin. 0: I/O port mode 1: External interrupt request (INTP000) input mode/TI000 input mode There is no register that switches between the external interrupt request (INTP000) input mode and TI000 input mode • When TI000 input mode is selected: Mask the external interrupt request (INTP000) or specify the CCC00 register as compare register. • When external interrupt request (INTP000) input mode (including timer capture input) is selected: Set the ETI0 bit of the TMCC01 register to 0.
0	PMC00	Port Mode Control Specifies operation mode of P00 pin. 0: I/O port mode 1: PWM0 output mode

(c) Port 0 function control register (PFC0)

This register can be read/written in 8-bit or 1-bit units. Bits 3 to 0, however, are fixed to 0, so writing 1 to these bits is ignored.

Caution When the port mode is specified by the port 0 mode control register (PMC0), the PFC0 setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC0	PFC07	PFC06	PFC05	PFC04	0	0	0	0	FFFFF460H	00H

Bit position	Bit name	Function
7 to 4	PFC0n (n = 7 to 4)	Port Function Control Specifies operation mode of P0n pin in control mode. 0: External interrupt request ($\overline{\text{INTP103}}$ to $\overline{\text{INTP100}}$) input mode 1: DMA ($\overline{\text{DMARQ3}}$ to $\overline{\text{DMARQ0}}$) request input mode

14.3.2 Port 1

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

	7	6	5	4	3	2	1	0	Address	After reset
P1	–	–	–	–	P13	P12	P11	P10	FFFFF402H	Undefined

Bit position	Bit name	Function
3 to 0	P1n (n = 3 to 0)	Port 1 I/O port

In addition to their function as port pins, the port 1 pins can also operate as real-time pulse unit (RPU) I/O, external interrupt request inputs, and PWM output in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 1	P10	PWM1	PWM output	A
	P11	TI010/INTP010	External interrupt request input/ Real-time pulse unit (RPU) input	B
	P12	INTP011	External interrupt request input	
	P13	TO01	Real-time pulse unit (RPU) output	A

(2) I/O mode/control mode setting

The port 1 I/O mode setting is performed by the port 1 mode register (PM1), and the control mode setting is performed by the port 1 mode control register (PMC1).

(a) Port 1 mode register (PM1)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM1	1	1	1	1	PM13	PM12	PM11	PM10	FFFFF422H	FFH

Bit position	Bit name	Function
3 to 0	PM1n (n = 3 to 0)	Port Mode Specifies input/output mode for P1n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 1 mode control register (PMC1)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC1	0	0	0	0	PMC13	PMC12	PMC11	PMC10	FFFFF442H	00H

Bit position	Bit name	Function
3	PMC13	Port Mode Control Specifies operation mode of P13 pin. 0: I/O port mode 1: TO01 output mode
2	PMC12	Port Mode Control Specifies operation mode of P12 pin. 0: I/O port mode 1: External interrupt request (INTP011) input mode
1	PMC11	Port Mode Control Specifies operation mode of P11 pin. 0: I/O port mode 1: External interrupt request (INTP010) input mode/TI010 input mode There is no register that switches between the external interrupt request (INTP010) input mode and TI010 input mode. • When the TI010 input mode is selected: Mask the external interrupt (INTP010) or specify the CCC10 register as compare register. • When external interrupt request (INTP010) input mode (including timer capture input) is selected: Set the ETI1 bit of the TMCC11 register to 0.
0	PMC10	Port Mode Control Specifies operation mode of P10 pin. 0: I/O port mode 1: PWM1 output mode

14.3.3 Port 2

Port 2 is an I/O port that can be set to the input or output mode in 1-bit units except for P20, which is an input-only pin.

Caution P20 is fixed to NMI input. The level of the NMI input can be read regardless of the PM2 and PMC2 registers' values.

	7	6	5	4	3	2	1	0	Address	After reset
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFFFF404H	Undefined

Bit position	Bit name	Function
7 to 1	P2n (n = 7 to 1)	Port 2 I/O port

In addition to their function as port pins, the port 2 pins can also operate as the real-time pulse unit (RPU) I/O, external interrupt request inputs, and the DMA end (terminal count) signal outputs in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 2	P20	NMI	Non-maskable interrupt request input	F
	P21	INTP020/TI020	External interrupt request input/ Real-time pulse unit (RPU) input	B
	P22	INTP021	External interrupt request input	
	P23	TO02	Real-time pulse unit (RPU) output	A
	P24 to 27	$\overline{TC0}$/$\overline{INTP110}$ to $\overline{TC3}$/$\overline{INTP113}$	DMA end signal outputs/External interrupt request inputs	N

(2) I/O mode/control mode setting

The port 2 I/O mode setting is performed by the port 2 mode register (PM2), and the control mode setting is performed by the port 2 function control register (PFC2).

(a) Port 2 mode register (PM2)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	1	FFFFF424H	FFH

Bit position	Bit name	Function
7 to 1	PM2n (n = 7 to 1)	Port Mode Specifies input/output mode for P2n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 2 mode control register (PMC2)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	1	FFFFF444H	01H

Bit position	Bit name	Function
7 to 4	PMC2n (n = 7 to 4)	Port Mode Control Specifies operation mode of P2n pin in combination with the PFC2 register. 0: I/O port mode (output buffer on) 1: External input request ($\overline{\text{INTP113}}$ to $\overline{\text{INTP110}}$) input mode/ DMA end signal ($\overline{\text{TC3}}$ to $\overline{\text{TC0}}$) output mode
3	PMC23	Port Mode Control Specifies operation mode of P23 pin. 0: I/O port mode 1: TO02 output mode
2	PMC22	Port Mode Control Specifies operation mode of P22 pin. 0: I/O port mode 1: External interrupt request (INTP021) input mode
1	PMC21	Port Mode Control Specifies operation mode of P21 pin. 0: I/O port mode 1: External interrupt request (INTP020) input mode/ TI020 input mode There is no register that switches between the external interrupt request (INTP020) input mode and TI020 input mode. • When the TI020 input mode is selected: Mask the external interrupt request (INTP020) or specify the CCC20 register as a compare register. • When the external interrupt request (INTP020) input mode (including timer capture input) is selected: Set the ETI2 bit of the TMCC21 register to 0.

(c) Port 2 function control register (PFC2)

This register can be read/written in 8-bit or 1-bit units. Bits 3 to 0, however, are fixed to 0 by hardware, so writing 1 to these bits is ignored.

Caution When the port mode is specified by the port 2 mode control register (PMC2), the PFC2 setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC2	PFC27	PFC26	PFC25	PFC24	0	0	0	0	FFFFF464H	00H

Bit position	Bit name	Function
7 to 4	PFC2n (n = 7 to 4)	Port Function Control Specifies operation mode of P2n pin in control mode. 0: External interrupt request ($\overline{\text{INTP113}}$ to $\overline{\text{INTP110}}$) input mode 1: DMA end signal ($\overline{\text{TC3}}$ to $\overline{\text{TC0}}$) output mode

14.3.4 Port 3

Port 3 is an 8-bit I/O port that can be set to the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFFFF406H	Undefined

Bit position	Bit name	Function
7 to 0	P3n (n = 7 to 0)	Port 3 I/O port

In addition to their function as port pins, the port 3 pins can also operate as the serial interface (CSI2, UART2) I/O, external interrupt request inputs, and A/D converter external trigger input in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 3	P30	SO2/$\overline{\text{INTP130}}$	Serial interface (CSI2) I/O/ External interrupt request inputs	N
	P31	SI2/$\overline{\text{INTP131}}$		H
	P32	$\overline{\text{SCK2}}$/$\overline{\text{INTP132}}$		I
	P33	TXD2/$\overline{\text{INTP133}}$	Serial interface (UART2) I/O/ External interrupt request inputs	N
	P34	RXD2/$\overline{\text{INTP120}}$		H
	P35	$\overline{\text{INTP121}}$	External interrupt request inputs	B
	P36	$\overline{\text{INTP122}}$		
	P37	ADTRG/$\overline{\text{INTP123}}$	A/D converter external trigger input/ External interrupt request input	

(2) I/O mode/control mode setting

The port 3 I/O mode setting is performed by the port 3 mode register (PM3), and the control mode setting is performed by the port 3 mode control register (PMC3) and the port 3 function control register 3 (PFC3).

(a) Port 3 mode register (PM3)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFFFF426H	FFH

Bit position	Bit name	Function
7 to 0	PM3n (n = 7 to 0)	Port Mode Specifies input/output mode for P3n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 3 mode control register (PMC3)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	FFFFF446H	00H

Bit position	Bit name	Function
7	PMC37	Port Mode Control Specifies operation mode of P37 pin. 0: I/O port mode 1: A/D converter external trigger (ADTRG) input mode/ External interrupt request ($\overline{\text{INTP123}}$) input mode There is no register that switches between the A/D converter external trigger (ADTRG) input mode and external interrupt request ($\overline{\text{INTP123}}$) input mode. • When the A/D converter external trigger (ADTRG) input mode is selected: Set to external trigger mode using the ADM1 register. • When the external interrupt request ($\overline{\text{INTP123}}$) input mode is selected: Set to the mode other than external trigger mode using the ADM1 register.
6	PMC36	Port Mode Control Specifies operation mode of P36 pin. 0: I/O port mode 1: External interrupt request ($\overline{\text{INTP122}}$) input mode
5	PMC35	Port Mode Control Specifies operation mode of P35 pin. 0: I/O port mode 1: External interrupt request ($\overline{\text{INTP121}}$) input mode
4	PMC34	Port Mode Control Specifies operation mode of P34 pin. 0: I/O port mode 1: RXD2 input mode/External interrupt request ($\overline{\text{INTP120}}$) input mode
3	PMC33	Port Mode Control Specifies operation mode of P33 pin. 0: I/O port mode 1: TXD2 output mode/External interrupt request ($\overline{\text{INTP133}}$) input mode
2	PMC32	Port Mode Control Specifies operation mode of P32 pin. 0: I/O port mode 1: $\overline{\text{SCK2}}$ input/output mode/External interrupt request ($\overline{\text{INTP132}}$) input mode
1	PMC31	Port Mode Control Specifies operation mode of P31 pin. 0: I/O port mode 1: SI2 input mode/External interrupt request ($\overline{\text{INTP131}}$) input mode
0	PMC30	Port Mode Control Specifies operation mode of P30 pin. 0: I/O port mode 1: SO2 output mode/External interrupt request ($\overline{\text{INTP130}}$) input mode

(c) Port 3 function control register (PFC3)

This register can be read/written in 8-bit or 1-bit units. Bits 5 to 7, however, are fixed to 0, so writing 1 to these bits is ignored.

Caution When the port mode is specified by the port 3 mode control register (PMC3), the PFC3 setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC3	0	0	0	PFC34	PFC33	PFC32	PFC31	PFC30	FFFFF466H	00H

Bit position	Bit name	Function
4	PFC34	Port Function Control Specifies operation mode of P34 pin in control mode. 0: RXD2 input mode 1: External interrupt request ($\overline{\text{INTP120}}$) input mode
3	PFC33	Port Function Control Specifies operation mode of P33 pin in control mode. 0: TXD2 output mode 1: External interrupt request ($\overline{\text{INTP133}}$) input mode
2	PFC32	Port Function Control Specifies operation mode of P32 pin in control mode. 0: $\overline{\text{SCK2}}$ I/O mode 1: External interrupt request ($\overline{\text{INTP132}}$) input mode
1	PFC31	Port Function Control Specifies operation mode of P31 pin in control mode. 0: SI2 input mode 1: External interrupt request ($\overline{\text{INTP131}}$) input mode
0	PFC30	Port Function Control Specifies operation mode of P30 pin in control mode. 0: SO2 output mode 1: External interrupt request ($\overline{\text{INTP130}}$) input mode

14.3.5 Port 4

Port 4 is a 6-bit I/O port that can be set to the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P4	–	–	P45	P44	P43	P42	P41	P40	FFFFF408H	Undefined

Bit position	Bit name	Function
5 to 0	P4n (n = 5 to 0)	Port 4 I/O port

In addition to their function as port pins, the port 4 pins can also operate as the serial interface (UART0/CSI0, UART1/CSI1) I/O in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 4	P40	TXD0/SO0	Serial interface (UART0/CSI0) I/O	L
	P41	RXD0/SI0		H
	P42	$\overline{SCK0}$		M
	P43	TXD1/SO1	Serial interface (UART1/CSI1) I/O	L
	P44	RXD1/SI1		H
	P45	$\overline{SCK1}$		M

(2) I/O mode/control mode setting

The port 4 I/O mode setting is performed by the port 4 mode register (PM4), and the control mode setting is performed by the port 4 mode control register (PMC4) and the port 4 function control register (PFC4).

(a) Port 4 mode register (PM4)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM4	1	1	PM45	PM44	PM43	PM42	PM41	PM40	FFFFF428H	FFH

Bit position	Bit name	Function
5 to 0	PM4n (n = 5 to 0)	Port Mode Specifies input/output mode for P4n pin 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 4 mode control register (PMC4)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC4	0	0	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40	FFFFF448H	00H

Bit position	Bit name	Function
5	PMC45	Port Mode Control Specifies operation mode of P45 pin. 0: I/O port mode 1: $\overline{SCK1}$ I/O mode
4	PMC44	Port Mode Control Specifies operation mode of P44 pin. 0: I/O port mode 1: RXD1/SI1 input mode
3	PMC43	Port Mode Control Specifies operation mode of P43 pin. 0: I/O port mode 1: TXD1/SO1 output mode
2	PMC42	Port Mode Control Specifies operation mode of P42 pin. 0: I/O port mode 1: $\overline{SCK0}$ I/O mode
1	PMC41	Port Mode Control Specifies operation mode of P41 pin. 0: I/O port mode 1: RXD0/SI0 input mode
0	PMC40	Port Mode Control Specifies operation mode of P40 pin. 0: I/O port mode 1: TXD0/SO0 output mode

(c) Port 4 function control register (PFC4)

This register can be read/written in 8-bit or 1-bit units. Bits 7 to 5 and 2, however, are fixed to 0, so writing 1 to these bits is ignored.

Caution When the port mode is specified by the port 4 mode control register (PMC4), the PFC4 register setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFC4	0	0	0	PFC44	PFC43	0	PFC41	PFC40	FFFFF468H	00H

Bit position	Bit name	Function
4	PFC44	Port Function Control Specifies operation mode of P44 pin in control mode. 0: SI1 input mode 1: RXD1 input mode
3	PFC43	Port Function Control Specifies operation mode of P43 pin in control mode. 0: SO1 output mode 1: TXD1 output mode
1	PFC41	Port Function Control Specifies operation mode of P41 pin in control mode. 0: SI0 input mode 1: RXD0 input mode
0	PFC40	Port Function Control Specifies operation mode of P40 pin in control mode. 0: SO0 output mode 1: TXD0 output mode

14.3.6 Port 5

Port 5 is a 3-bit I/O port that can be set to the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P5	–	–	–	–	–	P52	P51	P50	FFFFF40AH	Undefined

Bit position	Bit name	Function
2 to 0	P5n (n = 2 to 0)	Port 5 I/O port

In addition to their function as port pins, the port 5 pins can also operate as the real-time pulse unit (RPU) I/O and external interrupt request inputs in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 5	P50	INTP030/TI030	External interrupt request input/ Real-time pulse unit (RPU) input	B
	P51	INTP031	External interrupt request input	
	P52	TO03	Real-time pulse unit (RPU) output	A

(2) I/O mode/control mode setting

The port 5 I/O mode setting is performed by the port 5 mode register (PM5), and the control mode setting is performed by the port 5 mode control register (PMC5).

(a) Port 5 mode register (PM5)

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

	7	6	5	4	3	2	1	0	Address	After reset
PM5	1	1	1	1	1	PM52	PM51	PM50	FFFFF42AH	FFH

Bit position	Bit name	Function
2 to 0	PM5n (n = 2 to 0)	Port Mode Specifies input/output mode for P5n pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port 5 mode control register (PMC5)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMC5	0	0	0	0	0	PMC52	PMC51	PMC50	FFFFF44AH	00H

Bit position	Bit name	Function
2	PMC52	Port Mode Control Specifies operation mode of P52 pin. 0: I/O port mode 1: TO03 output mode
1	PMC51	Port Mode Control Specifies operation mode of P51 pin. 0: I/O port mode 1: External input request (INTP031) input mode
0	PMC50	Port Mode Control Specifies operation mode of P50 pin. 0: I/O port mode 1: External interrupt request (INTP030) input mode/TI030 input mode There is no register that switches between the external interrupt request (INTP030) input mode and TI030 input mode. • When the TI030 input mode is selected: Mask the external interrupt request (INTP030) or specify the CCC30 register as a compare register. • When the external interrupt request (INTP030) input mode (including timer capture input) is selected: Set the ETI3 bit of the TMCC31 register to 0.

14.3.7 Port 7

Port 7 is an 8-bit input-only port whose pins are fixed to input.

	7	6	5	4	3	2	1	0	Address	After reset
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFFFF40EH	Undefined

Bit position	Bit name	Function
7 to 0	P7n (n = 7 to 0)	Port 7 Input-only port

In addition to their function as port pins, the port 7 pins can also operate as the analog inputs to the A/D converter in the control mode.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port 7	P77 to P70	ANI7 to ANI0	Analog input to A/D converter	C

14.3.8 Port AL

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

When the higher 8 bits of port AL are used as port ALH (PALH) and the lower 8 bits as port ALL (PALL), port AL becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset
PAL	PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8	FFFFF001H	Undefined
	7	6	5	4	3	2	1	0	Address	
	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0	FFFFF000H	

Bit position	Bit name	Function
15 to 0	PALn (n = 15 to 0)	Port AL I/O port

In addition to their functions as port pins, in the control mode, the port AL pins operate as an address bus for when the memory is externally expanded.

(1) Operation in control mode

Port		Alternate Function	Remark	Block Type
Port AL	PAL15 to PAL0	A15 to A0	Address bus when memory expanded	J

(2) I/O mode/control mode setting

The port AL I/O mode setting is performed by the port AL mode register (PMAL), and control mode setting is performed by the port AL mode control register (PMCAL).

(a) Port AL mode register (PMAL)

The port AL mode register (PMAL) can be read/written in 16-bit units.

If the higher 8 bits of PMAL are used as port AL mode register H (PMALH), and the lower 8 bits as port AL mode register L (PMALL), these two 8-bit port mode registers can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset
PMAL	PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8	FFFFF021H	FFFFH
	7	6	5	4	3	2	1	0	Address	
	PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0	FFFFF020H	

Bit position	Bit name	Function
15 to 0	PMALn (n = 15 to 0)	Port Mode Specifies input/output mode for PALn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port AL mode control register (PMCAL)

The port AL mode control register (PMCAL) can be read/written in 16-bit units.

If the higher 8 bits of PMCAL are used as port AL mode control register H (PMCALH), and the lower 8 bits as port AL mode control register L (PMCALL), these two 8-bit port mode registers can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset[Note]
PMCAL	PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8	FFFFF041H	FFFFH/0000H
	7	6	5	4	3	2	1	0	Address	
	PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0	FFFFF040H	

Note In ROMless modes 0 and 1, and single-chip mode 1: FFFFH
In single-chip mode 0: 0000H

Bit position	Bit name	Function
15 to 0	PMCALn (n = 15 to 0)	Port Mode Control Specifies operation mode of PALn pin. 0: I/O port mode 1: A15 to A0 output mode

14.3.9 Port AH

Port AH (PAH) is a 16-bit I/O port that can be set in the input or output mode in 1-bit units.

When the higher 8 bits of port AH are used as port AHH (PAHH) and the lower 8 bits as port AHL (PAHL), port AH becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

Bits 15 to 10 of port AH (bits 7 to 2 of port AHH) are undefined.

	15	14	13	12	11	10	9	8	Address	After reset
PAH	–	–	–	–	–	–	PAH9	PAH8	FFFFF003H	Undefined
	7	6	5	4	3	2	1	0	Address	
	PAH7	PAH6	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0	FFFFF002H	

Bit position	Bit name	Function
9 to 0	PAHn (n = 9 to 0)	Port AH I/O port

In addition to their functions as port pins, in the control mode, the port AH pins operate as an address bus for when the memory is externally expanded.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port AH	PAL9 to PAL0	A25 to A16	Address bus when memory expanded	J

(2) I/O mode/control mode setting

The port AH I/O mode setting is performed by the port AH mode register (PMAH), and the control mode setting is performed by the port AH mode control register (PMCAH).

(a) Port AH mode register (PMAH)

The port AH mode register (PMAH) can be read/written in 16-bit units.

If the higher 8 bits of PMAH are used as port AH mode register H (PMAHH), and the lower 8 bits as port AH mode register L (PMAHL), these two 8-bit port mode registers can be read/written in 8-bit or 1-bit units.

Bits 15 to 10 of PMAH (bits 7 to 2 of PMAHH) are fixed to 1.

	15	14	13	12	11	10	9	8	Address	After reset
PMAH	1	1	1	1	1	1	PMAH9	PMAH8	FFFFF023H	FFFFH
	7	6	5	4	3	2	1	0	Address	
	PMAH7	PMAH6	PMAH5	PMAH4	PMAH3	PMAH2	PMAH1	PMAH0	FFFFF022H	

Bit position	Bit name	Function
9 to 0	PMAHn (n = 9 to 0)	Port Mode Specifies input/output mode for PAHn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port AH mode control register (PMCAH)

The port AH mode control register (PMCAH) can be read/written in 16-bit units.

If the higher 8 bits of PMCAH are used as port AH mode control register H (PMCAHH), and the lower 8 bits as port AH mode control register L (PMCAHL), these two 8-bit port mode registers can be read/written in 8-bit or 1-bit units.

Bits 15 to 10 of PMCAH (bits 7 to 2 of PMCAHH) are fixed to 0.

	15	14	13	12	11	10	9	8	Address	After reset[Note]
PMCAH	0	0	0	0	0	0	PMCAH9	PMCAH8	FFFFF043H	0000H/03FFH
	7	6	5	4	3	2	1	0	Address	
	PMCAH7	PMCAH6	PMCAH5	PMCAH4	PMCAH3	PMCAH2	PMCAH1	PMCAH0	FFFFF042H	

Note In ROMless modes 0 and 1, and single-chip mode 1: 03FFH
In single-chip mode 0: 0000H

Bit position	Bit name	Function
9 to 0	PMCAHn (n = 9 to 0)	Port Mode Control Specifies operation mode of PAHn pin. 0: I/O port mode 1: A25 to A16 output mode

14.3.10 Port DL

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

When the higher 8 bits of port DL are used as port DLH (PDLH), and the lower 8 bits as port DLL (PDLL), port DL becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset
PDL	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8	FFFFF005H	Undefined
	7	6	5	4	3	2	1	0	Address	
	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0	FFFFF004H	

Bit position	Bit name	Function
15 to 0	PDLn (n = 15 to 0)	Port DL I/O port

In addition to their functions as port pins, in the control mode, the port DL pins operate as a data bus for when the memory is externally expanded.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port DL	PDL15 to PDL0	D15 to D0	Data bus when memory expanded	O

(2) I/O mode/control mode setting

The port DL I/O mode setting is performed by the port DL mode register (PMDL), and the control mode setting is performed by the port DL mode control register (PMCDL).

(a) Port DL mode register (PMDL)

The port DL mode register (PMDL) can be read/written in 16-bit units.

If the higher 8 bits of PMDL are used as port DL mode register H (PMDLH), and the lower 8 bits as port DL mode register L (PMDLL), these two 8-bit port mode registers can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset
PMDL	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8	FFFFF025H	FFFFH
	7	6	5	4	3	2	1	0	Address	
	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0	FFFFF024H	

Bit position	Bit name	Function
15 to 0	PMDLn (n = 15 to 0)	Port Mode Specifies input/output mode for PDLn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port DL mode control register (PMCDL)

The port DL mode control register (PMCDL) can be read/written in 16-bit units.

If the higher 8 bits of PMCDL are used as port DL mode control register H (PMCDLH), and the lower 8 bits as port DL mode control register L (PMCDLL), these two 8-bit port mode registers can be read/written in 8-bit or 1-bit units.

	15	14	13	12	11	10	9	8	Address	After reset[Note]
PMCDL	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8	FFFFF045H	FFFFH/0000H
	7	6	5	4	3	2	1	0	Address	
	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0	FFFFF044H	

Note In ROMless modes 0 and 1, and single-chip mode 1: FFFFH
In single-chip mode 0: 0000H

Bit position	Bit name	Function
15 to 0	PMCDLn (n = 15 to 0)	Port Mode Control Specifies operation mode of PDLn pin. 0: I/O port mode 1: D15 to D0 output mode

14.3.11 Port CS

Port CS is an 8-bit I/O port that can be set to the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PCS	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0	FFFFF008H	Undefined

Bit position	Bit name	Function
7 to 0	PCSn (n = 7 to 0)	Port CS I/O port

In addition to their function as port pins, in the control mode, the port pins can also operate as the chip select signal outputs when memory is externally expanded, the row address strobe signal outputs to DRAM, and the read/write strobe signal output to an external I/O.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port CS	PCS0	$\overline{CS0}$	Chip select signal output	J
	PCS1	$\overline{CS1}$/$\overline{RAS1}$	Chip select signal output/ row address signal output	
	PCS2	$\overline{CS2}$/$\overline{IOWR}$	Chip select signal output/ write strobe signal output	K
	PCS3	$\overline{CS3}$/$\overline{RAS3}$	Chip select signal output/ row address signal output	J
	PCS4	$\overline{CS4}$/$\overline{RAS4}$		
	PCS5	$\overline{CS5}$/$\overline{IORD}$	Chip select signal output/ read strobe signal output	K
	PCS6	$\overline{CS6}$/$\overline{RAS6}$	Chip select signal output/ row address signal output	J
	PCS7	$\overline{CS7}$	Chip select signal output	

(2) I/O mode/control mode setting

The port CS I/O mode setting is performed by the port CS mode register (PMCS), and the control mode setting is performed by the port CS mode control register (PMCCS) and the port CS function control register (PFCCS).

(a) Port CS mode register (PMCS)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCS	PMCS7	PMCS6	PMCS5	PMCS4	PMCS3	PMCS2	PMCS1	PMCS0	FFFFF028H	FFH

Bit position	Bit name	Function
7 to 0	PMCSn (n = 7 to 0)	Port Mode Specifies input/output mode for PCSn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port CS mode control register (PMCCS)

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

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCCS	PMCCS7	PMCCS6	PMCCS5	PMCCS4	PMCCS3	PMCCS2	PMCCS1	PMCCS0	FFFFF048H	00H/FFH

Note In ROMless modes 0 and 1, and single-chip mode 1: FFH
In single-chip mode 0: 00H

Bit position	Bit name	Function
7	PMCCS7	Port Mode Control Specifies operation mode of PCS7 pin. 0: I/O port mode 1: $\overline{CS7}$ output mode
6	PMCCS6	Port Mode Control Specifies operation mode of PCS6 pin. 0: I/O port mode 1: $\overline{CS6}$/$\overline{RAS6}$ output mode ($\overline{CS6}$/$\overline{RAS6}$ signal automatically switched by accessing the targeted memory of each signal.)
5	PMCCS5	Port Mode Control Specifies operation mode of PCS5 pin. 0: I/O port mode 1: $\overline{CS5}$ output mode/$\overline{IORD}$ output mode
4	PMCCS4	Port Mode Control Specifies operation mode of PCS4 pin. 0: I/O port mode 1: $\overline{CS4}$/$\overline{RAS4}$ output mode ($\overline{CS4}$/$\overline{RAS4}$ signal automatically switched by accessing the targeted memory of each signal.)
3	PMCCS3	Port Mode Control Specifies operation mode of PCS3 pin. 0: I/O port mode 1: $\overline{CS3}$/$\overline{RAS3}$ output mode ($\overline{CS3}$/$\overline{RAS3}$ signal automatically switched by accessing the targeted memory of each signal.)
2	PMCCS2	Port Mode Control Specifies operation mode of PCS2 pin. 0: I/O port mode 1: $\overline{CS2}$ output mode/$\overline{IOWR}$ output mode
1	PMCCS1	Port Mode Control Specifies operation mode of PCS1 pin. 0: I/O port mode 1: $\overline{CS1}$/$\overline{RAS1}$ output mode ($\overline{CS1}$/$\overline{RAS1}$ signal automatically switched by accessing the targeted memory of each signal.)
0	PMCCS0	Port Mode Control Specifies operation mode of PCS0 pin. 0: I/O port mode 1: $\overline{CS0}$ output mode

(c) Port CS function control register (PFCCS)

This register can be read/written in 8-bit or 1-bit units. Bits 7, 6, 4, 3, 1, and 0, however, are fixed to 0, so writing 1 to these bits is ignored.

Caution When the port mode is specified by the port CS mode control register (PMCCS), the PFCCS setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFCCS	0	0	PFCCS5	0	0	PFCCS2	0	0	FFFFF049H	00H

Bit position	Bit name	Function
5	PFCCS5	Port Function Control Specifies operation mode of PCS5 pin in control mode. 0: $\overline{\text{CS5}}$ output mode 1: $\overline{\text{IORD}}$ output mode[Note]
2	PFCCS2	Port Function Control Specifies operation mode of PCS2 pin in control mode. 0: $\overline{\text{CS2}}$ output mode 1: $\overline{\text{IOWR}}$ output mode[Note]

Note To output the $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ signals during access to the external I/O other than by a DMA flyby transfer, the IOEN bit of the BCP register must be set.

14.3.12 Port CT

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

	7	6	5	4	3	2	1	0	Address	After reset
PCT	PCT7	PCT6	PCT5	PCT4	–	–	PCT1	PCT0	FFFFF00AH	Undefined

Bit position	Bit name	Function
7 to 4,1,0	PCTn (n = 7 to 4,1,0)	Port CT I/O port

In addition to their function as port pins, in the control mode, the port CT pins operate as control signal outputs for when the memory is externally expanded.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port CT	PCT0	$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM	Column address signal output/ write strobe signal output/ output disable/write mask signal	J
	PCT1	$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM	Column address signal output/ write strobe signal output/ output disable/write mask signal	
	PCT4	$\overline{\text{RD}}$	Read strobe signal output	
	PCT5	$\overline{\text{WE}}$	Write enable signal output	
	PCT6	$\overline{\text{OE}}$	Output enable signal output	
	PCT7	$\overline{\text{BCYST}}$	Bus cycle status signal output	

(2) I/O mode/control mode setting

The port CT I/O mode setting is performed by the port CT mode register (PMCT), and the control mode setting is performed by the port CT mode control register (PMCCT).

(a) Port CT mode register (PMCT)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCT	PMCT7	PMCT6	PMCT5	PMCT4	1	1	PMCT1	PMCT0	FFFFF02AH	FFH

Bit position	Bit name	Function
7 to 4,1,0	PMCTn (n = 7 to 4,1,0)	Port Mode Specifies input/output mode for PCTn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port CT mode control register (PMCCT)

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

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCCT	PMCCT7	PMCCT6	PMCCT5	PMCCT4	0	0	PMCCT1	PMCCT0	FFFFF04AH	00H/F3H

Note In ROMless modes 0 and 1, and single-chip mode 1: F3H
In single-chip mode 0: 00H

Bit position	Bit name	Function
7	PMCCT7	Port Mode Control Specifies operation mode of PCT7 pin. 0: I/O port mode 1: $\overline{\text{BCYST}}$ output mode
6	PMCCT6	Port Mode Control Specifies operation mode of PCT6 pin. 0: I/O port mode 1: $\overline{\text{OE}}$ output mode
5	PMCCT5	Port Mode Control Specifies operation mode of PCT5 pin. 0: I/O port mode 1: $\overline{\text{WE}}$ output mode
4	PMCCT4	Port Mode Control Specifies operation mode of PCT4 pin. 0: I/O port mode 1: $\overline{\text{RD}}$ output mode
1	PMCCT1	Port Mode Control Specifies operation mode of PCT1 pin. 0: I/O port mode 1: $\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM output mode ($\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/UDQM signal automatically switched by accessing the targeted memory of each signal.)
0	PMCCT0	Port Mode Control Specifies operation mode of PCT0 pin. 0: I/O port mode 1: $\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM output mode ($\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/LDQM signal automatically switched by accessing the targeted memory of each signal.)

14.3.13 Port CM

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

	7	6	5	4	3	2	1	0	Address	After reset
PCM	–	–	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0	FFFFF00CH	Undefined

Bit position	Bit name	Function
5 to 0	PCMn (n = 5 to 0)	Port CM I/O port

In addition to their function as port pins, in the control mode, the port CM pins operate as the wait insertion signal input, internal system clock output/bus clock output, bus hold control signal output, and refresh request signal output from DRAM.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port CM	PCM0	$\overline{\text{WAIT}}$	Wait insertion signal input	D
	PCM1	CLKOUT/BUSCLK	Internal system clock output/bus clock output	K
	PCM2	$\overline{\text{HLDAK}}$	Bus hold acknowledge signal output	J
	PCM3	$\overline{\text{HLDRQ}}$	Bus hold request signal input	D
	PCM4	$\overline{\text{REFRQ}}$	Refresh request signal output	J
	PCM5	SELFREF	Self-refresh request signal input	E

(2) I/O mode/control mode setting

The port CM I/O mode setting is performed by the port CM mode register (PMCM), and the control mode setting is performed by the port CM mode control register (PMCCM) and the port CM function control register (PFCCM).

(a) Port CM mode register (PMCM)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCM	1	1	PMCM5	PMCM4	PMCM3	PMCM2	PMCM1	PMCM0	FFFFF02CH	FFH

Bit position	Bit name	Function
5 to 0	PMCMn (n = 5 to 0)	Port Mode Specifies input/output mode for PCMn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port CM mode control register (PMCCM)

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

Caution If the mode of the PCM1/CLKOUT/BUSCLK pin is changed from the I/O port mode to the CLKOUT/BUSCLK mode, a glitch may be generated in the CLKOUT/BUSCLK output immediately after the change. Therefore, pull up the CLKOUT/BUSCLK pin when using it. In the PLL mode (CKSEL = 0), change the mode to the CLKOUT/BUSCLK mode at a multiple of 1 (CKDIV2 to CKDIV0 bits of CKC register = 000B).

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCCM	0	0	PMCCM5	PMCCM4	PMCCM3	PMCCM2	PMCCM1	PMCCM0	FFFFF04CH	00H/3FH

Note In ROMless modes 0 and 1, and single-chip mode 1: 3FH
In single-chip mode 0: 00H

Bit position	Bit name	Function
5	PMCCM5	Port Mode Control Specifies operation mode of PCM5 pin. 0: I/O port mode 1: SELFREF input mode
4	PMCCM4	Port Mode Control Specifies operation mode of PCM4 pin. 0: I/O port mode 1: $\overline{\text{REFRQ}}$ output mode
3	PMCCM3	Port Mode Control Specifies operation mode of PCM3 pin. 0: I/O port mode 1: $\overline{\text{HLDRQ}}$ input mode
2	PMCCM2	Port Mode Control Specifies operation mode of PCM2 pin. 0: I/O port mode 1: $\overline{\text{HLDAK}}$ output mode
1	PMCCM1	Port Mode Control Specifies operation mode of PCM1 pin. 0: I/O port mode 1: CLKOUT output mode/BUSCLK output mode
0	PMCCM0	Port Mode Control Specifies operation mode of PCM0 pin. 0: I/O port mode 1: $\overline{\text{WAIT}}$ input mode

(c) Port CM function control register (PFCCM)

This register can be read/written in 8-bit or 1-bit units. Bits 7 to 2 and 0, however, are fixed to 0, so writing 1 to these bits is ignored. To output the half clock of the internal system clock from the BUSCLK pin, the BCP bit of the BCP register must be set to 1.

If the BCP bit of the BCP register is set to 1 with the CLKOUT output mode selected, the external bus operates at half the frequency of the internal system clock frequency, but the CLKOUT pin outputs the internal operating frequency.

Caution When the port mode is specified by the port CM mode control register (PMCCM), the PFCCM setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFCCM	0	0	0	0	0	0	PFCCM1	0	FFFFF04DH	00H

Bit position	Bit name	Function
1	PFCCM1	Port Function Control Specifies operation mode of PCM1 pin in control mode. 0: CLKOUT output mode 1: BUSCLK output mode

14.3.14 Port CD

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

	7	6	5	4	3	2	1	0	Address	After reset
PCD	–	–	–	–	PCD3	PCD2	PCD1	PCD0	FFFFF00EH	Undefined

Bit position	Bit name	Function
3 to 0	PCDn (n = 3 to 0)	Port CD I/O port

In addition to their function as port pins, the port CD pins operate as the clock enable signal output to SDRAM, synchronous clock output, column address strobe signal output, row address strobe signal output, and byte enable signal output to SDRAM upon byte access, in the control mode.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port CD	PCD0	SDCKE	Clock enable signal output	J
	PCD1	SDCLK	Synchronous clock output	
	PCD2	$\overline{\text{LBE}}$/$\overline{\text{SDCAS}}$	Byte enable signal output/ column address strobe signal output	K
	PCD3	$\overline{\text{UBE}}$/$\overline{\text{SDRAS}}$	Byte enable signal output/ row address strobe signal output	

(2) I/O mode/control mode setting

The port CD I/O mode setting is performed by the port CD mode register (PMCD), and the control mode setting is performed by the port CD mode control register (PMCCD) and the port CD function control register (PFCCD).

(a) Port CD mode register (PMCD)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCD	1	1	1	1	PMCD3	PMCD2	PMCD1	PMCD0	FFFFF02EH	FFH

Bit position	Bit name	Function
3 to 0	PMCDn (n = 3 to 0)	Port Mode Specifies input/output mode for PCDn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port CD mode control register (PMCCD)

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

Cautions 1. Do not perform the SDCLK and SDCKE output mode setting simultaneously. Be sure to perform the SDCLK output mode setting before the SDCKE output mode setting.

2. When in single-chip mode 1, and in ROMless modes 0 and 1, bits 1 and 0 of the PMCCD register become SDCLK output mode and SDCKE output mode after the reset is released, however, bits 3 and 2 become $\overline{\text{UBE}}$ output mode and $\overline{\text{LBE}}$ output mode. When using SDRAM be sure to set the $\overline{\text{SDRAS}}$ output mode and $\overline{\text{SDCAS}}$ output mode using the PFCCD register.

	7	6	5	4	3	2	1	0	Address	After reset[Note]
PMCCD	0	0	0	0	PMCCD3	PMCCD2	PMCCD1	PMCCD0	FFFFF04EH	00H/0FH

Note In ROMless modes 0 and 1, and single-chip mode 1: 0FH
In single-chip mode 0: 00H

Bit position	Bit name	Function
3	PMCCD3	Port Mode Control Specifies operation mode of PCD3 pin. 0: I/O port mode 1: $\overline{\text{UBE}}$/$\overline{\text{SDRAS}}$ output mode
2	PMCCD2	Port Mode Control Specifies operation mode of PCD2 pin. 0: I/O port mode 1: $\overline{\text{LBE}}$/$\overline{\text{SDCAS}}$ output mode
1	PMCCD1	Port Mode Control Specifies operation mode of PCD1 pin. 0: I/O port mode 1: SDCLK output mode
0	PMCCD0	Port Mode Control Specifies operation mode of PCD0 pin. 0: I/O port mode 1: SDCKE output mode

(c) Port CD function control register (PFCCD)

This register can be read/written in 8-bit or 1-bit units. Bits 7 to 4, 1, and 0, however, are fixed to 0, so writing 1 to these bits is ignored.

Caution When the port mode is specified by the port CD mode control register (PMCCD), the PFCCD setting becomes invalid.

	7	6	5	4	3	2	1	0	Address	After reset
PFCCD	0	0	0	0	PFCCD3	PFCCD2	0	0	FFFFF04FH	00H

Bit position	Bit name	Function
3	PFCCD3	Port Function Control Specifies operation mode of PCD3 pin in control mode. 0: $\overline{\text{UBE}}$ output mode 1: $\overline{\text{SDRAS}}$ output mode
2	PFCCD2	Port Function Control Specifies operation mode of PCD2 pin in control mode. 0: $\overline{\text{LBE}}$ output mode 1: $\overline{\text{SDCAS}}$ output mode

14.3.15 Port BD

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

	7	6	5	4	3	2	1	0	Address	After reset
PBD	–	–	–	–	PBD3	PBD2	PBD1	PBD0	FFFFF012H	Undefined

Bit position	Bit name	Function
3 to 0	PBDn (n = 3 to 0)	Port BD I/O port

In addition to their function as port pins, the port BD pins operate as the DMA acknowledge signal outputs in the control mode.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port BD	PBD0 to PBD3	$\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$	DMA acknowledge signal output	J

(2) I/O mode/control mode setting

The port BD I/O mode setting is performed by the port BD mode register (PMBD), and the control mode setting is performed by the port BD mode control register (PMCBD).

(a) Port BD mode register (PMBD)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMBD	1	1	1	1	PMBD3	PMBD2	PMBD1	PMBD0	FFFFF032H	FFH

Bit position	Bit name	Function
3 to 0	PMBDn (n = 3 to 0)	Port Mode Specifies input/output mode for PBDn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(b) Port BD mode control register (PMCBD)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCBD	0	0	0	0	PMCBD3	PMCBD2	PMCBD1	PMCBD0	FFFFF052H	00H

Bit position	Bit name	Function
3 to 0	PMCBDn (n = 3 to 0)	Port Mode Control Specifies operation mode of PBDn pin. 0: I/O port mode 1: $\overline{\text{DMAAK}}$n output mode

CHAPTER 15 RESET FUNCTIONS

When a low-level signal is input to the $\overline{\text{RESET}}$ pin, a system reset is effected and the hardware is initialized.

When the $\overline{\text{RESET}}$ signal level changes from low to high, the reset state is released and CPU starts program execution. Register contents must be initialized as required in the program.

15.1 Features

The reset pin ($\overline{\text{RESET}}$) incorporates a noise eliminator that uses analog delay (≒ 60 ns) to prevent malfunction due to noise.

15.2 Pin Functions

During a system reset, most pins (all but the CLKOUT[Note], $\overline{\text{RESET}}$, X2, V_{DD}, V_{SS}, CV_{DD}, CV_{SS}, AV_{DD}/AV_{REF}, and AV_{SS} pins) enter the high-impedance state. Therefore, when memory is connected externally, a pull-up or pull-down resistor must be connected to the specified pins of ports AL, AH, DL, CS, CT, CM, CD, and BD. If no resistor is connected, external memory may be destroyed when these pins enter the high-impedance state.

For the same reason, the output pins of the internal peripheral I/O functions and other output ports should be handled in the same manner.

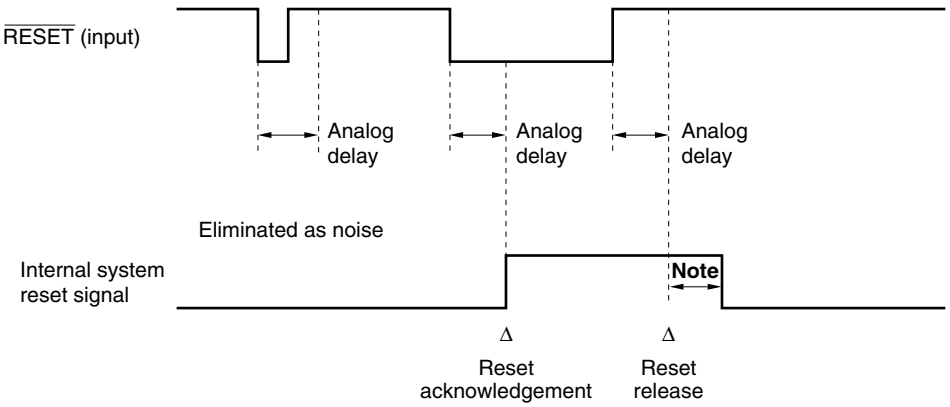
Note In ROMless modes 0 and 1, and in single-chip mode 1, the CLKOUT signal is output even during reset. In single-chip mode 0, the CLKOUT signal is not output until the PMCCM register is set.

The operation status of each pin during reset is shown below (Table 15-1).

Table 15-1. Operation Status of Each Pin During Reset

Pin Name		Pin State			
		Single-Chip Mode 0	Single-Chip Mode 1	ROMless Mode 0	ROMless Mode 1
A0 to A15, A16 to A25, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS7}}$, $\overline{\text{RAS1}}$, $\overline{\text{RAS3}}$, $\overline{\text{RAS4}}$, $\overline{\text{RAS6}}$, $\overline{\text{LWR}}$, $\overline{\text{UWR}}$, $\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$, LDQM, UDQM, $\overline{\text{RD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{BCYST}}$, $\overline{\text{WAIT}}$, $\overline{\text{HLDAK}}$, $\overline{\text{HLDRQ}}$, $\overline{\text{REFREQ}}$, SELFREF, SDCKE, SDCLK, $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$		(Port mode)	High impedance		
CLKOUT		(Port mode)	Operating		
Port pin	Ports 0 to 5, 7, BD	(Input)			
	Ports AL, AH, DL, CM, CT, CS, CD	(Input)	(Control mode)		

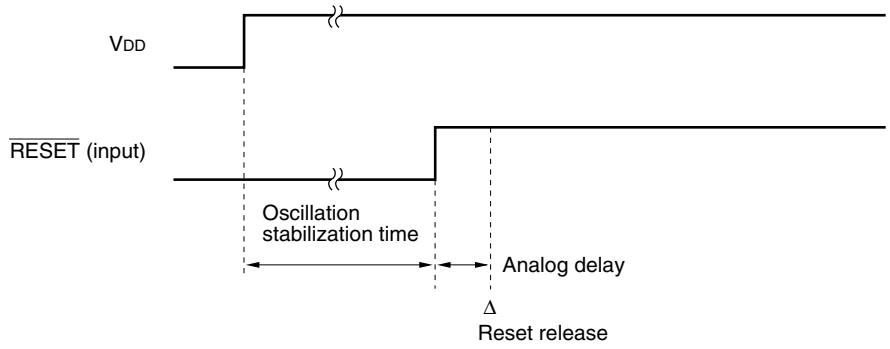
(1) Acknowledging the reset signal



Note The internal system reset signal continues in the active state for at least 4 system clock cycles after reset clear timing by the RESET signal.

(2) Reset when turning on the power

In a reset operation when the power is turned on, because of the low-level width of the RESET signal, it is necessary to secure the oscillation stabilization time between when the power is turned on and when the reset is acknowledged.



15.3 Initialization

Initialize the contents of each register as necessary while programming.

The initial values of the CPU, internal RAM, and on-chip peripheral I/O after a reset are shown in Table 15-2.

Table 15-2. Initial Value of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (1/3)

Internal Hardware		Register Name	Initial Value After Reset
CPU	Program registers	General-purpose register (r0)	00000000H
		General-purpose registers (r1 to r31)	Undefined
		Program counter (PC)	00000000H
	System registers	Status saving registers during interrupt (EIPC, EIPSW)	Undefined
		Status saving registers during NMI (FEPC, FEPSW)	Undefined
		Interrupt source register (ECR)	00000000H
		Program status word (PSW)	00000020H
		Status saving registers during CALLT execution (CTPC, CTPSW)	Undefined
		Status saving registers during exception/debug trap (DBPC, DBPSW)	Undefined
		CALLT base pointer (CTBP)	Undefined
Internal RAM		—	Undefined
On-chip peripheral I/O	Port functions	Ports (P0 to P5, P7, PAL, PAH, PDL, PCS, PCT, PCM, PCD, PBD)	Undefined
		Mode registers (PM0 to PM5, PMCS, PMCT, PMCM, PMCD, PMBD)	FFH
		Mode registers (PMAL, PMAH, PMDL)	FFFFH
		Mode control registers (PMC0, PMC1, PMC3 to PMC5, PMCBD)	00H
		Mode control register (PMC2)	01H
		Mode control registers (PMCAL, PMCDL)	0000H/FFFFH
		Mode control register (PMCAH)	0000H/03FFH
		Mode control register (PMCCS)	00H/FFH
		Mode control register (PMCCT)	00H/F3H
		Mode control register (PMCCM)	00H/3FH
		Mode control register (PMCCD)	00H/0FH
		Function control registers (PFC0, PFC2 to PFC4, PFCCS, PFCCM, PFCCD)	00H
	Timer/counter functions	Timer Cn (TMCn) (n = 0 to 3)	0000H
		Capture/compare registers Cn0 and Cn1 (CCCn0 and CCCn1) (n = 0 to 3)	0000H
		Timer mode control register Cn0 (TMCCn0) (n = 0 to 3)	00H
		Timer mode control register Cn1 (TMCCn1) (n = 0 to 3)	20H
		Timer Dn (TMDn) (n = 0 to 3)	0000H
		Compare register (CMDn) (n = 0 to 3)	0000H
		Timer mode control register Dn (n = 0 to 3)	00H

Table 15-2. Initial Value of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (2/3)

Internal Hardware		Register Name	Initial Value After Reset
On-chip peripheral I/O	Serial interface functions	Clocked serial interface mode register n (CSIMn) (n = 0 to 2)	00H
		Clocked serial interface clock select register n (CSICn) (n = 0 to 2)	00H
		Clocked serial interface transmit buffer register n (SOTBn) (n = 0 to 2)	00H
		Serial I/O shift register n (SIOn) (n = 0 to 2)	00H
		Receive-only serial I/O shift register n (SIOEn) (n = 0 to 2)	00H
		Receive buffer register n (RXBn) (n = 0 to 2)	FFH
		Transmit buffer register n (TXBn) (n = 0 to 2)	FFH
		Asynchronous serial interface mode register n (ASIMn) (n = 0 to 2)	01H
		Asynchronous serial interface status register n (ASISn) (n = 0 to 2)	00H
		Asynchronous serial interface transmit status register n (ASIFn) (n = 0 to 2)	00H
		Clock select register n (CKSRn) (n = 0 to 2)	00H
		Baud rate generator control register n (BRGCn) (n = 0 to 2)	FFH
	A/D converter	A/D converter mode registers 0 and 2 (ADM0 and ADM2)	00H
		A/D converter mode register 1 (ADM1)	07H
		A/D conversion result register n (10 bits) (n = 0 to 7)	0000H
		A/D conversion result register nH (8 bits) (n = 0 to 7)	00H
	PWM	PWM control register n (PWMCn) (n = 0, 1)	40H
		PWM buffer register n (PWMBn) (n = 0, 1)	0000H
	Interrupt/exception control functions	In-service priority register (ISPR)	00H
		External interrupt mode register n (INTMn) (n = 0 to 4)	00H
		Interrupt mask register n (IMRn) (n = 0 to 3)	FFFFH
		Valid edge select register Cn (SESCn) (n = 0 to 3)	00H
		Interrupt control registers (OVIC00 to OVIC03, P00IC0, P00IC1, P01IC0, P01IC1, P02IC0, P02IC1, P03IC0, P03IC1, P10IC0 to P10IC3, P11IC0 to P11IC3, P12IC0 to P12IC3, P13IC0 to P13IC3, CMICD0 to CMICD3, DMAIC0 to DMAIC3, CSIIC0 to CSIIC2, SEIC0 to SEIC2, SRIC0 to SRIC2, STIC0 to STIC2, ADIC)	47H
	Memory control functions	Page ROM configuration register (PRC)	7000H
		DRAM configuration register n (SCRn) (n = 1, 3, 4, 6)	3FC1H
		SDRAM configuration register n (SCRn) (n = 1, 3, 4, 6)	0000H
		Refresh control register n (RFSn) (n = 1, 3, 4, 6)	0000H
		SDRAM refresh control register n (RFSn) (n = 1, 3, 4, 6)	0000H
		Refresh wait control register (RWC)	00H

Table 15-2. Initial Value of CPU, Internal RAM, and On-Chip Peripheral I/O After Reset (3/3)

Internal Hardware		Register Name	Initial Value After Reset
On-chip peripheral I/O	DMA functions	DMA addressing control register n (DADCn) (n = 0 to 3)	0000H
		DMA byte count register n (DBCn) (n = 0 to 3)	Undefined
		DMA channel control register n (DCHCn) (n = 0 to 3)	00H
		DMA destination address register nH (DDAnH) (n = 0 to 3)	Undefined
		DMA destination address register nL (DDAnL) (n = 0 to 3)	Undefined
		DMA disable status register (DDIS)	00H
		DMA restart register (DRST)	00H
		DMA source address register nH (DSAnH) (n = 0 to 3)	Undefined
		DMA source address register nL (DSAnL) (n = 0 to 3)	Undefined
		DMA terminal count output control register (DTOC)	01H
		DMA trigger source register n (DTFRn) (n = 0 to 3)	00H
	Bus control functions	Address setup wait control register (ASC)	FFFFH
		Bus cycle control register (BCC)	FFFFH
		Bus cycle period control register (BCP)	00H
		Bus cycle type configuration register n (BCTn) (n = 0, 1)	8888H
		Endian configuration register (BEC)	0000H
		Bus size configuration register (BSC)	0000H/5555H
		Chip area select control register n (CSCn) (n = 0, 1)	2C11H
		Data wait control register n (DWCn) (n = 0, 1)	7777H
	Power-save control functions	Command register (PRCMD)	Undefined
		Power-save control register (PSC)	00H
		Clock control register (CKC)	00H
		Power-save mode register (PSMR)	00H
	System control	Peripheral command register (PHCMD)	Undefined
		Peripheral status register (PHS)	00H
		System wait control register (VSWC)	77H
		Flash programming mode control register (FLPMC)	08H/0CH/00H
		Lock register (LOCKR)	0×H

Caution "Undefined" in the above table is undefined after power-on-reset, or undefined as a result of data destruction when $\overline{\text{RESET}}$↓ is input and the data write timing has been synchronized. For other $\overline{\text{RESET}}$↓ signals, data is held in the same state it was in before the $\overline{\text{RESET}}$ operation.

CHAPTER 16 FLASH MEMORY (μPD70F3107)

The μPD70F3107 is the flash memory version of the V850E/MA1 and it has an on-chip 256 KB flash memory configured as two 128 KB areas.

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When preproducing an application set with the flash memory version and then mass producing it with the mask ROM version, be sure to conduct sufficient evaluations on the commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.

Writing to flash memory can be performed with memory mounted on the target system (on board). A dedicated flash programmer is connected to the target system to perform writing.

The following can be considered as the development environment and the applications using flash memory.

- Software can be changed after the V850E/MA1 is solder mounted on the target system.
- Small scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.

16.1 Features

- All area batch erase, or erase in block units (128 KB)
- Communication through serial interface from the dedicated flash programmer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming
- Flash memory programming by self-programming in block units (128 KB) is possible

16.2 Writing with Flash Programmer

Writing can be performed either on-board or off-board by the dedicated flash programmer.

(1) On-board programming

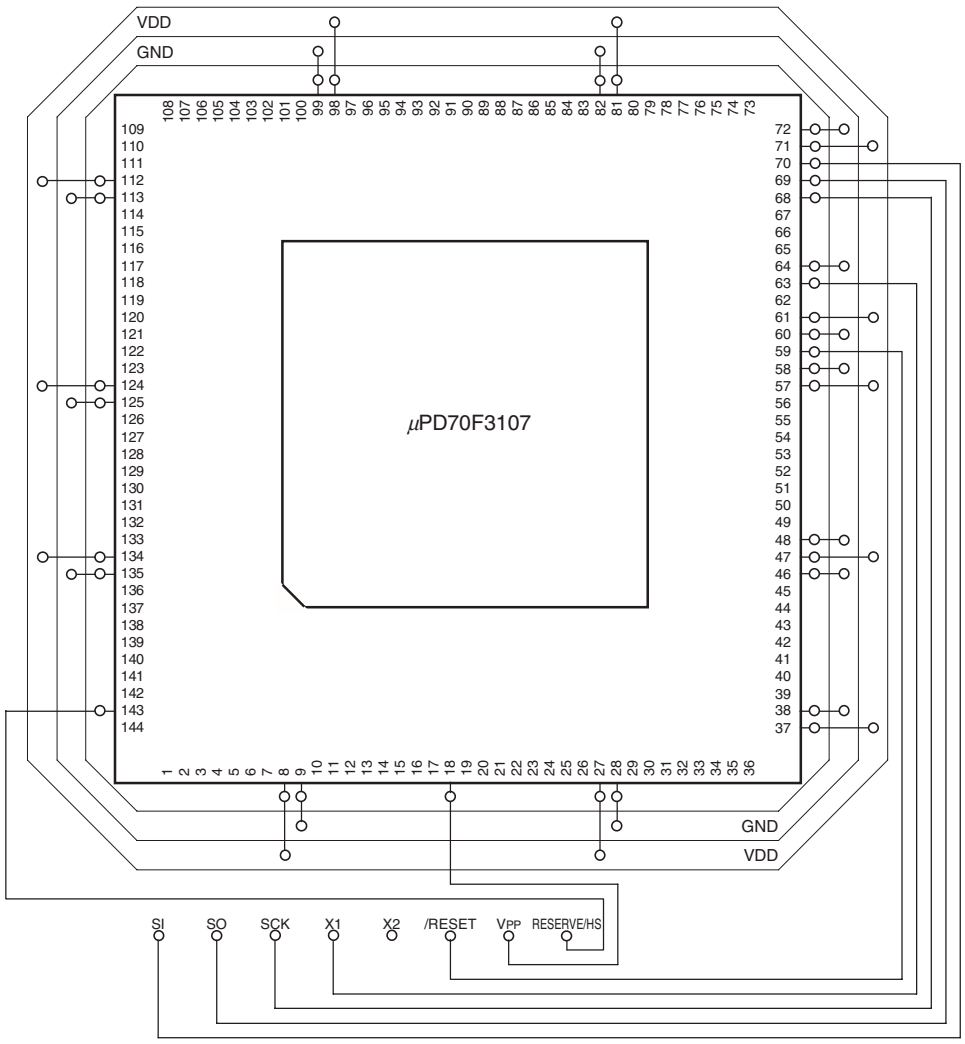
The contents of the flash memory are rewritten after the V850E/MA1 is mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

(2) Off-board programming

Writing to flash memory is performed by the dedicated program adapter (FA Series), etc., before mounting the V850E/MA1 on the target system.

Remark The FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

Figure 16-1. Connection Example of Adapter (FA-144GJ-UEN) for V850E/MA1 Flash Memory Programming



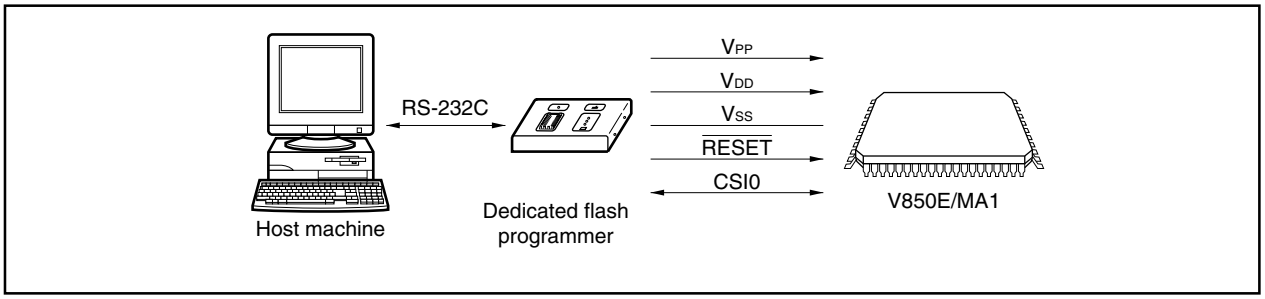
★ **Remarks 1.** Pins whose connections are not indicated should be connected according to the recommended connections of unused pins (refer to **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**).

2. When connecting to V_{DD} via a resistor, it is recommended to use a resistor of 1 kΩ to 10 kΩ.

3. This adapter is for the 144-pin plastic LQFP package.

16.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/MA1.



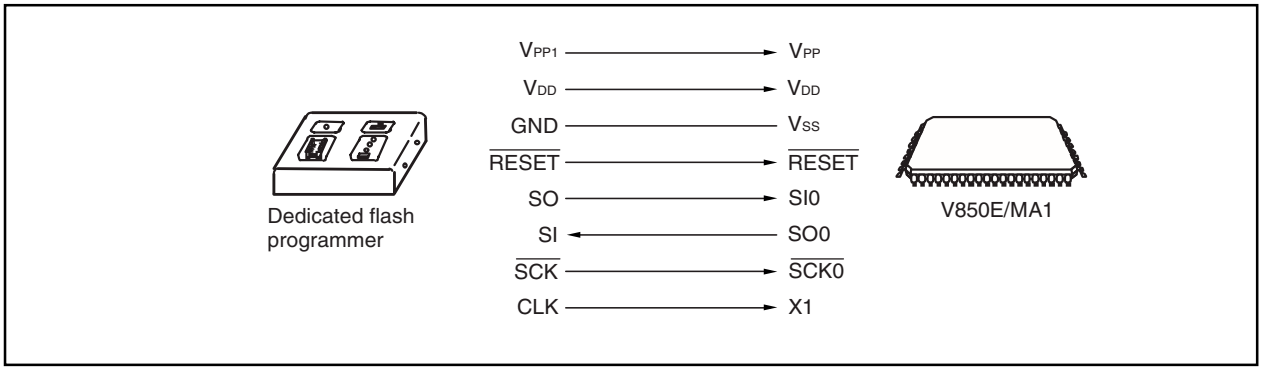
A host machine is required for controlling the dedicated flash programmer.

CSI0 is used for the interface between the dedicated flash programmer and the V850E/MA1 to perform writing, erasing, etc. A dedicated program adapter (FA Series) is required for off-board writing.

16.4 Communication Mode

(1) CSI0

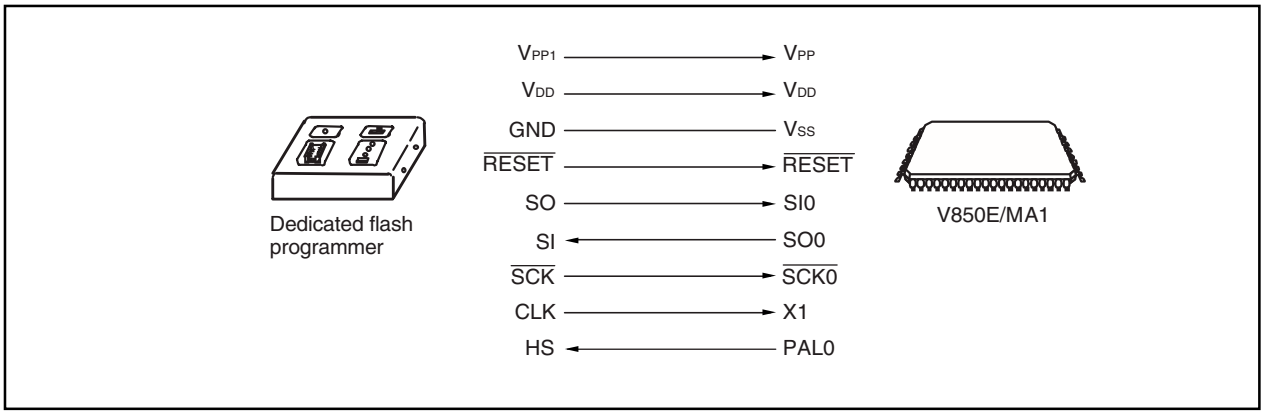
Transfer rate: Up to 2 MHz (MSB first)



The dedicated flash programmer outputs the transfer clock and the V850E/MA1 operates as a slave.

(2) Handshake-supported CSI communication

Transfer rate: Up to 2 MHz (MSB first)



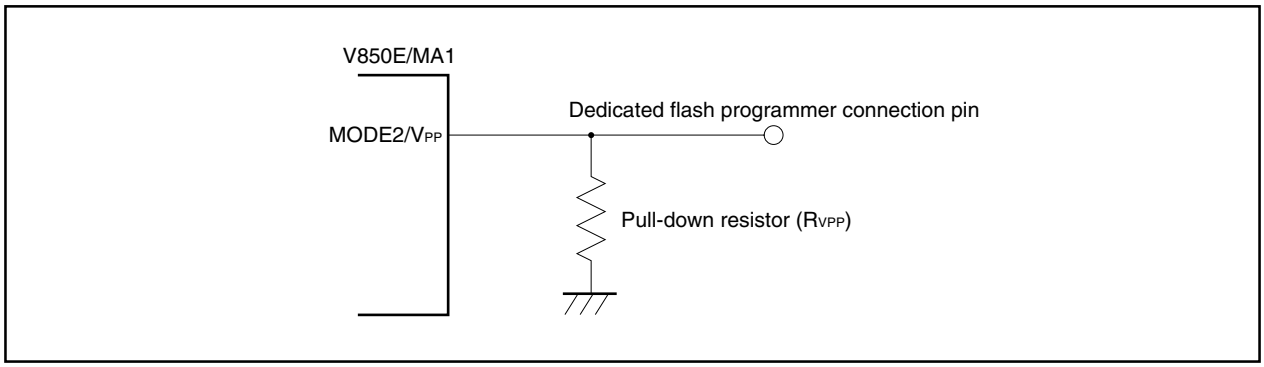
16.5 Pin Connection

When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, install a function to switch from the normal operation mode (single-chip modes 0, 1 or ROMless modes 0, 1) to the flash memory programming mode.

In the flash memory programming mode, all the pins not used for flash memory programming enter the same status as they were immediately after reset in single-chip mode 0. Therefore, because all the ports become output high-impedance, pin connection is required when the external device does not acknowledge the output high-impedance status.

16.5.1 MODE2/V_{PP} pin

In the normal operation mode, 0 V is input to the MODE2/V_{PP} pin. In the flash memory programming mode, a 7.8 V writing voltage is supplied to the MODE2/V_{PP} pin. The following shows an example of the connection of the MODE2/V_{PP} pin.



16.5.2 Serial interface pin

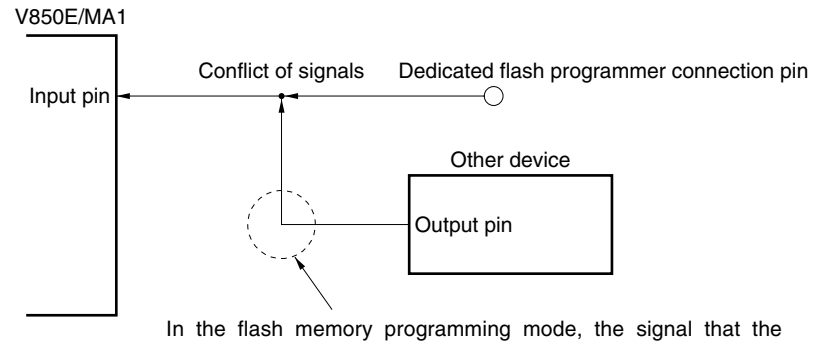
The following shows the pins used by each serial interface.

Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{SCK0}$

When connecting a dedicated flash programmer to a serial interface pin that is connected to other devices on-board, care should be taken to avoid the conflict of signals and the malfunction of other devices, etc.

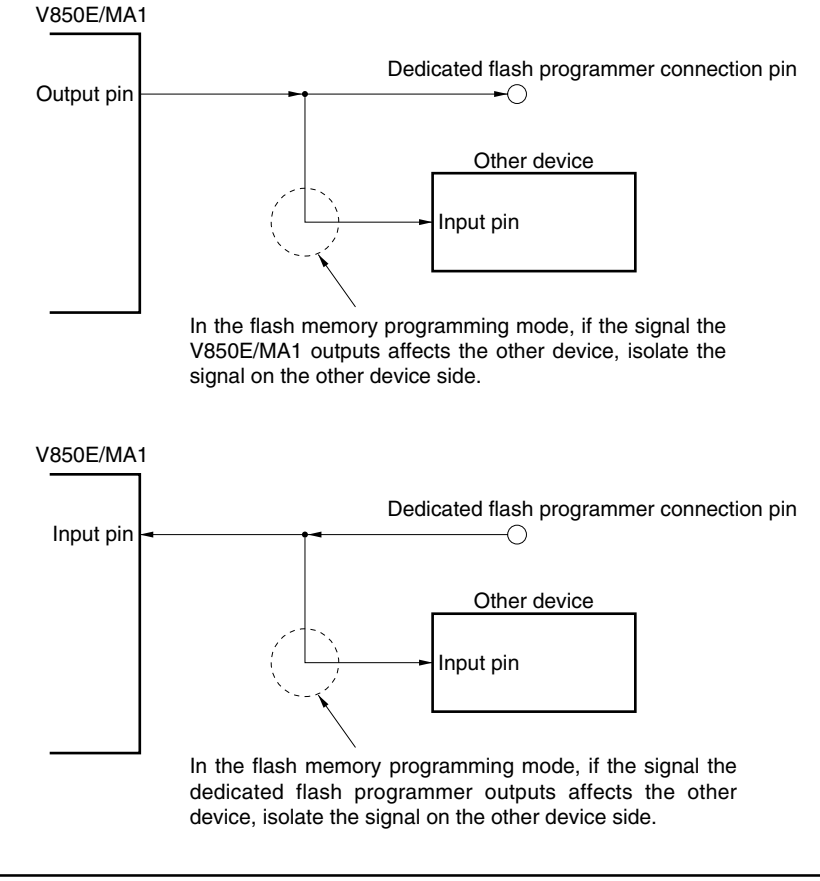
(1) Conflict of signals

When connecting a dedicated flash programmer (output) to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.



(2) Malfunction of other device

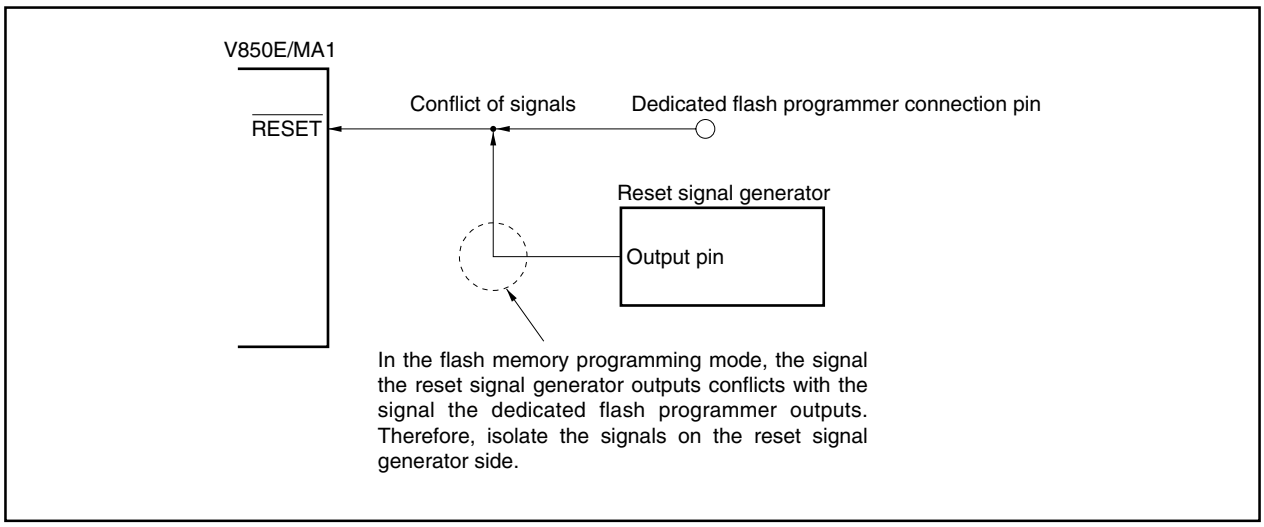
When connecting a dedicated flash programmer (output or input) to a serial interface pin (input or output) connected to another device (input), the signal output to the other device may cause the device to malfunction. To avoid this, isolate the connection to the other device or set so that the input signal to the other device is ignored.



16.5.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin, which is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When the reset signal is input from the user system in flash memory programming mode, the programming operations will not be performed correctly. Therefore, do not input signals other than the reset signal from the dedicated flash programmer.



16.5.4 NMI pin

Do not change the signal input to the NMI pin in flash memory programming mode. If it is changed in flash memory programming mode, programming may not be performed correctly.

16.5.5 MODE0 to MODE2 pins

If MODE0 is set as a high-level or low-level input and MODE1 is set as a high-level input, a write voltage (7.8 V) is applied to the MODE2/V_{PP} pin and when reset is released, these pins change to the flash memory programming mode.

16.5.6 Port pins

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. These pins must be connected according to the recommended connection of unused pins (refer to **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**).

16.5.7 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode.

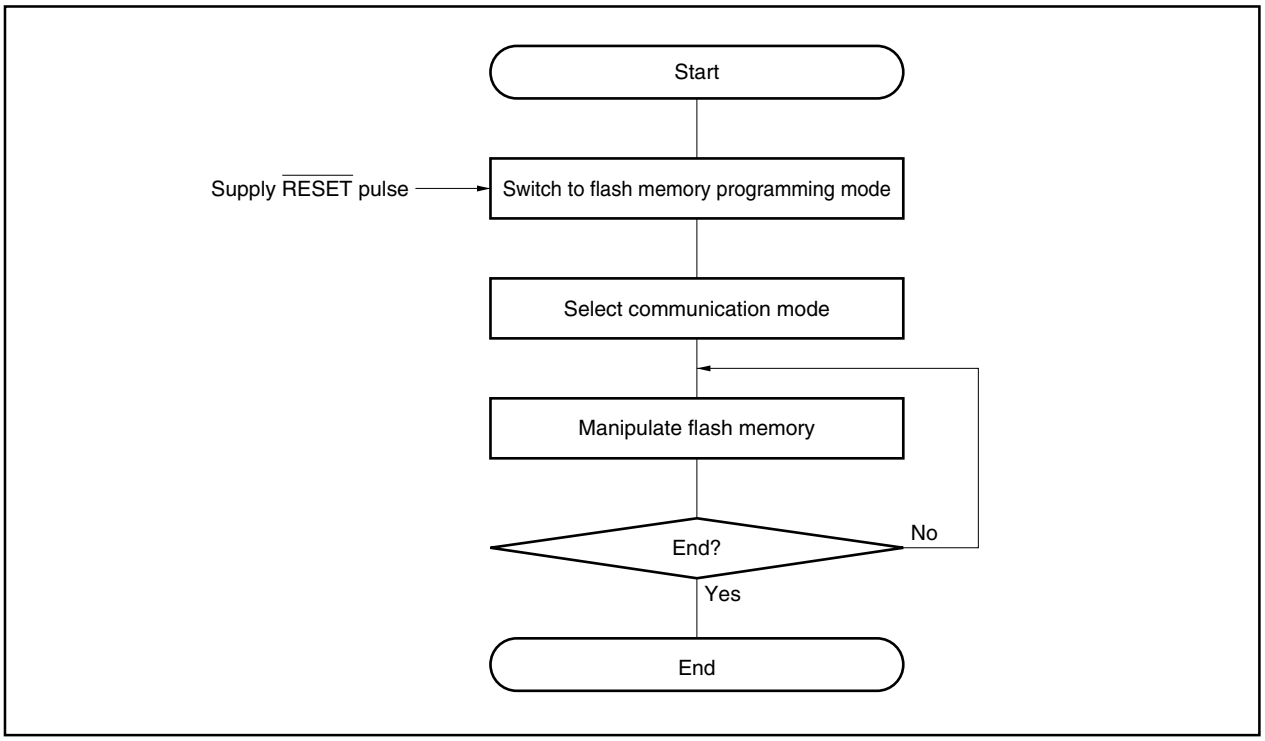
16.5.8 Power supply

Supply the power (V_{DD}, V_{SS}, AV_{DD}, AV_{REF}, AV_{SS}, CV_{DD}, and CV_{SS}) the same as when in normal operation mode. Connect V_{DD} and GND of the dedicated flash programmer to V_{DD} and V_{SS}. (V_{DD} of the dedicated flash programmer is provided with a power supply monitoring function.)

16.6 Programming Method

16.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

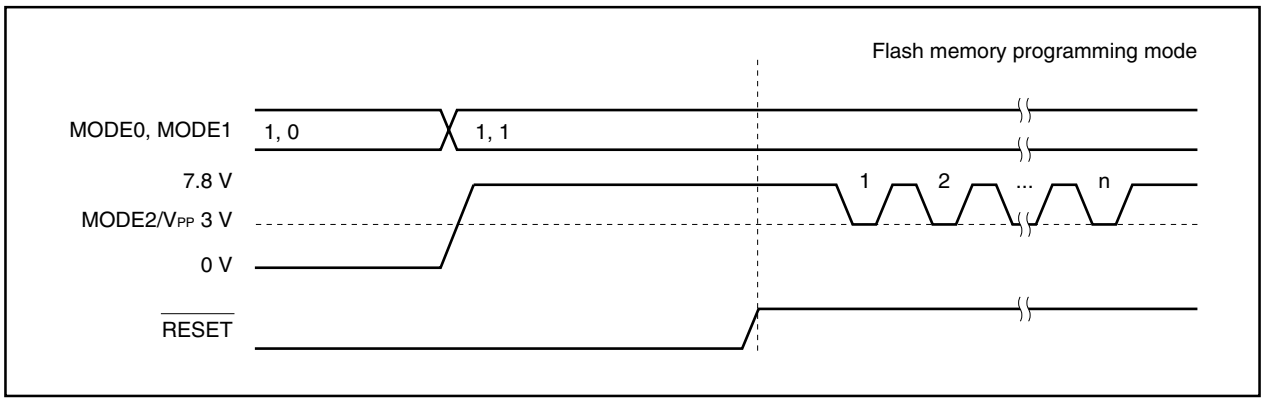


16.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850E/MA1 in the flash memory programming mode. To switch to this mode, set the MODE0 to MODE1 and MODE2/V_{PP} pins before releasing reset.

When performing on-board writing, switch modes using a jumper, etc.

- MODE0: High-level or low-level input
- MODE1: High-level input
- MODE2/V_{PP}: 7.8 V



16.6.3 Selection of communication mode

In the V850E/MA1, the communication mode is selected by inputting pulses (16 pulses max.) to the V_{PP} pin after switching to the flash memory programming mode. The V_{PP} pulse is generated by the dedicated flash programmer.

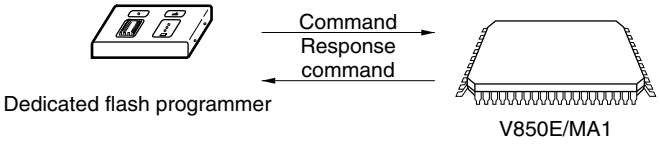
The following shows the relationship between the number of pulses and the communication mode.

Table 16-1. List of Communication Modes

V_{PP} Pulse	Communication Mode	Remarks
0	CSI0	V850E/MA1 performs slave operation, MSB first
3	Handshake-supporting CSI	
Other	RFU (reserved)	Setting prohibited

16.6.4 Communication commands

The V850E/MA1 communicates with the dedicated flash programmer by means of commands. A command sent from the dedicated flash programmer to the V850E/MA1 is called the “command”. The response signal sent from the V850E/MA1 to the dedicated flash programmer is called the “response command”.



The following shows the commands for controlling the flash memory of the V850E/MA1. All of these commands are issued from the dedicated flash programmer, and the V850E/MA1 performs the various processing corresponding to the commands.

Category	Command Name	Function
Verify	Batch verify command	Compares the contents of the entire memory and the input data.
	Block verify command	Compares the contents of the specified memory block and the input data.
Erase	Batch erase command	Erases the contents of the entire memory.
	Block erase command	Erases the contents of the specified memory block.
	Write back command	Writes back the contents which were erased.
Blank check	Batch blank check command	Checks the erase state of the entire memory.
	Block blank check command	Checks the erase state of the specified memory block.
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 oscillation frequency.
	Erasure time setting command	Sets the erasing time of batch erase.
	Write time setting command	Sets the writing time of data write.
	Write back time setting command	Sets the write back time.
	Silicon signature command	Reads outs the silicon signature information.
	Reset command	Escapes from each state.

The V850E/MA1 sends back response commands for the commands issued from the dedicated flash programmer. The following shows the response commands the V850E/MA1 sends out.

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

16.7 Flash Memory Programming by Self-Programming

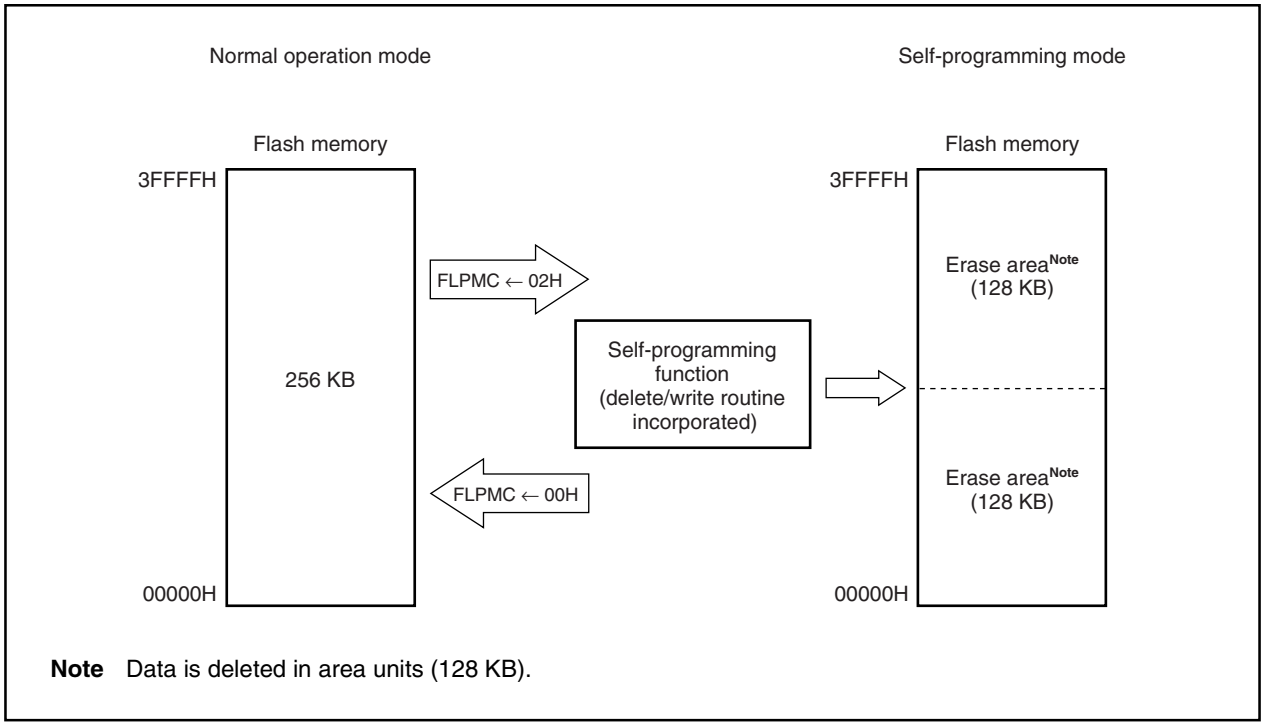
The μPD70F3107 supports a self-programming function to rewrite the flash memory using a user program. By using this function, the flash memory can be rewritten with a user application. This self-programming function can be also used to upgrade the program in the field.

16.7.1 Outline of self-programming

Self-programming implements erasure and writing of the flash memory by calling the self-programming function (device's internal processing) on the program placed in the block 0 space (000000H to 1FFFFFH) and areas other ★ than internal ROM area. To place the program in the block 0 space and internal ROM area, copy the program to areas other than 000000H to 1FFFFFH (e.g. internal RAM area) and execute the program to call the self-programming function.

To call the self-programming function, change the operating mode from normal mode to self-programming mode using the flash programming mode control register (FLPMC).

Figure 16-2. Outline of Self-Programming



Note Data is deleted in area units (128 KB).

16.7.2 Self-programming function

The μPD70F3107 provides self-programming functions, as shown below. By combining these functions, erasing/writing flash memory becomes possible.

Table 16-2. Function List

Type	Function Name	Function
Erase	Area erase	Erases the specified area.
Write	Continuous write in word units	Continuously writes the specified memory contents from the specified flash memory address, for the number of words specified in 4-byte units.
	Prewrite	Writes 0 to flash memory before erasure.
Check	Erase verify	Checks whether an over erase occurred after erasure.
	Erase byte verify	Checks whether erasure is complete.
	Internal verify	Checks whether the signal level of the post-write data in flash memory is appropriate.
Write back	Area write back	Writes back the flash memory area in which an over erase occurred.
Acquire information	Flash memory information read	Reads out information about flash memory.

16.7.3 Outline of self-programming interface

To execute self-programming using the self-programming interface, the environmental conditions of the hardware and software for manipulating the flash memory must be satisfied.

It is assumed that the self-programming interface is used in an assembly language.

(1) Entry program

This program is to call the internal processing of the device.

It is a part of the application program, and must be executed in memory other than the block 0 space and internal ROM area (flash memory).

(2) Device internal processing

This is manipulation of the flash memory executed inside the device.

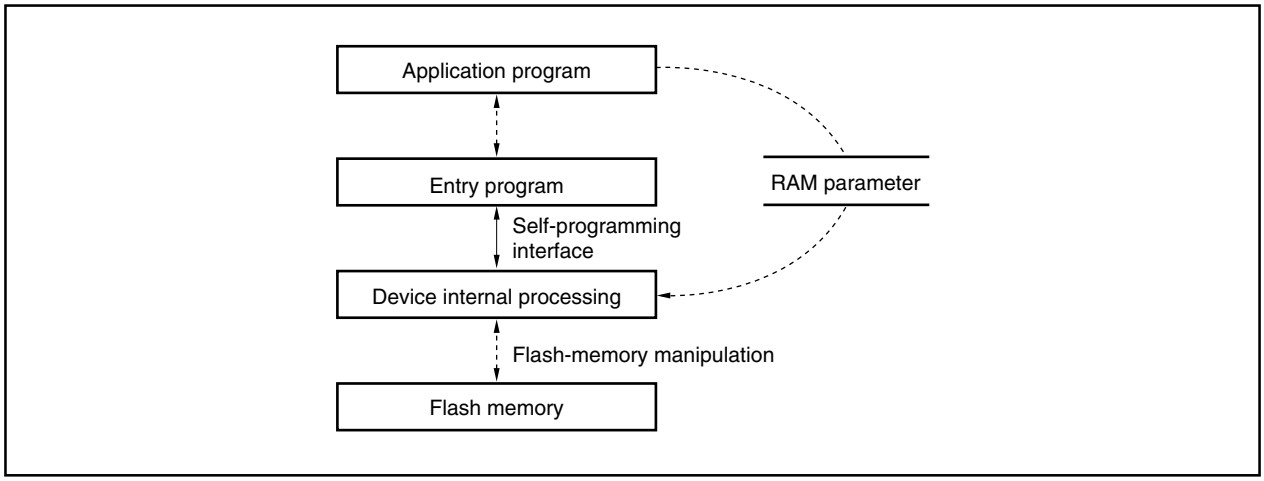
This processing manipulates the flash memory after it has been called by the entry program.

(3) RAM parameter

This is a RAM area to which the parameters necessary for self-programming, such as write time and erase time, are written. It is set by the application program and referenced by the device internal processing.

The self-programming interface is outlined below.

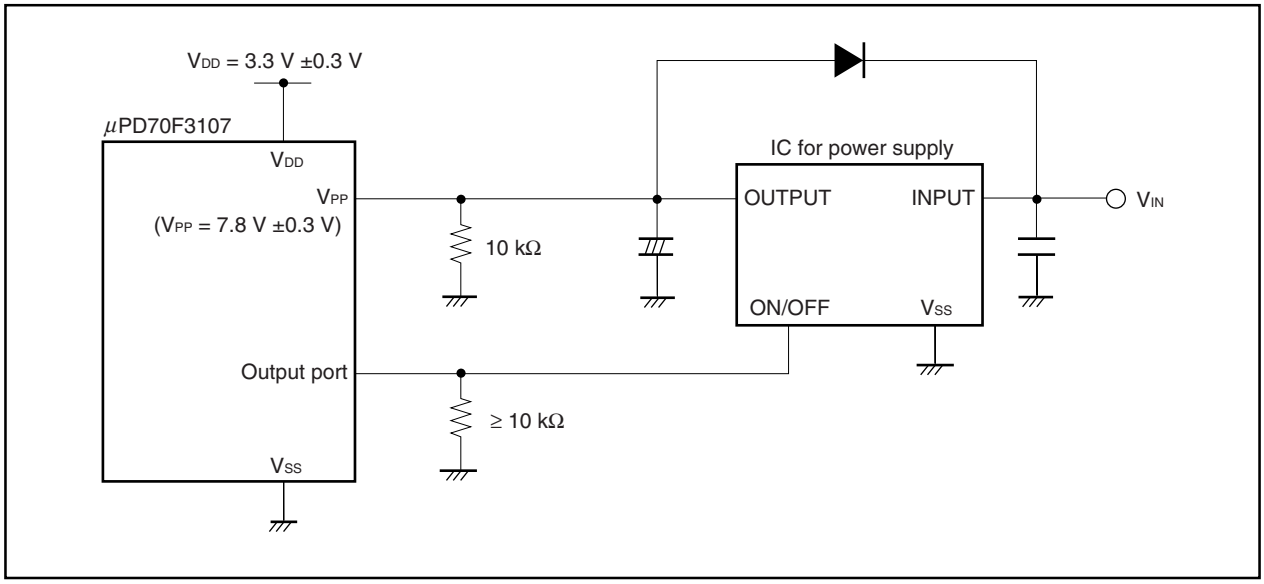
Figure 16-3. Outline of Self-Programming Interface



16.7.4 Hardware environment

To write or erase the flash memory, a high voltage must be applied to the V_{PP} pin. To execute self-programming, a circuit that can generate a write voltage (V_{PP}) and that can be controlled by software is necessary on the application system. An example of a circuit that can select a voltage to be applied to the V_{PP} pin by manipulating a port is shown below.

Figure 16-4. Example of Self-Programming Circuit Configuration



The voltage applied to the V_{PP} pin must satisfy the following conditions:

- Hold the voltage applied to the V_{PP} pin at 0 V in the normal operation mode and hold the V_{PP} voltage only while the flash memory is being manipulated.
- The V_{PP} voltage must be stable from before manipulation of the flash memory starts until manipulation is complete.

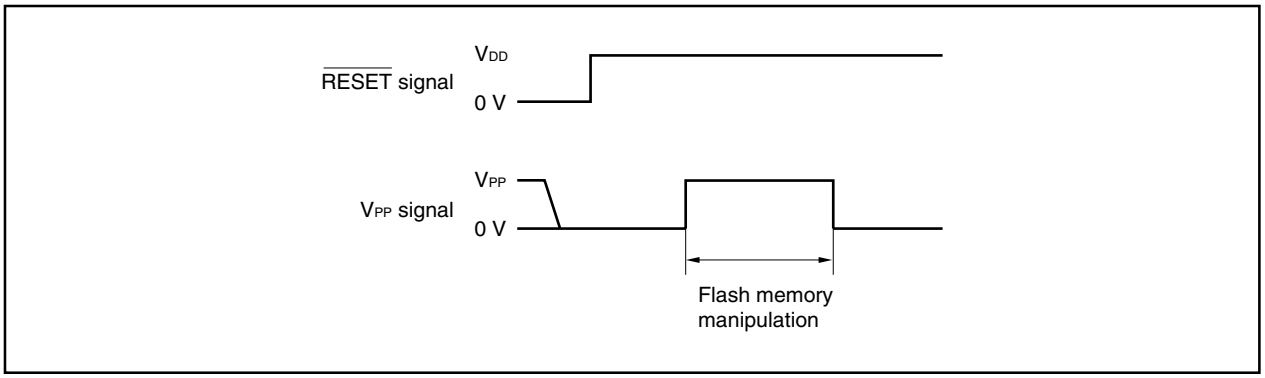
Cautions 1. Apply 0 V to the V_{PP} pin when reset is released.

2. Implement self-programming in single-chip mode 0 or 1.

3. Apply the voltage to the V_{PP} pin in the entry program.

4. If both writing and erasing are executed by using the self-programming function and flash memory programmer on the target board, be sure to communicate with the programmer using CSI0 (do not use the handshake-supporting CSI).

Figure 16-5. Timing to Apply Voltage to V_{PP} Pin



16.7.5 Software environment

The following conditions must be satisfied before using the entry program to call the device internal processing.

Table 16-3. Software Environmental Conditions

Item	Description
Location of entry program	Execute the entry program in memory other than the block 0 space and flash memory area. The device internal processing cannot be directly called by the program that is executed on the flash memory.
Execution status of program	The device internal processing cannot be called while an interrupt is being serviced (NP bit of PSW = 1, ID bit of PSW = 1).
Masking interrupts	Mask all the maskable interrupts used. Mask each interrupt by using the corresponding interrupt control register. To mask a maskable interrupt, be sure to specify masking by using the corresponding interrupt control register. Mask the maskable interrupt even when the ID bit of the PSW = 1 (interrupts are disabled).
Manipulation of V_{PP} voltage	Stabilize the voltage applied to the V_{PP} pin (V_{PP} voltage) before starting manipulation of the flash memory. After completion of the manipulation, return the voltage of the V_{PP} pin to 0 V.
Initialization of internal timer	Do not use the internal timer while the flash memory is being manipulated. Because the internal timer is initialized after the flash memory has been used, initialize the timer with the application program to use the timer again.
Stopping reset signal input	Do not input the reset signal while the flash memory is being manipulated. If the reset signal is input while the flash memory is being manipulated, the contents of the flash memory under manipulation become undefined.
Stopping NMI signal input	Do not input the NMI signal while the flash memory is being manipulated. If the NMI signal is input while the flash memory is being manipulated, the flash memory may not be correctly manipulated by the device internal processing. If an NMI occurs while the device internal processing is in progress, the occurrence of the NMI is reflected in the NMI flag of the RAM parameter. If manipulation of the flash memory is affected by the occurrence of the NMI, the function of each self-programming function is reflected in the return value.
Reserving stack area	The device internal processing takes over the stack used by the user program. It is necessary that an area of 300 bytes be reserved for the stack size of the user program when the device internal processing is called. r3 is used as the stack pointer.
Saving general-purpose registers	The device internal processing rewrites the contents of r6 to r14, r20, and r31 (lp). Save and restore these register contents as necessary.

16.7.6 Self-programming function number

To identify a self-programming function, the following numbers are assigned to the respective functions. These function numbers are used as parameters when the device internal processing is called.

Table 16-4. Self-Programming Function Number

Function No.	Function Name
0	Acquiring flash information
1	Erasing area
2 to 4	RFU
5	Area write back
6 to 8	RFU
9	Erase byte verify
10	Erase verify
11 to 15	RFU
16	Successive write in word units
17 to 19	RFU
20	Pre-write
21	Internal verify
Other	Prohibited

Remark RFU: Reserved for Future Use

16.7.7 Calling parameters

The arguments used to call the self-programming function are shown in the table below. In addition to these arguments, parameters such as the write time and erase time are set to the RAM parameters indicated by ep (r30).

Table 16-5. Calling Parameters

Function Name	First Argument (r6) Function No.	Second Argument (r7)	Third Argument (r8)	Fourth Argument (r9)	Return Value (r10)
Acquiring flash information	0	Option number[Note 1]	–	–	**Note 1**
Erasing area	1	Area erase start address	–	–	0: Normal completion Other than 0: Error
Area write back	5	None (acts on erase manipulation area immediately before)	–	–	None
Erase byte verify	9	Verify start address	Number of bytes to be verified	–	0: Normal completion Other than 0: Error
Erase verify	10	None (acts on erase manipulation area immediately before)	–	–	0: Normal completion Other than 0: Error
Successive write in word units[Note 2]	16	Write start address[Note 3]	Start address of write source data[Note 3]	Number of words to be written (word units)	0: Normal completion Other than 0: Error
Pre-write	20	Write start address	Number of bytes to be written	–	0: Normal completion Other than 0: Error
Internal verify	21	Verify start address	Number of bytes to be verified	–	0: Normal completion Other than 0: Error

Notes 1. See **16.7.10 Flash information** for details.

2. Prepare write source data in memory other than the flash memory when data is written successively in word units.

3. This address must be at a 4-byte boundary.

Caution For all the functions, ep (r30) must indicate the first address of the RAM parameter.

16.7.8 Contents of RAM parameters

Reserve the following 48-byte area in the internal RAM or external RAM for the RAM parameters, and set the parameters to be input. Set the base addresses of these parameters to ep (r30).

Table 16-6. Description of RAM Parameter

Address	Size	I/O	Description
ep+0	4 bytes	–	For internal operations
ep+4:Bit 5[Note 1]	1 bit	Input	Operation flag (Be sure to set this flag to 1 before calling the device internal processing.) 0: Normal operation in progress 1: Self-programming in progress
ep+4:Bit 7[Notes 2, 3]	1 bit	Output	NMI flag 0: NMI not detected 1: NMI detected
ep+8	4 bytes	Input	Erase time (unsigned 4 bytes) Expressed as 1 count value in units of the internal operation unit time (100 μs). Set value = Erase time (μs)/internal operation unit time (μs) Example: If erase time is 0.4 s → 0.4 × 1,000,000/100 = 4,000 (integer operation)
ep+0xc	4 bytes	Input	Write back time (unsigned 4 bytes) Expressed as 1 count value in units of the internal operation unit time (100 μs). Set value = Write back time (μs)/internal operation unit time (μs) Example: If write back time is 1 ms → 1 × 1,000/100 = 10 (integer operation)
ep+0x10	2 bytes	Input	Timer set value for creating internal operation unit time (unsigned 2 bytes) Write a set value that makes the value of timer D the internal operation unit time (100 μs). Set value = Operating frequency (Hz)/1,000,000 × Internal operation unit time (μs)/ Timer division ratio (4) + 1[Note 4] Example: If the operating frequency is 50 MHz → 50,000,000/1,000,000 × 100/4 + 1 = 1,251 (integer operation)
ep+0x12	2 bytes	Input	Timer set value for creating write time (unsigned 2 bytes) Write a set value that makes the value of timer D the write time. Set value = Operating frequency (Hz)/Write time (μs)/Timer division ratio (4) + 1[Note 4] Example: If the operating frequency is 50 MHz and the write time is 20 μs → 50,000,000/1,000,000 × 20/4 + 1 = 251 (integer operation)
ep+0x14	28 bytes	–	For internal operations

Notes 1. Fifth bit of address of ep+4 (least significant bit is bit 0.)

2. Seventh bit of address of ep+4 (least significant bit is bit 0.)

3. Clear the NMI flag by the user program because it is not cleared by the device internal processing.

4. The device internal processing sets this value minus 1 to the timer. Because the fraction is rounded up, add 1 as indicated by the expression of the set value.

Caution Be sure to reserve the RAM parameter area at a 4-byte boundary.

16.7.9 Errors during self-programming

The following errors related to manipulation of the flash memory may occur during self-programming. An error occurs if the return value (r10) of each function is not 0.

Table 16-7. Errors During Self-Programming

Error	Function	Description
Overerase error	Erase verify	Excessive erasure occurs.
Undererase error (blank check error)	Erase byte verify	Erasure is insufficient. Additional erase operation is needed.
Verify error	Successive writing in word units	The written data cannot be correctly read. Either an attempt has been made to write to flash memory that has not been erased, or writing is not sufficient.
Internal verify error	Internal verify	The written data is not at the correct signal level.

Caution The overerase error and undererase error may simultaneously occur in the entire flash memory.

16.7.10 Flash information

For the flash information acquisition function (function No. 0), the option number (r7) to be specified and the contents of the return value (r10) are as follows. To acquire all flash information, call the function as many times as required in accordance with the format shown below.

★ **Table 16-8. Flash Information**

Option No. (r7)	Return Value (r10)
0	Specification prohibited
1	Specification prohibited
2	Bit representation of return value (MSB: bit 31) FFFFFFFFFFFFFFFFAAAAAAAAFFFFFFFF (LSB: bit 0) Bits 31 to 16: FFFFFFFFFFFFFFFF (reserved for future use) Mask bits 31 to 16 because they are not normally 0. Bits 15 to 8: AAAAAAAA (number of areas) (unsigned 8 bits) Bits 7 to 0: FFFFFFFF (reserved for future use) Mask bits 7 to 0 because they are not normally 0.
3+0	End address of area 0
3+1	End address of area 1

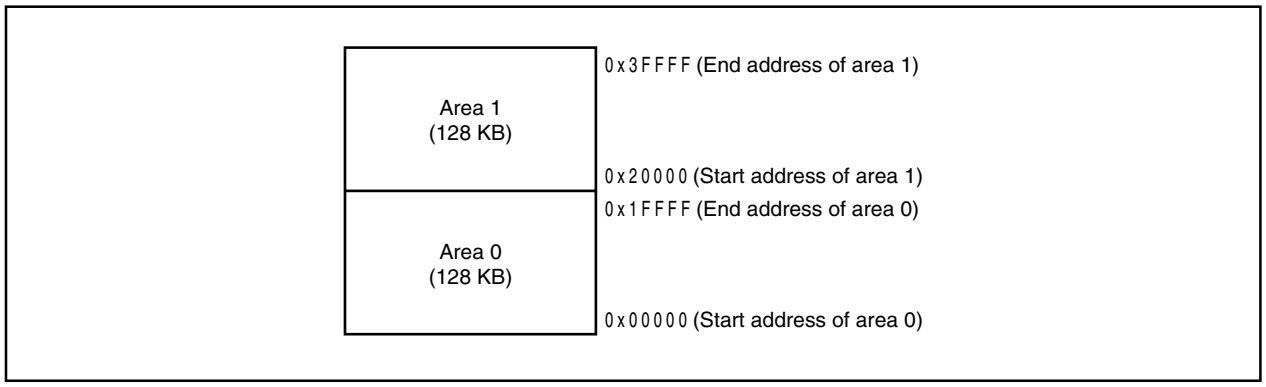
Cautions 1. The start address of area 0 is 0. The "end address + 1" of the preceding area is the start address of the next area.

2. The flash information acquisition function does not check values such as the maximum number of areas specified by the argument of an option. If an illegal value is specified, an undefined value is returned.

16.7.11 Area number

The area numbers and memory map of the μPD70F3107 are shown below.

Figure 16-6. Area Configuration



16.7.12 Flash programming mode control register (FLPMC)

The flash memory mode control register (FLPMC) is a register used to enable/disable writing to flash memory and to specify the self-programming mode.

This register can be read/written in 8-bit or 1-bit units (the VPP bit (bit 2) is read-only).

Cautions 1. Be sure to transfer control to the internal RAM or external memory beforehand to manipulate the FLSPM bit. However, in on-board programming mode set by the flash programmer, the specification of FLSPM bit is ignored.

2. Do not change the initial value of bits 0 and 4 to 7.

	7	6	5	4	<3>	<2>	<1>	0	Address	After reset[Note]
FLPMC	0	0	0	0	VPPDIS	VPP	FLSPM	0	FFFFF8D4H	08H/0CH/00H

Note 08H: When writing voltage is not applied to the V_{PP} pin
0CH: When writing voltage is applied to the V_{PP} pin
00H: Product not provided with flash memory (μPD703103, 703105, 703106, 703107)

Bit position	Bit name	Function
3	VPPDIS	V_{PP} Disable Enables/disables writing/deleting on-chip flash memory. When this bit is 1, writing/deleting on-chip flash memory is disabled even if a high voltage is applied to the V_{PP} pin. 0: Enables writing/deleting flash memory 1: Disables writing/deleting flash memory
2	VPP	V_{PP} Indicates the voltage applied to the V_{PP} pin reaches the writing-enabled level. This bit is used to check whether writing is possible or not in the self-programming mode. 0: Indicates high-voltage application is not detected (the voltage has not reached the writing voltage enable level) 1: Indicates high-voltage application is detected (the voltage has reached the writing voltage enable level)
1	FLSPM	Flash Self Programming Mode Controls switching between internal ROM and the self-programming interface. This bit can switch the mode between the normal mode set by the mode pin on the application system and the self-programming mode. The setting of this bit is valid only if the voltage applied to the V_{PP} pin reaches the writing voltage enable level. 0: Normal mode (for all addresses, instruction fetch is performed from on-chip flash memory) 1: Self-programming mode (device internal processing is started.)

Setting data to the flash programming mode control register (FLPMC) is performed in the following sequence.

<1> Disable interrupts (set the NP bit and ID bit of the PSW to 1)
<2> Prepare the data to be set in the specific register in a general-purpose register
<3> Write data to the peripheral command register (PHCMD)
<4> Set the flash memory programming mode control register (FLPMC) by executing the following instructions
- Store instruction (ST/SST instructions)
- Bit manipulation instruction (SET1/CLR1/NOT1 instructions)

<5> Insert NOP instructions (5 instructions <5> to <9>)
<10> Cancel the interrupt disabled state (reset the NP bit of the PSW to 0)

[Description example]

```
<1> LDSR  rX, 5
<2> MOV   0x02, r10
<3> ST.B  r10, PHCMD [r0]
<4> ST.B  r10, FLPMC [r0]
<5> NOP
<6> NOP
<7> NOP
<8> NOP
<9> NOP
<10>LDSR  rY, 5
```

Remark rX: Value written to the PSW
rY: Value returned to the PSW

No special sequence is required for reading a specific register.

★ **Cautions 1. If an interrupt is acknowledged between when PHCMD is issued (<3>) and writing to a specific register (<4>) immediately after issuing PHCMD, writing to the specific register may not be performed and a protection error may occur (the PRERR bit of the PHS register = 1). Therefore, set the NP bit of the PSW to 1 (<1>) to disable interrupt acknowledgment. Similarly, disable acknowledgement of interrupts when a bit manipulation instruction is used to set a specific register.**

2. Use the same general-purpose register used to set a specific register (<3>) for writing to the PHCMD register (<4>) even though the data written to the PHCMD register is dummy data. This is the same as when a general-purpose register is used for addressing.

3. Do not use DMA transfer for writing to the PHCMD register and a specific register.

16.7.13 Calling device internal processing

This section explains the procedure to call the device internal processing from the entry program.

Before calling the device internal processing, make sure that all the conditions of the hardware and software environments are satisfied and that the necessary arguments and RAM parameters have been set. Call the device internal processing by setting the FLSPM bit of the flash programming mode control register (FLPMC) to 1 and then executing the trap 0x1f instruction. The processing is always called using the same procedure. It is assumed that the program of this interface is described in an assembly language.

<1> Set the FLPMC register as follows:
- VPPDIS bit = 0 (to enable writing/erasing flash memory)
- FLSPM bit = 1 (to select self-programming mode)

<2> Clear the NP bit of the PSW to 0 (to enable NMIs (only when NMIs are used on the application)).
<3> Execute trap 0x1f to transfer the control to the device's internal processing.
<4> Set the NP bit and ID bit of the PSW to 1 (to disable all interrupts).
<5> Set the value to the peripheral command register (PHCMD) that is to be set to the FLPMC register.
<6> Set the FLPMC register as follows:
- VPPDIS bit = 1 (to disable writing/erasing flash memory)
- FLSPM bit = 0 (to select normal operation mode)

<7> Wait for the internal manipulation setup time (see **16.7.13 (5) Internal manipulation setup parameter**).

(1) Parameter

r6: First argument (sets a self-programming function number)
r7: Second argument
r8: Third argument
r9: Fourth argument
ep: First address of RAM parameter

(2) Return value

r10: Return value (return value from device internal processing of 4 bytes)
ep+4:Bit 7: NMI flag (flag indicating whether an NMI occurred while the device internal processing was being executed)
0: NMI did not occur while device internal processing was being executed.
1: NMI occurred while device internal processing was being executed.

If an NMI occurs while control is being transferred to the device internal processing, the NMI request may never be reflected. Because the NMI flag is not internally reset, this bit must be cleared before calling the device internal processing. After the control returns from the device internal processing, NMI dummy processing can be executed by checking the status of this flag using software.

(3) Description

Transfer control to the device internal processing specified by a function number using the trap instruction. To do this, the hardware and software environmental conditions must be satisfied. Even if trap 0x1f is used in the user application program, trap 0x1f is treated as another operation after the FLPMC register has been set. Therefore, use of the trap instruction is not restricted on the application.

(4) Program example

An example of a program in which the entry program is executed as a subroutine is shown below. In this example, the return address is saved to the stack and then the device internal processing is called. This program must be located in memory other than the block 0 space and flash memory area.

```
ISETUP     130                              -- Internal manipulation setup parameter
EntryProgram:
   add         -4, sp                       -- Prepare
   st.w        lp, 0[sp]                    -- Save return address
   movea       lo(0x00a0), r0, r10          --
   ldsr        r10, 5                       -- PSW = NP, ID
   mov         lo(0x0002), r10              --
   st.b        r10, PHCMD[r0]               -- PHCMD = 2
   st.b        r10, FLPMC[r0]               -- VPPDIS = 0, FLSPM = 1
   nop
   nop
   nop
   nop
   nop
   movea       lo(0x0020), r0, r10          --
   ldsr        r10, 5                       -- PSW = ID
   trap        0x1f                         -- Device Internal Process
   movea       lo(0x00a0), r0, r6           --
   ldsr        r6, 5                        -- PSW = NP, ID
   mov         lo(0x08), r6
   st.b        r6, PHCMD[r0]                -- PRCMD = 8
   st.b        r6, FLPMC[r0]                -- VPPDIS = 1, FLSPM = 0
   nop
   nop
   nop
   nop
   nop
   mov         ISETUP, lp                   -- loop time = 130
loop:
   divh        r6, r6                       -- To kill time
   add         -1, lp                       -- Decrement counter
   jne         loop                         --
   ld.w        0[sp], lp                    -- Reload lp
   add         4, sp                        -- Dispose
   jmp         [lp]                         -- Return to caller
```

(5) Internal manipulation setup parameter

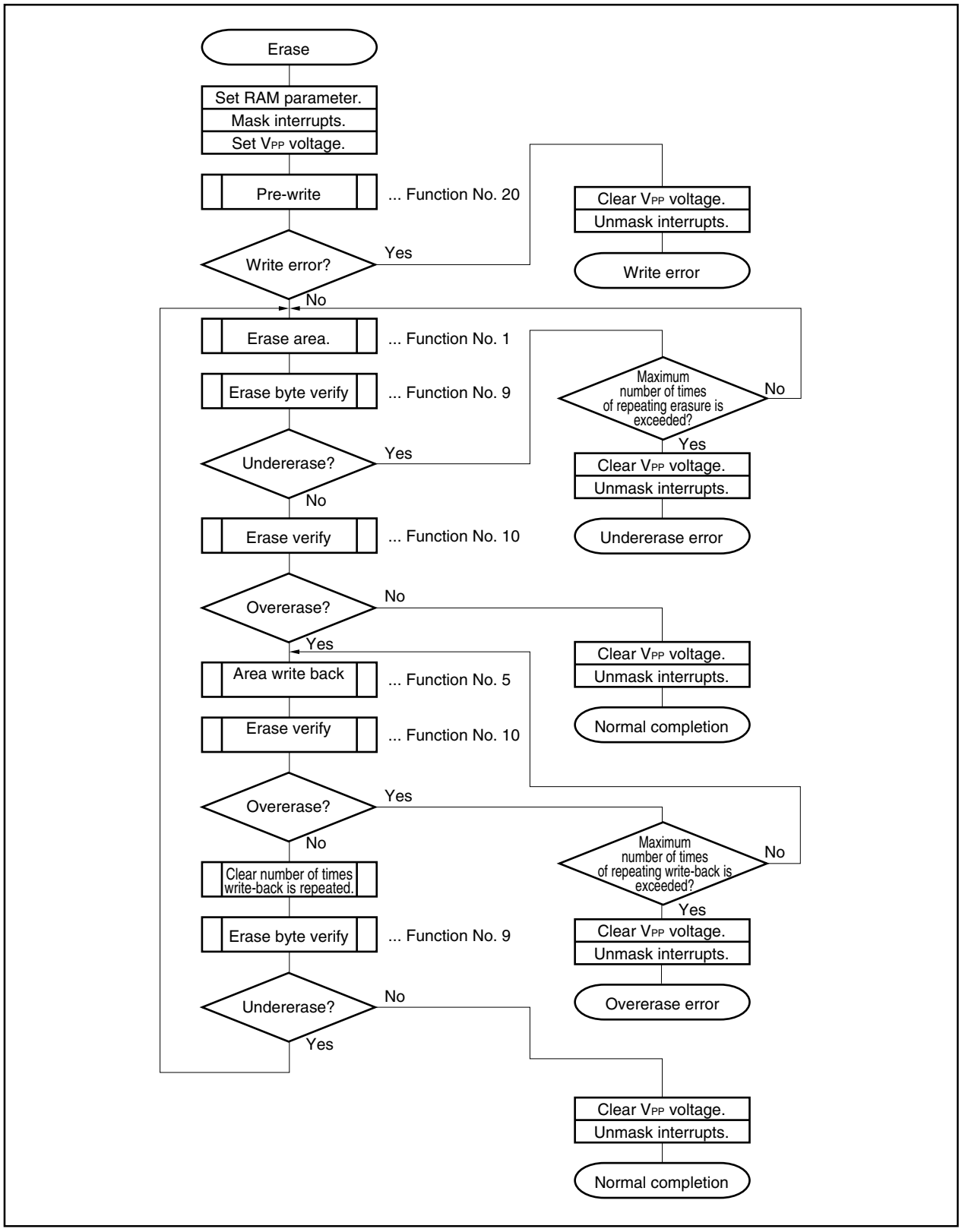
If the self-programming mode is switched to the normal operation mode, the μPD70F3107 must wait for 100 μs before it accesses the flash memory. In the program example in (4) above, the elapse of this wait time is ensured by setting ISETUP to "130" (@ 50 MHz operation). The total number of execution clocks in this example is 39 clocks (divh instruction (35 clocks) + add instruction (1 clock) + jne instruction (3 clocks)). Ensure that a wait time of 100 μs elapses by using the following expression.

39 clocks (total number of execution clocks) × 20 ns (@ 50 MHz operation) × 130 (ISETUP) = 101.4 μs (wait time)

16.7.14 Erasing flash memory flow

The procedure to erase the flash memory is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

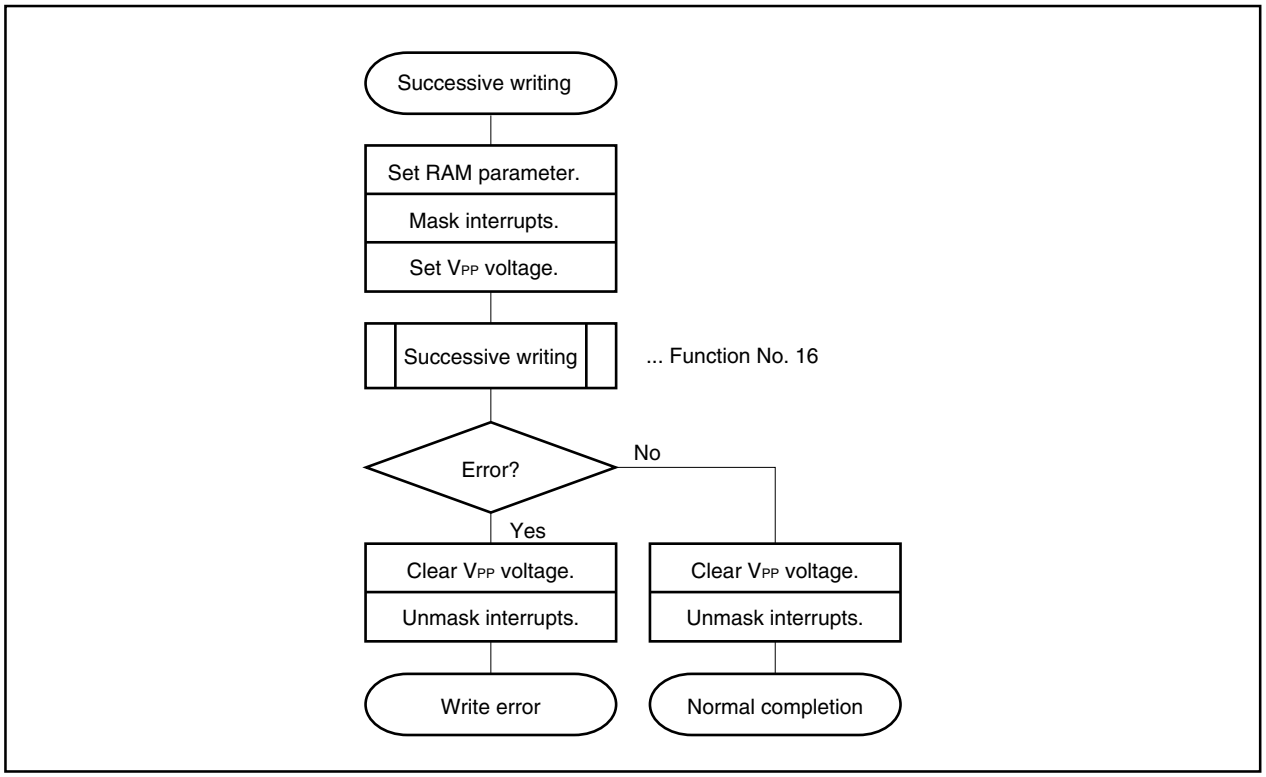
Figure 16-7. Erasing Flash Memory Flow



16.7.15 Successive writing flow

The procedure to write data all at once to the flash memory by using the function to successively write data in word units is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

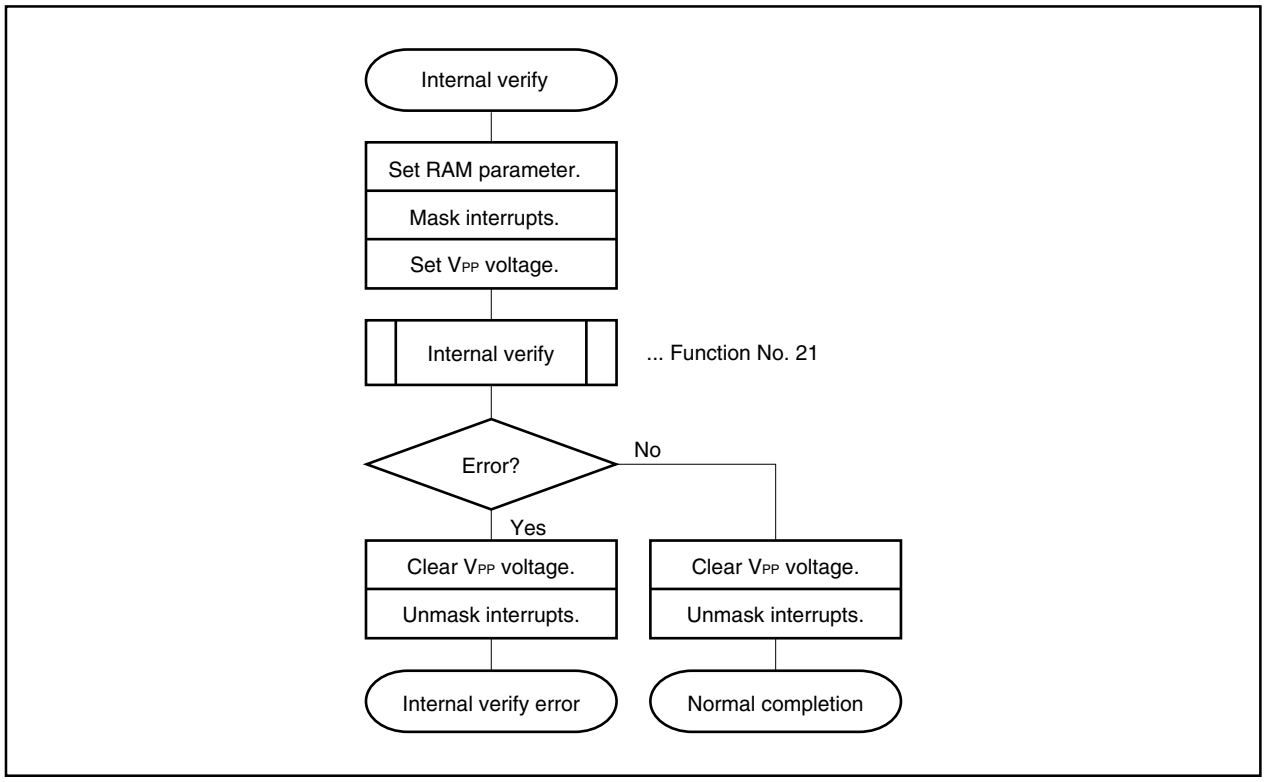
Figure 16-8. Successive Writing Flow



16.7.16 Internal verify flow

The procedure of internal verification is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

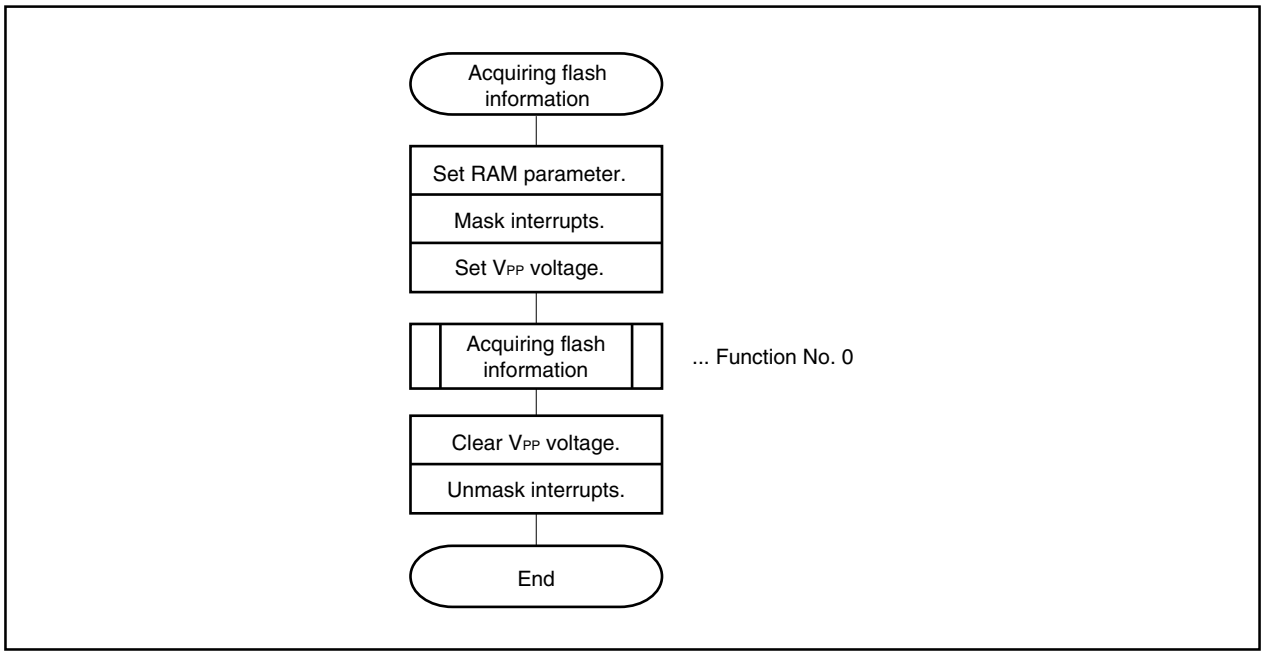
Figure 16-9. Internal Verify Flow



16.7.17 Acquiring flash information flow

The procedure to acquire the flash information is illustrated below. The processing of each function number must be executed in accordance with the specified calling procedure.

Figure 16-10. Acquiring Flash Information Flow



16.7.18 Self-programming library

An application note for the flash memory self-programming library is available for reference when executing self-programming.

In this application note, the library uses the self-programming interface of the V850 Family and can be used in C as a utility and as part of the application program. To use the library, thoroughly evaluate it on the application system.

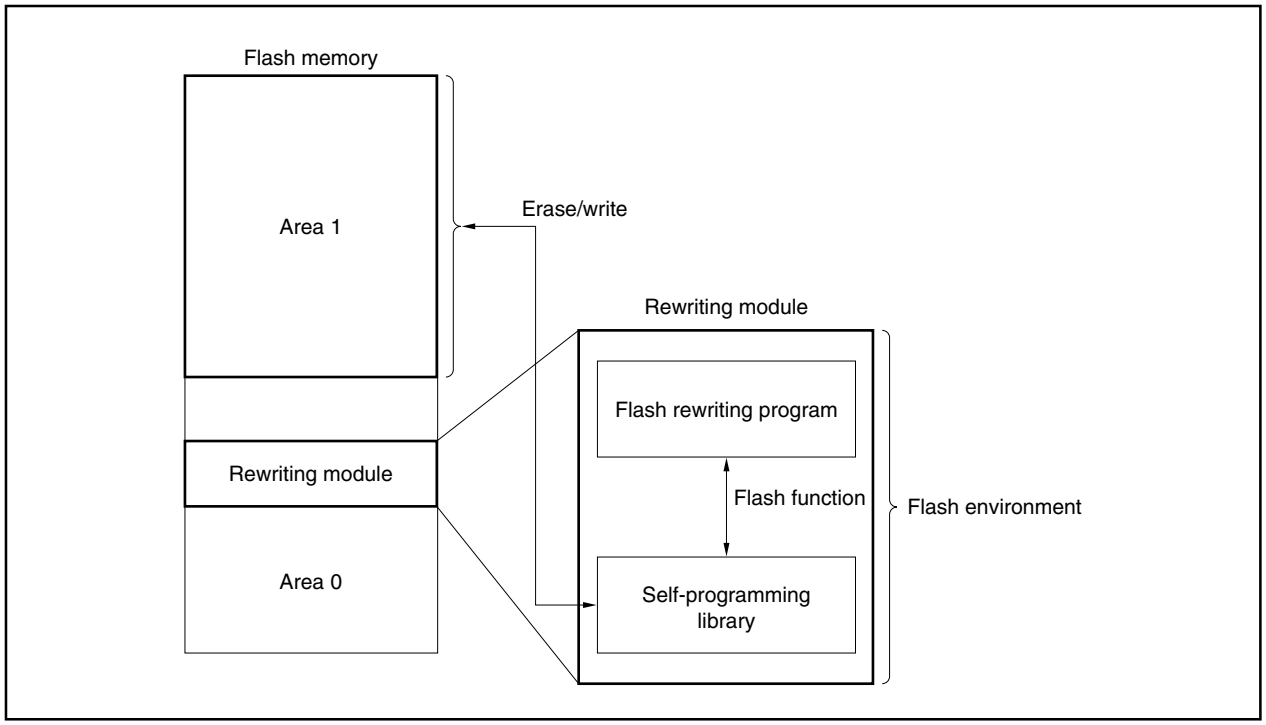
(1) Functional outline

Figure 16-11 outlines the function of the self-programming library. In this figure, a rewriting module is located in area 0 and the data in area 1 is rewritten or erased.

The rewriting module is a user program to rewrite the flash memory. The other areas can be also rewritten by using the flash functions included in this self-programming library. The flash functions expand the entry program in the external memory or internal RAM and call the device internal processing.

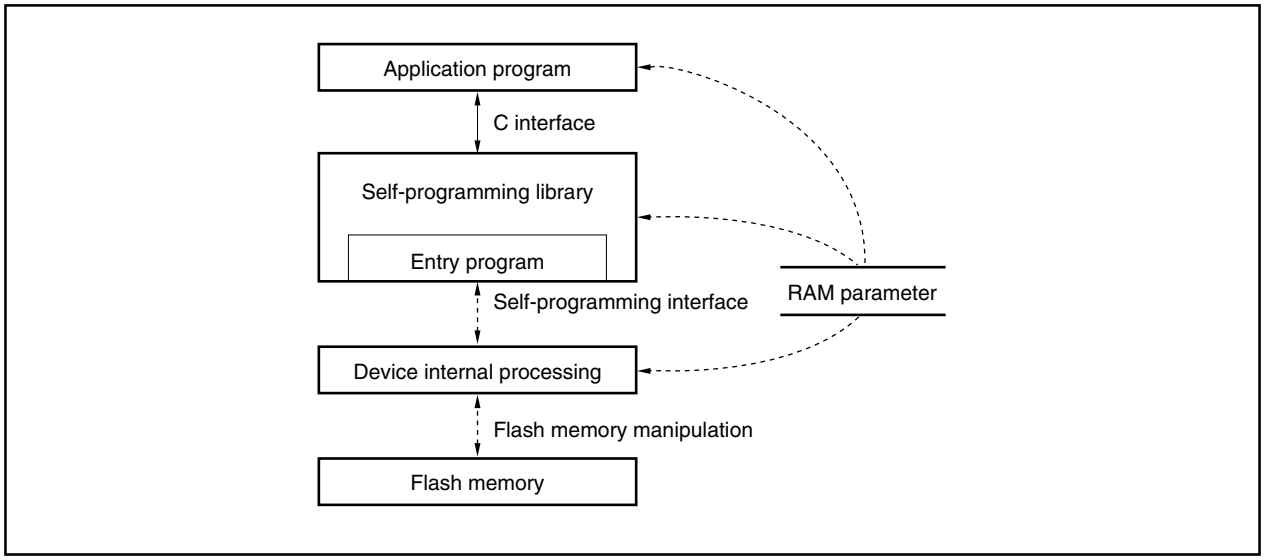
When using the self-programming library, make sure that the hardware conditions, such as the write voltage, and the software conditions, such as interrupts, are satisfied.

Figure 16-11. Functional Outline of Self-Programming Library



The configuration of the self-programming library is outlined below.

Figure 16-12. Outline of Self-Programming Library Configuration



16.8 How to Distinguish Flash Memory and Mask ROM Versions

It is possible to distinguish a flash memory version (μPD70F3107) and mask ROM versions (μPD703105, 703106, 703107) by means of software, using the methods shown below.

<1> Disable interrupts (set the NP bit of PSW to 1).
<2> Write data to the peripheral command register (PHCMD).
<3> Set the VPPDIS bit of the flash programming mode control register (FLPMC) to 1.
<4> Insert NOP instructions (5 instructions (<4> to <8>)).
<9> Cancel the interrupt disabled state (reset the NP bit of the PSW to 0).
<10> Read the VPPDIS bit of the flash programming mode control register (FLPMC).
If the value read is 0: Mask ROM version (μPD703105, 703106, 703107)
If the value read is 1: Flash memory version (μPD70F3107)

[Description example]

```
<1> LDSR  rX, 5
<2> ST.B  r10, PHCMD [r0]
<3> SET1  3, FLPMC [r0]
<4> NOP
<5> NOP
<6> NOP
<7> NOP
<8> NOP
<9> LDSR  rY, 5
<10>TST1  3, FLPMC [r0]
    BNZ               <Start address of self-programming routine>
    BR                <Routine when writing is not performed>
```

Remark rX: Value written to the PSW
rY: Value returned to the PSW

Cautions 1. If an interrupt is acknowledged between when PHCMD is issued (<2>) and writing to a specific register (<3>) immediately after issuing PHCMD, writing to a specific register may not be performed and a protection error may occur (the PRERR bit of the PHS register = 1). Therefore, set the NP bit of the PSW to 1 (<1>) to disable interrupt acknowledgment. Similarly, disable acknowledgement of interrupts when a bit manipulation instruction is used to set a specific register.

2. When a store instruction is used for setting a specific register, be sure to use the same general-purpose register used to set the specific register for writing to the PHCMD register even though the data written to the PHCMD register is dummy data. This is the same as when a general-purpose register is used for addressing.

3. Do not use DMA transfer for writing to the PHCMD register and a specific register.

APPENDIX A REGISTER INDEX

(1/8)

Register Symbol	Register Name	Unit	Page
ADCR0	A/D conversion result register 0 (10 bits)	ADC	406
ADCR0H	A/D conversion result register 0H (8 bits)	ADC	406
ADCR1	A/D conversion result register 1 (10 bits)	ADC	406
ADCR1H	A/D conversion result register 1H (8 bits)	ADC	406
ADCR2	A/D conversion result register 2 (10 bits)	ADC	406
ADCR2H	A/D conversion result register 2H (8 bits)	ADC	406
ADCR3	A/D conversion result register 3 (10 bits)	ADC	406
ADCR3H	A/D conversion result register 3H (8 bits)	ADC	406
ADCR4	A/D conversion result register 4 (10 bits)	ADC	406
ADCR4H	A/D conversion result register 4H (8 bits)	ADC	406
ADCR5	A/D conversion result register 5 (10 bits)	ADC	406
ADCR5H	A/D conversion result register 5H (8 bits)	ADC	406
ADCR6	A/D conversion result register 6 (10 bits)	ADC	406
ADCR6H	A/D conversion result register 6H (8 bits)	ADC	406
ADCR7	A/D conversion result register 7 (10 bits)	ADC	406
ADCR7H	A/D conversion result register 7H (8 bits)	ADC	406
ADIC	Interrupt control register	INTC	277
ADM0	A/D converter mode register 0	ADC	402
ADM1	A/D converter mode register 1	ADC	404
ADM2	A/D converter mode register 2	ADC	405
ASC	Address setup wait control register	BCU	120
ASIF0	Asynchronous serial interface transmission status register 0	UART0	363
ASIF1	Asynchronous serial interface transmission status register 1	UART1	363
ASIF2	Asynchronous serial interface transmission status register 2	UART2	363
ASIM0	Asynchronous serial interface mode register 0	UART0	359
ASIM1	Asynchronous serial interface mode register 1	UART1	359
ASIM2	Asynchronous serial interface mode register 2	UART2	359
ASIS0	Asynchronous serial interface status register 0	UART0	362
ASIS1	Asynchronous serial interface status register 1	UART1	362
ASIS2	Asynchronous serial interface status register 2	UART2	362
BCC	Bus cycle control register	BCU	125
BCP	Bus cycle period control register	BCU	121
BCT0	Bus cycle type configuration register 0	BCU	101
BCT1	Bus cycle type configuration register 1	BCU	101
BEC	Endian configuration register	BCU	104

(2/8)

Register Symbol	Register Name	Unit	Page
BRGC0	Baud rate generator control register 0	BRG0	379
BRGC1	Baud rate generator control register 1	BRG1	379
BRGC2	Baud rate generator control register 2	BRG2	379
BSC	Bus size configuration register	BCU	103
CCC00	Capture/compare register C00	RPU	324
CCC01	Capture/compare register C01	RPU	324
CCC10	Capture/compare register C10	RPU	324
CCC11	Capture/compare register C11	RPU	324
CCC20	Capture/compare register C20	RPU	324
CCC21	Capture/compare register C21	RPU	324
CCC30	Capture/compare register C30	RPU	324
CCC31	Capture/compare register C31	RPU	324
CKC	Clock control register	CG	300
CKSR0	Clock select register 0	UART0	378
CKSR1	Clock select register 1	UART1	378
CKSR2	Clock select register 2	UART2	378
CMD0	Compare register D0	RPU	348
CMD1	Compare register D1	RPU	348
CMD2	Compare register D2	RPU	348
CMD3	Compare register D3	RPU	348
CMICD0	Interrupt control register	INTC	277
CMICD1	Interrupt control register	INTC	277
CMICD2	Interrupt control register	INTC	277
CMICD3	Interrupt control register	INTC	277
CSC0	Chip area select control register 0	BCU	97
CSC1	Chip area select control register 1	BCU	97
CSIC0	Clocked serial interface clock selection register 0	CSI0	389
CSIC1	Clocked serial interface clock selection register 1	CSI1	389
CSIC2	Clocked serial interface clock selection register 2	CSI2	389
CSIIC0	Interrupt control register	INTC	277
CSIIC1	Interrupt control register	INTC	277
CSIIC2	Interrupt control register	INTC	277
CSIM0	Clocked serial interface mode register 0	CSI0	387
CSIM1	Clocked serial interface mode register 1	CSI1	387
CSIM2	Clocked serial interface mode register 2	CSI2	387
DADC0	DMA addressing control register 0	DMAC	212
DADC1	DMA addressing control register 1	DMAC	212
DADC2	DMA addressing control register 2	DMAC	212
DADC3	DMA addressing control register 3	DMAC	212

(3/8)

Register Symbol	Register Name	Unit	Page
DBC0	DMA byte count register 0	DMAC	211
DBC1	DMA byte count register 1	DMAC	211
DBC2	DMA byte count register 2	DMAC	211
DBC3	DMA byte count register 3	DMAC	211
DCHC0	DMA channel control register 0	DMAC	214
DCHC1	DMA channel control register 1	DMAC	214
DCHC2	DMA channel control register 2	DMAC	214
DCHC3	DMA channel control register 3	DMAC	214
DDA0H	DMA destination address register 0H	DMAC	209
DDA0L	DMA destination address register 0L	DMAC	210
DDA1H	DMA destination address register 1H	DMAC	209
DDA1L	DMA destination address register 1L	DMAC	210
DDA2H	DMA destination address register 2H	DMAC	209
DDA2L	DMA destination address register 2L	DMAC	210
DDA3H	DMA destination address register 3H	DMAC	209
DDA3L	DMA destination address register 3L	DMAC	210
DDIS	DMA disable status register	DMAC	215
DMAIC0	Interrupt control register	INTC	277
DMAIC1	Interrupt control register	INTC	277
DMAIC2	Interrupt control register	INTC	277
DMAIC3	Interrupt control register	INTC	277
DRST	DMA restart register	DMAC	215
DSA0H	DMA source address register 0H	DMAC	207
DSA0L	DMA source address register 0L	DMAC	208
DSA1H	DMA source address register 1H	DMAC	207
DSA1L	DMA source address register 1L	DMAC	208
DSA2H	DMA source address register 2H	DMAC	207
DSA2L	DMA source address register 2L	DMAC	208
DSA3H	DMA source address register 3H	DMAC	207
DSA3L	DMA source address register 3L	DMAC	208
DTFR0	DMA trigger factor register 0	DMAC	217
DTFR1	DMA trigger factor register 1	DMAC	217
DTFR2	DMA trigger factor register 2	DMAC	217
DTFR3	DMA trigger factor register 3	DMAC	217
DTOC	DMA terminal count output control register	DMAC	216
DWC0	Data wait control register 0	BCU	118
DWC1	Data wait control register 1	BCU	118
FLPMC	Flash programming mode control register	CPU	518

(4/8)

Register Symbol	Register Name	Unit	Page
IMR0	Interrupt mask register 0	INTC	280
IMR1	Interrupt mask register 1	INTC	280
IMR2	Interrupt mask register 2	INTC	280
IMR3	Interrupt mask register 3	INTC	280
INTM0	External interrupt mode register 0	INTC	269
INTM1	External interrupt mode register 1	INTC	282
INTM2	External interrupt mode register 2	INTC	282
INTM3	External interrupt mode register 3	INTC	282
INTM4	External interrupt mode register 4	INTC	282
ISPR	In-service priority register	INTC	281
LOCKR	Lock register	CPU	303
OVIC00	Interrupt control register	INTC	277
OVIC01	Interrupt control register	INTC	277
OVIC02	Interrupt control register	INTC	277
OVIC03	Interrupt control register	INTC	277
P0	Port 0	Port	456
P00IC0	Interrupt control register	INTC	277
P00IC1	Interrupt control register	INTC	277
P01IC0	Interrupt control register	INTC	277
P01IC1	Interrupt control register	INTC	277
P02IC0	Interrupt control register	INTC	277
P02IC1	Interrupt control register	INTC	277
P03IC0	Interrupt control register	INTC	277
P03IC1	Interrupt control register	INTC	277
P1	Port 1	Port	459
P10IC0	Interrupt control register	INTC	277
P10IC1	Interrupt control register	INTC	277
P10IC2	Interrupt control register	INTC	277
P10IC3	Interrupt control register	INTC	277
P11IC0	Interrupt control register	INTC	277
P11IC1	Interrupt control register	INTC	277
P11IC2	Interrupt control register	INTC	277
P11IC3	Interrupt control register	INTC	277
P12IC0	Interrupt control register	INTC	277
P12IC1	Interrupt control register	INTC	277
P12IC2	Interrupt control register	INTC	277
P12IC3	Interrupt control register	INTC	277
P13IC0	Interrupt control register	INTC	277
P13IC1	Interrupt control register	INTC	277
P13IC2	Interrupt control register	INTC	277

(5/8)

Register Symbol	Register Name	Unit	Page
P13IC3	Interrupt control register	INTC	277
P2	Port 2	Port	461
P3	Port 3	Port	465
P4	Port 4	Port	468
P5	Port 5	Port	471
P7	Port 7	Port	473
PAH	Port AH	Port	476
PAL	Port AL	Port	474
PBD	Port BD	Port	492
PCD	Port CD	Port	489
PCM	Port CM	Port	486
PCS	Port CS	Port	480
PCT	Port CT	Port	484
PDL	Port DL	Port	478
PFC0	Port 0 function control register	Port	458
PFC2	Port 2 function control register	Port	464
PFC3	Port 3 function control register	Port	467
PFC4	Port 4 function control register	Port	470
PFCCD	Port CD function control register	Port	491
PFCCM	Port CM function control register	Port	488
PFCCS	Port CS function control register	Port	483
PHCMD	Peripheral command register	CPU	299
PHS	Peripheral status register	CPU	302
PM0	Port 0 mode register	Port	456
PM1	Port 1 mode register	Port	459
PM2	Port 2 mode register	Port	462
PM3	Port 3 mode register	Port	465
PM4	Port 4 mode register	Port	468
PM5	Port 5 mode register	Port	471
PMAH	Port AH mode register	Port	477
PMAL	Port AL mode register	Port	474
PMBD	Port BD mode register	Port	492
PMC0	Port 0 mode control register	Port	457
PMC1	Port 1 mode control register	Port	460
PMC2	Port 2 mode control register	Port	463
PMC3	Port 3 mode control register	Port	466
PMC4	Port 4 mode control register	Port	469
PMC5	Port 5 mode control register	Port	472

(6/8)

Register Symbol	Register Name	Unit	Page
PMCAH	Port AH mode control register	Port	477
PMCAL	Port AL mode control register	Port	475
PMCBD	Port BD mode control register	Port	493
PMCCD	Port CD mode control register	Port	490
PMCCM	Port CM mode control register	Port	487
PMCCS	Port CS mode control register	Port	482
PMCCT	Port CT mode control register	Port	485
PMCD	Port CD mode register	Port	489
PMCDL	Port DL mode control register	Port	479
PMCM	Port CM mode register	Port	486
PMCS	Port CS mode register	Port	481
PMCT	Port CT mode register	Port	484
PMDL	Port DL mode register	Port	478
PRC	Page ROM configuration register	MEMC	153
PRCMD	Command register	CPU	306
PSC	Power-save control register	CPU	307
PSMR	Power-save mode register	CPU	306
PWMB0	PWM buffer register 0	PWM	435
PWMB1	PWM buffer register 1	PWM	435
PWMC0	PWM control register 0	PWM	433
PWMC1	PWM control register 1	PWM	433
RFS1	Refresh control register 1	MEMC	169
	SDRAM refresh control register 1	MEMC	195
RFS3	Refresh control register 3	MEMC	169
	SDRAM refresh control register 3	MEMC	195
RFS4	Refresh control register 4	MEMC	169
	SDRAM refresh control register 4	MEMC	195
RFS6	Refresh control register 6	MEMC	169
	SDRAM refresh control register 6	MEMC	195
RWC	Refresh wait control register	MEMC	171
RXB0	Receive buffer register 0	UART0	364
RXB1	Receive buffer register 1	UART1	364
RXB2	Receive buffer register 2	UART2	364
SCR1	DRAM configuration register 1	MEMC	161
	SDRAM configuration register 1	MEMC	179
SCR3	DRAM configuration register 3	MEMC	161
	SDRAM configuration register 3	MEMC	179

(7/8)

Register Symbol	Register Name	Unit	Page
SCR4	DRAM configuration register 4	MEMC	161
	SDRAM configuration register 4	MEMC	179
SCR6	DRAM configuration register 6	MEMC	161
	SDRAM configuration register 6	MEMC	179
SEIC0	Interrupt control register	INTC	277
SEIC1	Interrupt control register	INTC	277
SEIC2	Interrupt control register	INTC	277
SESC0	Valid edge select register C0	INTC	284, 330
SESC1	Valid edge select register C1	INTC	284, 330
SESC2	Valid edge select register C2	INTC	284, 330
SESC3	Valid edge select register C3	INTC	284, 330
SIO0	Serial I/O shift register 0	CSI0	391
SIO1	Serial I/O shift register 1	CSI1	391
SIO2	Serial I/O shift register 2	CSI2	391
SIOE0	Receive-only serial I/O shift register 0	CSI0	392
SIOE1	Receive-only serial I/O shift register 1	CSI1	392
SIOE2	Receive-only serial I/O shift register 2	CSI2	392
SOTB0	Clocked serial interface transmit buffer register 0	CSI0	393
SOTB1	Clocked serial interface transmit buffer register 1	CSI1	393
SOTB2	Clocked serial interface transmit buffer register 2	CSI2	393
SRIC0	Interrupt control register	INTC	277
SRIC1	Interrupt control register	INTC	277
SRIC2	Interrupt control register	INTC	277
STIC0	Interrupt control register	INTC	277
STIC1	Interrupt control register	INTC	277
STIC2	Interrupt control register	INTC	277
TMC0	Timer C0	RPU	322
TMC1	Timer C1	RPU	322
TMC2	Timer C2	RPU	322
TMC3	Timer C3	RPU	322
TMCC00	Timer mode control register C00	RPU	326
TMCC01	Timer mode control register C01	RPU	328
TMCC10	Timer mode control register C10	RPU	326
TMCC11	Timer mode control register C11	RPU	328
TMCC20	Timer mode control register C20	RPU	326
TMCC21	Timer mode control register C21	RPU	328
TMCC30	Timer mode control register C30	RPU	326
TMCC31	Timer mode control register C31	RPU	328

(8/8)

Register Symbol	Register Name	Unit	Page
TMCD0	Timer mode control register D0	RPU	350
TMCD1	Timer mode control register D1	RPU	350
TMCD2	Timer mode control register D2	RPU	350
TMCD3	Timer mode control register D3	RPU	350
TMD0	Timer D0	RPU	347
TMD1	Timer D1	RPU	347
TMD2	Timer D2	RPU	347
TMD3	Timer D3	RPU	347
TXB0	Transmit buffer register 0	UART0	365
TXB1	Transmit buffer register 1	UART1	365
TXB2	Transmit buffer register 2	UART2	365
VSWC	System wait control register	BCU	93

APPENDIX B INSTRUCTION SET LIST

B.1 Convention

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose register: Used as source register.
reg2	General-purpose register: Used mainly as destination register. Also used as source register in some instructions.
reg3	General-purpose register: Used mainly to store the remainders of division results and the higher 3 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list
S	1-bit data that specifies a system register in the register list

(3) Register symbols used in operation

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Half-word	Halfword (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in an execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
X	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z/E	0 0 1 0	Z = 1	Zero Equal
NZ/NE	1 0 1 0	Z = 0	Not zero Not equal
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
N	0 1 0 0	S = 1	Negative
P	1 1 0 0	S = 0	Positive
T	0 1 0 1	—	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (In Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock i	Execution Clock r	Execution Clock l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	0	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR[reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Half-word))		4	4	4					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (returned PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0 0 0 0 0 1 1 0 0 1 i i i i i L L L L L L L L L L L L 0 0 0 0 0	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0 0 0 0 0 1 1 0 0 1 i i i i i L L L L L L L L L L L L R R R R R **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) R[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 1 0 0 0 0 0 0	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35					
DIVH	reg1,reg2	r r r r r 0 0 0 0 1 0 R R R R R	GR[reg2]←GR[reg2]÷GR[reg1] Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 0 0 0	GR[reg2]←GR[reg2]÷GR[reg1] Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 0 1 0	GR[reg2]←GR[reg2]÷GR[reg1] Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 1 0 0 0 0 1 0	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	r r r r r 1 1 1 1 1 1 0 0 0 0 0 w w w w w 0 1 1 0 1 0 0 0 1 0 0	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	r r r r r 1 1 1 1 0 d d d d d d d d d d d d d d d d d d d d d 0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	0 0 0 0 0 0 0 0 0 1 1 R R R R R	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110d d d d d d d d d d d d d d d d d d d d d 0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	r r r r r 1 1 1 0 0 0 R R R R R d d d d d d d d d d d d d d d d	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	r r r r r 1 1 1 1 0 b R R R R R d d d d d d d d d d d d d d d 1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Half-word))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-exend(disp16) GR[reg2]←zero-extend(Load-memory(adr,half-word)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-exend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	2 Note 14	2					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]Note 6xGR[reg1]Note 6		1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2]Note 6xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]Note 6ximm16		1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	2 Note 14	2					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm Note 15	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5) ep←sp/imm	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Half-word))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010ddddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Half-word)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010ddddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Half-word)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Half-word))) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 iiiii 0000000100000000	EIPC ←PC+4 (Return PC) EIPSW ←PSW ECR.EICC ←Interrupt Code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. ddddddddd: Higher 8 bits of disp9.

2. 3 clocks if the final instruction includes PSW write access.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list X load registers. (According to the number of wait states. Also, if there are no wait states, n is the number of list X registers.)

5. RRRRR: other than 00000.

6. The lower halfword data only is valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Lower 4 bits of imm9.

14. In the case of reg2 = reg3 (the lower 32 bits of the results are not written in the register) or reg3 = r0 (the higher 32 bits of the results are not written in the register), shortened by 1 clock.

15. sp/imm: Specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr : Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

APPENDIX C INDEX

[D]

[E]

[F]

[H]

[R]

[S]

[T]

[U]

[V]

[W]

[X]

[MEMO]

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	❑	❑	❑	❑